LV-CTC: NON-AUTOREGRESSIVE ASR WITH CTC AND LATENT VARIABLE MODELS

Yuya Fujita[1], *Shinji Watanabe*[2], *Xuankai Chang*[2], *Takashi Maekaku*[1]

[1] LY Corporation, Tokyo, Japan, [2]Carnegie Mellon University, PA, USA

ABSTRACT

Non-autoregressive (NAR) models for automatic speech recognition (ASR) aim to achieve high accuracy and fast inference by simplifying the autoregressive (AR) generation process of conventional models. Connectionist temporal classification (CTC) is one of the key techniques used in NAR ASR models. In this paper, we propose a new model combining CTC and a latent variable model, which is one of the state-of-the-art models in the neural machine translation research field. A new neural network architecture and formulation specialized for ASR application are introduced. In the proposed model, CTC alignment is assumed to be dependent on the latent variables that are expected to capture dependencies between tokens. Experimental results on a 100 hours subset of Librispeech corpus showed the best recognition accuracy among CTC-based NAR models. On the TED-LIUM2 corpus, the best recognition accuracy is achieved including AR E2E models with faster inference speed.

Index Terms— Latent variable models, CTC, Non-autoregressive, Iterative decoding

1. INTRODUCTION

Automatic speech recognition (ASR) is a technology which has been widely used in the real world speech interface. In most cases, faster inference with high recognition accuracy is preferred. For example, an ASR engine for the input method of a smartphone is required to return the recognition result as soon as possible after the end of an utterance for a better user experience. Another example is automatic captioning of user generated audios or videos whose length might be tens of thousands of hours in total, which demands huge computing resources to process all the contents in a realistic time.

One of the successful models of ASR is an autoregressive (AR) end-to-end (E2E) one [1–4]. The E2E model is comprised of a single neural network (NN). It receives an acoustic feature sequence then generates a recognition hypothesis in a sequential manner by feeding back the previously generated token to the decoder. Specifically, Transformer [5] architecture has been a fundamental building block of recently proposed E2E models [6–8]. It can achieve higher accuracy, but the sequential nature of the decoder limits the efficient use of the parallel computing capability of GPUs or ASICs specialized for NN computation. By utilizing such a capability, aiming to not only achieve faster inference but also reduce power consumption, which should be strictly controlled in an on-device application, non-autoregressive (NAR) E2E model was proposed in the research field of neural machine translation (NMT) [9–11]. It eliminates or eases such sequential processes by generating multiple tokens in one iteration step. It can efficiently use the parallel computing capability and achieve faster decoding at the expense of a drop in accuracy compared with AR models. Some NAR E2E models have been proposed for ASR [12–18] and it is reported that they achieved competitive performance to AR models with faster inference speed under certain conditions [19]. The basic formulation of NAR models assume the offline decoding case however, they can be applied to the streaming case by using block-wise decoding [20, 21] or using them as the second pass refinement of a streaming model [22].

In such NAR models for ASR, connectionist temporal classification (CTC) [23] and its variants [13, 24–26] are widely used. CTC has a monotonic alignment property which is thought to be reasonable for ASR because a sentence is read in left-to-right order. Another major property of CTC is the assumption of conditional independence between tokens. On the other hand, in the research field of NMT, latent variable models are applied as a way to relax the conditional independence assumption of NAR E2E models [27, 28]. They introduce latent variables and assume the output token is dependent on the latent variable space. It is expected that the latent variable captures dependencies between tokens and achieves competitive translation quality compared to AR models with faster inference speed. Inspired by the success of latent variable models in NMT, we propose a NAR E2E ASR based on latent variable models. It is quite natural to apply it to ASR, however, to the best of the author's knowledge, there is no prior work of NAR E2E ASR based on latent variable models.

In this paper, a new model for ASR which combines CTC and latent variable models is proposed. A new architecture of NN and its formulation are introduced. The model can generate a hypothesis using the prior estimator network of the latent variable, which looks at only the acoustic feature sequence, by a single step. It can also refine the hypothesis in an iterative way by feeding back the generated hypothesis to the posterior estimator network which looks at both the acoustic features and the token sequences. The architecture of the NN and its formulation allow the proposed model to theoretically guarantee the performance of CTC and can improve its accuracy not only by iterative decoding but also by introducing additional techniques like intermediate CTC [24].

Experiments are conducted on a 100 hours subset of the Librispeech [29] and TED-LIUM2 [30] corpora. By intensive hyperparamter tuning and the combination of intermediate CTC [24] and self-distillation [31] techniques, the proposed model achieved the best accuracy among CTC-based NAR models on Librispeech. On TED-LIUM2, the proposed model outperformed not only NAR models but also state-of-the-art AR models based on RNN-T and CTC/attention hybrid.

2. RELATED WORK

One of the unique properties of the proposed model is that the variational approximate posterior over latent variables is explicitly dependent on the output token sequence and the CTC alignment is assumed to be dependent on these latent variables. None of the existing NAR E2E ASR using CTC [10, 12–15, 17, 25, 32] assumes such a dependency of CTC alignment in this latent space. For example, the combination of an insertion-based model and CTC proposed in [14] explicitly assumes that the CTC alignment is dependent on a partial hypothesis, but not on latent variables. Another unique aspect of the proposed model is that it can additionally employ techniques used in encoder-decoder architectures like a masked language model (MLM) [11], glancing language model (GLM) [33], and self-

979-8-3503-0689-7/23/$31.00 ©2023 IEEE

distillation [31].

3. METHODS

3.1. General formulation of E2E ASR and CTC

E2E ASR utilizes a NN to model the following posterior distribution $p(C|X)$ over a token sequence $C = (c_n \in \mathcal{V}|n = 1, \cdots, N)$, given a d-dimensional acoustic feature sequence $X = (\mathbf{x}_t \in \mathbb{R}^d|t = 1, \cdots, T)$:

$$p(C|X) = \text{NN}(C, X; \theta). \tag{1}$$

N, T are the length of token and acoustic feature sequences, and $\mathcal{V}$ is a set of distinct tokens. θ is the parameters of the NN. The difference between various E2E models is how to define the $p(C|X)$ and the NN architecture of $\text{NN}(C, X; \theta)$ in Eq. (1). For example, attention-based encoder decoder (AED) [1, 3] assumes left-to-right generation of a token sequence:

$$p(C|X) = \prod_{n=1}^{N} p(c_n|X, c_1, \cdots, c_{n-1}). \tag{2}$$

Then, the posterior of the n-th token, $p(c_n|X, c_1, \cdots, c_{n-1})$ in Eq. (2), is modeled by an encoder and decoder network through an attention mechanism.

CTC, which is one of the E2E models we focus on in this paper, introduces a latent alignment sequence and a mapping function $\mathcal{F}(\cdot)$. The mapping function $\mathcal{F}(\cdot)$ deletes the repetition of the same token and a special *blank* token. Then, the posterior of a token sequence is defined as a summation over all the alignments that gives the same token sequence through the mapping function $\mathcal{F}(\cdot)$.

The alignment sequence A is defined as a sequence of tokens with $\langle \mathtt{b} \rangle$, which is the special *blank* token, and the length is the same as the acoustic feature sequence:

$$A = (a_t \in \mathcal{V} \cup \langle \mathtt{b} \rangle | t = 1, \cdots, T). \tag{3}$$

Then, the posterior over token sequence is defined as follows:

$$\begin{cases} p(C|X) = \displaystyle\sum_{A \in \mathcal{F}^{-1}(C)} \prod_{t=1}^{T} p(a_t|X), \\ p(a_t|X) = \text{NN}(X; \theta). \end{cases} \tag{4}$$

The encoder layer of Transformer [5] and its variants [34, 35] can be used as the $\text{NN}(\cdot)$ in Eq. (4). Conditional independence is assumed because in Eq. (4), the posterior of the alignment depends only on the acoustic feature sequence, unlike Eq. (2), so the dependency between tokens is not taken into account.

3.2. Latent variable models

In latent variable models, the posterior in Eq. (1) is assumed to be marginalized over d^{lat}-dimensional latent variable $Z = (z_u \in \mathbb{R}^{d^{\text{lat}}}|u = 1, \cdots, U)$ whose length is U:

$$p(C|X) = \int p(C|Z, X) p(Z|X) dZ. \tag{5}$$

In general, the integral of Eq. (5) is intractable, hence variational approximate posterior $q(\cdot)$ is introduced and the lower bound, which is also called evidence lower bound (ELBO), is maximized:

$$\begin{aligned} \mathcal{L}^{\text{ELBO}} = \mathbb{E}_{Z \sim q} \left[\log p^{\text{dec}}(C|Z, X) \right] \\ - D^{\text{KL}} \left[q(Z|C, X) || p^{\text{prior}}(Z|X) \right], \end{aligned} \tag{6}$$

where $D^{\text{KL}}(\cdot)$ is the Kullback-Leibler (KL) divergence. The three distributions, $p^{\text{dec}}(\cdot), q(\cdot)$, and $p^{\text{prior}}(\cdot)$ in Eq. (6) are modeled by NNs. The ELBO can be maximized using the reparameterization trick [36] by assuming a Gaussian distribution over the latent variable Z.

In addition, depending on the type of $p^{\text{dec}}(\cdot)$, a length prediction module, which estimates the length U of the latent variable Z, is introduced. For example, if AED is used as in [28], the length prediction module is expected to estimate the length of the output token sequence.

At the inference stage, single step decoding can be performed by sampling Z from the prior $p^{\text{prior}}(Z|X)$ and then feeding it to the decoder $p^{\text{dec}}(C|X, Z)$. Another way is to find a hypothesis which maximizes the ELBO in Eq. (6) using an algorithm with iteration. One such algorithm is proposed in [28], which does not require sampling or beam search.

3.3. Proposed CTC-based ASR using latent variable models

3.3.1. Basic architecture

In the proposed method, CTC is used as the decoder[1] $p^{\text{dec}}(\cdot)$ in Eq. (6) and the alignment posterior of the CTC in Eq. (4) is assumed to be dependent only on the latent variable Z:

$$p^{\text{dec}}(C|Z, X) \triangleq p(C|Z) = \sum_{A \in \mathcal{F}^{-1}(C)} \prod_{t=1}^{T} p(a_t|Z). \tag{7}$$

The reasons for employing CTC are twofold: (1) it does not require a length prediction module and (2) the monotonic alignment property is thought to be reasonable for ASR.

Then, a Transformer or Conformer architecture is used as the NN of each component of Eq. (6) and Eq. (7). The alignment posterior $p(a_t|Z)$ in Eq. (7) is modeled by Conformer self-attention layers, which are depicted as "Decoder" on the top of Figure 1:

$$\begin{cases} p(a_t|Z) = \text{Softmax}\left(\text{Linear}(H^{\text{dec}}; \theta^{\text{out}})\right), \\ H^{\text{dec}} = \text{ConformerSA}(Z; \theta), \\ Z \sim q(Z|C, X). \end{cases} \tag{8}$$

The variational approximate posterior $q(\cdot)$ in Eq. (6) is modeled by the cross-attention layers of Transformer, which is depicted as "posterior estimator" at the bottom right of Figure 1:

$$\begin{cases} q(Z|C, X) = \mathcal{N}(\mu^{\text{pst}}, \sigma^{\text{pst}}), \\ H^{\text{pst}} = \text{TransformerCA}(q = X, k = C, v = C; \phi), \\ \mu^{\text{pst}} = \text{FF}^{\text{pst}}(H^{\text{pst}}; \phi^{\text{m}}), \\ \sigma^{\text{pst}} = \text{FF}^{\text{pst}}(H^{\text{pst}}; \phi^{\text{s}}). \end{cases} \tag{9}$$

The prior $p^{\text{prior}}(\cdot)$ in Eq. (6) is modeled by other self-attention layers of Conformer, which is depicted as "prior estimator" in Figure 1:

$$\begin{cases} p^{\text{prior}}(Z|X) = \mathcal{N}(\mu^{\text{prior}}, \sigma^{\text{prior}}), \\ H^{\text{prior}} = \text{ConformerSA}(X; \omega), \\ \mu^{\text{prior}} = \text{FF}^{\text{prior}}(H^{\text{prior}}; \omega^{\text{m}}), \\ \sigma^{\text{prior}} = \text{FF}^{\text{prior}}(H^{\text{prior}}; \omega^{\text{s}}). \end{cases} \tag{10}$$

The meanings of each component are:

- $\text{ConformerSA}(\cdot; \psi)$: Self-attention layers (encoder block) of Conformer
- $\text{TransformerCA}(q, k, v; \psi)$: Cross-attention layers (decoder block) of Transformer where q, k, v are query, key, and value, respectively
- $\text{FF}(\cdot; \psi)$: Feed forward layer
- $\text{Linear}(\cdot; \psi)$: Linear layer

where ψ is the parameter of the component. The whole architecture is depicted in Figure 1.

By employing this architecture, the latent variables Z are expected to capture token dependencies because the minimization of

[1] When CTC is used, the NN has only an encoder block but, by following the convention of latent variable models, we call the NN as decoder.

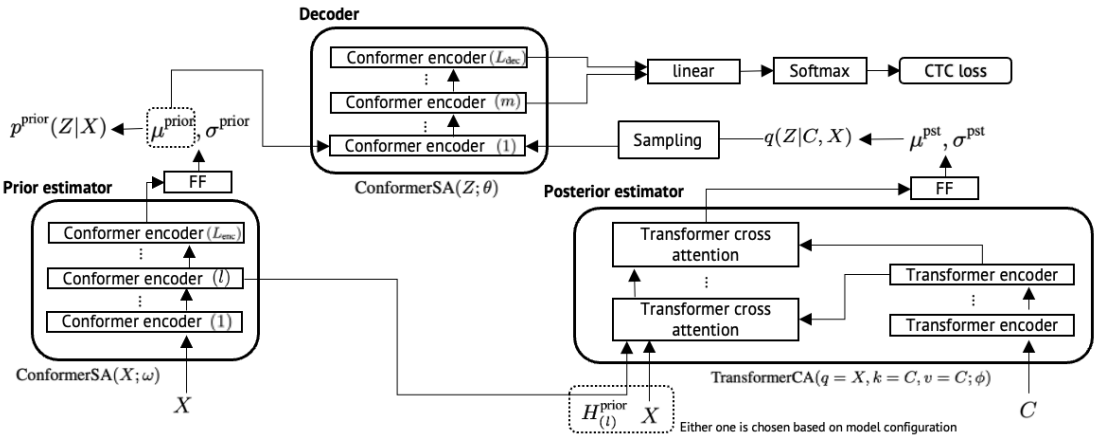


Fig. 1: Architecture of the proposed model.

KL divergence between the posterior $q(Z|C,X)$, which looks at both entire acoustic feature X and token sequences C, and the prior $p^{\text{prior}}(Z|X)$ reflects the token dependency inside C.

3.3.2. Additional techniques

In addition to the basic architecture introduced in the previous section, the following techniques are introduced which are expected to achieve higher accuracy.

Compatibility with vanilla CTC As CTC is a strong baseline, we make the proposed entire network to be compatible with vanilla CTC, so that the training would be stable and guarantee the accuracy of vanilla CTC. This is realized by feeding the mean value calculated by the prior estimator μ^{prior} in Eq. (10) into the decoder's self-attention layers ConformerSA$(\cdot;\theta)$ in Eq. (8) and obtain the alignment posterior similar to Eq. (7):

$$\begin{cases} p(a_t|Z=\mu^{\text{prior}}) = \text{Softmax}\left(\text{Linear}(H^{\text{dec}}_{\text{prior}};\theta^{\text{out}})\right), \\ H^{\text{dec}}_{\text{prior}} = \text{ConformerSA}(Z=\mu^{\text{prior}};\theta). \end{cases} \tag{11}$$

Then, the CTC loss is computed using the alignment posterior and is jointly trained with ELBO of Eq. (6),

$$\mathcal{L}^{\text{ctc}}_{\text{cp}} = \log \sum_{A\in\mathcal{F}^{-1}(C)} \prod_{t=1}^{T} p(a_t|Z=\mu^{\text{prior}}). \tag{12}$$

Sharing encoder layer between prior and posterior estimator In the preliminary experiment, it is observed that the posterior estimator's source-target attention TransformerCA$(\cdot)$ in Eq. (9) tends to be unstable. It might be because the acoustic feature is not well transformed to fit to the source-target attention where similarity between transformed features from a token sequence is calculated. Therefore, the intermediate output of prior estimator's encoder layer ConformerSA$(\cdot;\omega)$ in Eq. (10) is fed into the TransformerCA$(\cdot)$ in Eq. (9):

$$H^{\text{pst}} = \text{TransformerCA}(q=H^{\text{prior}}_{(l)}, k=C, v=C;\phi), \tag{13}$$

where $H^{\text{prior}}_{(l)}$ is the l-th layer's output of ConformerSA$(\cdot;\omega)$ in Eq. (10). In Figure 1, this sharing is depicted as the arrow from the prior estimator on the left side to the posterior estimator on the right side.

Intermediate CTC loss Intermediate CTC [24] is a technique adding CTC losses at the intermediate layers of the encoder block and accuracy improvement is reported. We employed the technique in our proposed architecture by adding an extra CTC loss at an intermediate layer of decoder's ConformerSA$(\cdot;\theta)$ in Eq. (11) and Eq. (8):

$$\begin{cases} p^{\text{prior}}(a_t|H^{\text{dec}}_{\text{prior},(m)}) = \text{Softmax}\left(\text{Linear}(H^{\text{dec}}_{\text{prior},(m)};\theta^{\text{out}})\right), \\ p^{\text{pst}}(a_t|H^{\text{dec}}_{(m)}) = \text{Softmax}\left(\text{Linear}(H^{\text{dec}}_{(m)};\theta^{\text{out}})\right), \end{cases} \tag{14}$$

where $H^{\text{dec}}_{\text{prior},(m)}$ is the output of the m-layer of ConformerSA$(\cdot;\theta)$ in Eq. (11) and $H^{\text{dec}}_{(m)}$ is the output of m-th-layer of ConformerSA$(\cdot;\theta)$ of Eq. (8). Note that the parameters of the Linear layer θ^{out} is shared. Then, the CTC loss is computed using the alignment posterior and they are jointly trained with ELBO of Eq. (6):

$$\begin{cases} \mathcal{L}^{\text{ictc}}_{\text{prior}} = \log \sum_{A\in\mathcal{F}^{-1}(C)} \prod_{t=1}^{T} p(a_t|H^{\text{dec}}_{\text{prior},(m)}), \\ \mathcal{L}^{\text{ictc}}_{\text{pst}} = \log \sum_{A\in\mathcal{F}^{-1}(C)} \prod_{t=1}^{T} p(a_t|H^{\text{dec}}_{(m)}). \end{cases} \tag{15}$$

It is also expected that the instability of source-target attention of TransformerCA$(\cdot)$ in Eq. (9) mentioned before can be mitigated.

Self-distillation Self-distillation (SD) [31] is a technique that was originally proposed to perform distillation in AED model by assuming the output of the decoder as teacher and the encoder is the student during training from scratch [31]. In our proposed model architecture, the alignment posterior of Eq. (8) which is computed by the latent variable sampled from the posterior in Eq. (9) can be viewed as the teacher because it looks at the entire token sequence. Then, the alignment posterior of Eq. (11) becomes the student and the KL-divergence between them are added to the ELBO of Eq. (6):

$$\mathcal{L}^{\text{SD}} = -\sum_{t} D^{\text{KL}}\left(p(a_t|Z=\mu^{\text{prior}})||p(a_t|Z)\right). \tag{16}$$

3.3.3. The loss function

The loss function of the proposed model is the summation of all the losses defined so far. By introducing coefficients, $\alpha_{(\cdot)}$, to adjust the dynamic range of each loss, the loss function is defined as follows:

$$\mathcal{L} = \underbrace{\alpha_{\text{dec}}\mathbb{E}_{Z\sim q}\left[\log p^{\text{dec}}(C|X,Z)\right] - \alpha_{\text{KL}} D^{\text{KL}}\left[q(Z|C,X)||p^{\text{prior}}(Z|X)\right]}_{\mathcal{L}^{\text{ELBO}}}$$
$$+ \alpha_{\text{cp}}\mathcal{L}^{\text{ctc}}_{\text{cp}} + \alpha_{\text{ic1}}\mathcal{L}^{\text{ictc}}_{\text{prior}} + \alpha_{\text{ic2}}\mathcal{L}^{\text{ictc}}_{\text{pst}} + \alpha_{\text{SD}}\mathcal{L}^{\text{SD}}. \tag{17}$$

4. EXPERIMENTS

Two corpora, Librispeech [29] and TEDLIUM2 (TED2) [30] are used to evaluate the proposed method. First, basic experiments of investigating the detail of the NN architecture are performed using

Table 1: WERs of "dev" sets of LS-100 by changing number of layers of prior estimator L_{enc}, the decoder L_{dec}, and the index of encoder sharing l in Eq. (13). Results without iteration (Greedy) and 3 iterations are shown.

			Greedy		3 iterations	
L_{enc}	L_{dec}	l	clean	other	clean	other
3	12	-	7.5	21.0	8.2	23.4
6	9		7.5	21.4	8.4	23.7
9	6		7.9	22.3	7.9	22.1
3	12	1	7.3	21.0	7.1	19.9
		2	7.2	21.4	6.6	19.1
6	9	2	8.2	22.8	88.1	90.3
9	6	3	7.7	22.1	7.2	20.3

a 100 hours subset of Librispeech (LS-100). Then, based on the hyperparameters chosen by the LS-100 results, TED2 is evaluated. Finally, comparison to some existing AR/NAR models are shown.

4.1. Basic experiments and analysis on LS-100

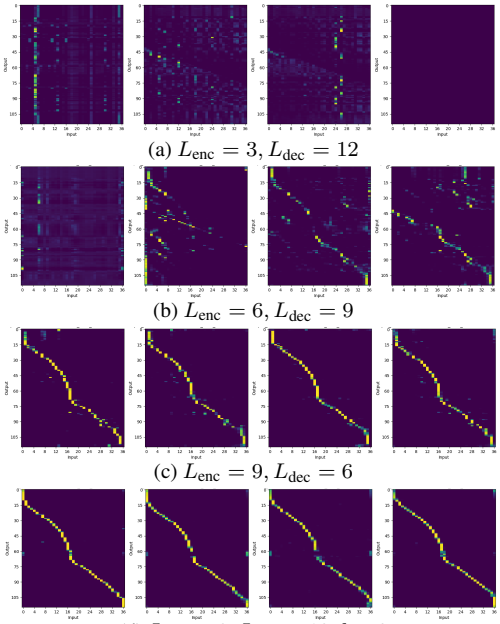
4.1.1. Setup

First, speed perturbation with scaling factor of 0.9 and 1.1 is applied and the perturbed data are added to the original training data. The acoustic feature is 80 dimensional log Mel-filterbank. SpecAugment [37] is applied to the acoustic feature sequence whose parameters are set identical to the recipe of ESPnet [38]. For the output token, byte pair encoding (BPE) is applied with the vocabulary size of 100. Before it is fed into the TransformerCA$(\cdot)$ in Eq. (9), an embedding layer which converts token ids into a real-valued vector is applied. Then, SpecAugment with only time masking is applied. The number of masks are set as 10% of the length of the output token sequence.

The network architecture of the proposed method is as follows. The acoustic feature sequence is down-sampled to 1/4 of the original rate by using 2 layers of convolutional neural network (CNN) whose channels, stride, and kernel size are 256, 2, and 3, respectively. Then, the output of the CNN is fed into ConformerSA$(\cdot)$ and TransformerCA$(\cdot)$ in Eq. (9)-(10). For the attention modules of ConformerSA$(\cdot)$ and TransformerCA$(\cdot)$ in Eq. (8)-(10), relative positional encoding is used [39]. The dimension of the attention is set as 256 and the number of head is 4. The length of kernel, number of hidden units of FF module, and the activation function of ConformerSA$(\cdot)$ in Eq. (8), (10) are 15, 1024, and swish, respectively. The parameters of TransformerCA$(\cdot)$ in Eq. (9) are the same as ConformerSA$(\cdot)$ except that there is no CNN module in it. The number of hidden units of the FF$(\cdot)$ module in Eq. (9)-(10) is 1024 and the activation function is hyperbolic tangent. For all components, the dropout rate is set as 0.1.

The training is run for 50 epochs using 4 Tesla V100 GPUs. For the optimizer and scheduler, Adam [40] with $\beta_1 = 0.90, \beta_2 = 0.98$ and Noam scheduling [5] with warmup step of 15,000, are used. The peak learning rate is set as 0.002. The weight decay is 0.00001. The decoding is performed using the averaged model over the top 10 validation scores.

First, we show a basic experiment of changing the number of layers of ConformerSA$(\cdot)$ in Eq. (8),(10) and the index of the layer shared between prior and posterior network, i.e., l in Eq. (13)[2]. Note that the following parameters are fixed based on a preliminary experiment or intuition. Number of layers of TransformerCA$(\cdot)$ in Eq. (9) is set to 2 and $\alpha_{\text{dec}}, \alpha_{\text{KL}}$, and α_{cp} in Eq. (17) are set to 0.09, 0.1, and 0.81, respectively by intuition [3]. The dimension of the latent variable is 64. In addition, when the KL-divergence of Eq. (17) is smaller than b in a training batch, α_{KL} is set to 0. $b = 0.5$ is used. The last two parameters are fixed by preliminary experiments. We plan to make the configuration files to be publicly available upon publication of the paper.



(a) $L_{\text{enc}} = 3, L_{\text{dec}} = 12$

(b) $L_{\text{enc}} = 6, L_{\text{dec}} = 9$

(c) $L_{\text{enc}} = 9, L_{\text{dec}} = 6$

(d) $L_{\text{enc}} = 3, L_{\text{dec}} = 12, l = 2$

Fig. 2: Visualization of the attention weight of the cross-attention of posterior estimator TransformerCA$(\cdot)$ in Eq. (9). An utterance from training set of Librispeech is chosen. The horizontal axis is the index of token sequence and the vertical axis is the index of acoustic feature sequence.

4.1.2. Results

The number of layers L_{enc} of ConformerSA$(\cdot;\omega)$ in Eq. (10), the number of layers L_{dec} of ConformerSA$(\cdot;\theta)$ in Eq. (8), and l in Eq. (13) are searched. The results are shown in Table 1.

From the upper side of the table, it can be seen that when L_{enc} is small, i.e., the shallower the prior estimator is, better accuracy is obtained except for the case when $L_{\text{enc}} = 9, L_{\text{dec}} = 6$ and iterative decoding is performed. In this case, the attention patterns of the cross attention of TransformerCA$(\cdot)$ of Eq. (9) look more monotonic than the other cases, as visualized in Figure 2. This might lead to slight improvements by iterative decoding.

From the bottom side of the table, the sharing of the encoder in the way of Eq. (13) is effective except the case when $L_{\text{enc}} = 6, L_{\text{dec}} = 9$. In this case, the iterative decoding fails almost completely. On the other hand, by setting $L_{\text{enc}} = 3$, $L_{\text{dec}} = 12$, and $l = 2$, the best WER is achieved. With this setting, the attention patterns of the cross attention of TransformerCA$(\cdot)$ of Eq. (9) look more monotonic as visualized in Figure 2d. According to these results, the monotonic property of the cross attention is important for obtaining better WERs with iterative decoding. In addition, the proposed model is quite sensitive to the parameters of $L_{\text{enc}}, L_{\text{dec}}$, and l.

4.2. Comparison to baseline models

In addition to the best configuration of the previous section, intermediate CTC and self-distillation are introduced. The index of intermediate layer m in Eq. (14) is set to 4. When only the intermediate CTC is applied, the coefficients in Eq. (17) are

[2]There are many other hyperparameters in the proposed model but, according to preliminary experiments, these parameters are crucial to obtain reasonable accuracy so the results are shown in the paper.

[3]It looks α. are too specific but it comes from the difference between actual implementation and the formulation of Eq. (17). In the experiments, actual values for each term were 0.1.

Table 2: Comparison of WERs on LS-100 and TED2. The RTFs are measured on "dev-other" of Librispeech. "Speedup" is the improvement of RTF from Conformer-T decoded with beam size of 10.

Model	Beam or #Iter.	RTF	Speedup	Librispeech dev clean	Librispeech dev other	Librispeech test clean	Librispeech test other	TEDLIUM2 dev	TEDLIUM2 test
AED	1	0.299	0.97	6.7	18.1	7.1	18.2	7.7	8.4
+ Beam search	10	0.797	0.37	6.1	17.9	6.4	17.9	**7.3**	**7.8**
Conformer-T	1	0.057	5.11	6.3	17.6	6.7	17.5	8.1	8.1
+ Beam search	10	0.291	1.00	**5.7**	**16.8**	**6.2**	**16.8**	8.0	8.0
CTC	1	0.044	6.61	6.8	19.7	6.9	19.8	8.1	7.8
Intermediate CTC	1	0.045	6.47	6.1	18.8	6.8	18.9	7.2	7.5
SC-CTC	1	0.047	6.19	6.1	18.9	6.4	19.1	7.3	7.7
LV-CTC (proposed)	1	0.043	6.77	6.4	18.5	6.5	18.9	7.4	7.6
+ Iteration	3	0.084	3.46	6.0	16.8	6.1	17.2	**7.0**	7.2
+ SD	1	0.040	7.28	6.2	18.3	6.5	18.5	7.5	7.5
+ Iteration	3	0.086	3.38	**5.9**	**16.7**	**6.1**	**17.1**	7.1	**7.1**
KERMIT [14]	5	-	-	6.4	17.9	6.6	18.2	8.6	8.0
Improved Mask-CTC [13, 19]	5	-	-	7.0	19.8	7.3	20.2	8.8	8.3
SC-CTC [19, 25]	1	-	-	6.6	19.4	6.9	19.7	8.7	8.0
HC-CTC [26]	1	-	-	6.9	17.1	7.1	17.8	8.0	7.6

$(\alpha_{\text{dec}}, \alpha_{\text{KL}}, \alpha_{\text{cp}}, \alpha_{\text{ic1}}, \alpha_{\text{ic2}})$ =(0.081, 0.1, 0.729, 0.009, 0.081). When both intermediate CTC and self-distillation are applied, the coefficients are $(\alpha_{\text{dec}}, \alpha_{\text{KL}}, \alpha_{\text{cp}}, \alpha_{\text{ic1}}, \alpha_{\text{ic2}}, \alpha_{\text{SD}})$ =(0.073, 0.1, 0.656, 0.008, 0.073, 0.090). Then, the training run for 100 epochs. Four models, Conformer-based AED [41], Conformer-Transducer (Conformer-T) [42], and Conformer-based CTC including intermediate CTC [24] and Self-conditioned CTC (SC-CTC) [25], are reproduced and evaluated as baseline.

Conformer-based AED has 12 layers of Conformer encoder and 6 layers of Transformer decoder. The parameters are the same as the proposed model except that the attention module of the Transformer uses absolute positional embedding. Conformer-based CTC models have 18 layers of Conformer encoder layers whose parameters are the same as the proposed model. For intermediate CTC and SC-CTC, the 6-th and 12-th layers are used as intermediate layers. The Conformer-T has 17 layers of Conformer encoder whose parameters are the same as the proposed model and 1 layer of LSTM decoder whose hidden unit size is 420. The dimension of the joint network is 320. CTC loss is added to the encoder network during training with the weight of 0.3 for AED.

These models are trained for 100 epochs using the same GPUs as the proposed model. The settings of optimizer and scheduler are the same as the proposed model for all the baseline models. The decoding is performed using the averaged model over top 10 validation scores. For the AED model, joint CTC decoding [43] is used with the CTC weight of 0.3. Language model is not used for any of the models. RTF is measured on the dev-other set of Librispeech on an Intel(R) Xeon(R) Gold 6138 CPU @ 2.00GHz CPU using 4 threads for NN inference.

The results are shown in Table 2. In addition to the reproduced baseline models, WERs of some related works are also shown. When compared between baseline CTC models and the proposed LV-CTC without iteration, WERs of LV-CTC with self-distillation (SD) [4] are better than CTC models on "other" sets and are competitive on "clean" sets. From this result, the proposed architecture can at least maintain the performance of baseline CTC models. The LV-CTC with iteration performs the best among all the NAR models shown in the table at the expense of around 2 times increase in RTFs compared with baseline CTC models. It outperforms AED model even with beam search at smaller RTF. When compared with Conformer-T, the proposed method achieved better WERs than greedy decoding results at the expense of the RTF increased to 1.5 times. With beam search, they are competitive or the proposed method is slightly worse but the RTF of the proposed method is reduced to 0.3 times.

[4] In this experiment, the dropout rate of posterior encoder is increased to 0.2 because it performed better when trained until 100 epochs.

4.3. Results on TED2

Based on the best hyperparameter of LS-100 experiments, evaluation on TED2 is performed. Most of the training configurations are the same except the number of epochs is 100. The difference from the best hyperparameter on LS-100 are as follows:

- Number of Transformer encoders in TransformerCA$(\cdot)$ in Eq. (9) is increased from 2 to 3 because in the preliminary experiments TransformerCA$(\cdot)$ is diverged.
- Increased dropout rate and the number of masks of SpecAugment of token embedding because there is a possibility of over-fitting.
- Decreased self-distillation weight α_{SD} to 0.001 because when the weight is larger than it, WER degraded.

The results are shown in Table 2. When compared between baseline CTC models and the proposed LV-CTC without iteration, WERs of baseline models are better than LV-CTC. By using iterative decoding, the proposed LV-CTC has the best WER among all the models shown in the table. Overall, the trend of WER on TED2 is different from that of the LS-100 case. Self-distillation improved test set but degraded dev set. Even AR models' WERs are sometimes worse than baseline CTC models. The speaking style is different between these two corpora: LS-100 is read speech and TED2 is presentation talks. This might be the reason for the different trends but further investigation is left as future work.

5. CONCLUSION

In this paper, we proposed a new ASR model by combining CTC and latent variable models. By introducing a new architecture of NN and its formulation, the proposed model gave at least the same performance of vanilla CTC. In addition, iterative decoding could refine the hypothesis and achieved higher accuracy at the expense of increased inference speed. Experiments are conducted on a 100 hours subset of Librispeech and TED-LIUM2 corpora. On the Librispeech corpus, the proposed model achieved the best WER compared with CTC-based NAR models. On the TED-LIUM2 corpus, the proposed model outperformed NAR and AR models. Investigation of the different trends in WER between the two corpora is left as future work.

6. ACKNOWLEDGEMENT

We would like to thank Jaesong Lee for introduction of his preliminary trial on latent variable models for ASR.

7. REFERENCES

[1] J. K. Chorowski *et al.*, "Attention-based models for speech recognition," in *Proc. Advances in Neural Information Processing Systems (NIPS) 28*, 2015, pp. 577–585.

[2] A. Graves, "Sequence transduction with recurrent neural networks," in *Proc. of the 29th International Conference on Machine Learning (ICML)*, 2012.

[3] W. Chan *et al.*, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2016, pp. 4960–4964.

[4] R. Prabhavalkar *et al.*, "End-to-end speech recognition: A survey," *arXiv preprint arXiv:2303.03329*, 2023.

[5] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Advances in Neural Information Processing Systems (NIPS) 30*, 2017, pp. 5998–6008.

[6] S. Karita *et al.*, "Improving Transformer-Based End-to-End Speech Recognition with Connectionist Temporal Classification and Language Model Integration," in *Proc. Interspeech 2019*, 2019, pp. 1408–1412.

[7] S. Karita *et al.*, "A comparative study on Transformer vs RNN in speech applications," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 449–456.

[8] L. Dong, S. Xu, and B. Xu, "Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition," in *Proc. ICASSP*, IEEE, 2018, pp. 5884–5888.

[9] J. Gu *et al.*, "Non-autoregressive neural machine translation," in *Proc. International Conference on Learning Representations (ICLR)*, 2018.

[10] J. Lee, E. Mansimov, and K. Cho, "Deterministic non-autoregressive neural sequence modeling by iterative refinement," in *Proc. 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 1173–1182.

[11] M. Ghazvininejad *et al.*, "Mask-predict: Parallel decoding of conditional masked language models," in *Proceedings of the EMNLP-IJCNLP*, Hong Kong, China, Nov. 2019, pp. 6112–6121.

[12] W. Chan *et al.*, "Imputer: Sequence modelling via imputation and dynamic programming," in *Proce. of the 37th International Conference on Machine Learning (ICML)*, 2020, pp. 1403–1413.

[13] Y. Higuchi *et al.*, "Improved Mask-CTC for non-autoregressive end-to-end ASR," in *Proc. 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 8363–8367.

[14] Y. Fujita *et al.*, "Insertion-Based Modeling for End-to-End Automatic Speech Recognition," in *Proc. Interspeech 2020*, 2020, pp. 3660–3664.

[15] E. A. Chi, J. Salazar, and K. Kirchhoff, "Align-Refine: Non-autoregressive speech recognition via iterative realignment," in *Proceedings of NAACL-HLT*, 2021, pp. 1920–1927.

[16] N. Chen *et al.*, "Listen and fill in the missing letters: Non-autoregressive transformer for speech recognition.," *arXiv preprint arXiv:1911.04908*, 2020.

[17] R. Fan *et al.*, "CASS-NAT: CTC alignment-based single step non-autoregressive transformer for speech recognition," in *Proc. 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 5889–5893.

[18] F. Yu *et al.*, "Boundary and context aware training for cif-based non-autoregressive end-to-end asr," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2021, pp. 328–334.

[19] Y. Higuchi *et al.*, "A comparative study on non-autoregressive modelings for speech-to-text generation," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2021, pp. 47–54.

[20] T. Wang *et al.*, "Streaming End-to-End ASR Based on Blockwise Non-Autoregressive Models," in *Proc. Interspeech 2021*, 2021, pp. 3755–3759.

[21] Y. Fujita *et al.*, "Toward Streaming ASR with Non-Autoregressive Insertion-Based Model," in *Proc. Interspeech 2021*, 2021, pp. 3740–3744.

[22] W. Wang, K. Hu, and T. N. Sainath, "Deliberation of streaming rnn-transducer by non-autoregressive decoding," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 7452–7456.

[23] A. Graves *et al.*, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. of the 23rd International Conference on Machine Learning (ICML)*, 2006, pp. 369–376.

[24] J. Lee and S. Watanabe, "Intermediate loss regularization for CTC-based speech recognition," in *Proc. 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2021, pp. 6224–6228.

[25] J. Nozaki and T. Komatsu, "Relaxing the Conditional Independence Assumption of CTC-Based ASR by Conditioning on Intermediate Predictions," in *Proc. Interspeech 2021*, 2021, pp. 3735–3739.

[26] Y. Higuchi *et al.*, "Hierarchical conditional end-to-end ASR with CTC and multi-granular subword units," in *Proc. 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2022, pp. 7797–7801.

[27] L. Kaiser *et al.*, "Fast decoding in sequence models using discrete latent variables," in *International Conference on Machine Learning*, PMLR, 2018, pp. 2390–2399.

[28] R. Shu *et al.*, "Latent-variable non-autoregressive neural machine translation with deterministic inference using a delta posterior," in *Proceedings of the aaai conference on artificial intelligence*, vol. 34, 2020, pp. 8846–8853.

[29] V. Panayotov *et al.*, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2015, pp. 5206–5210.

[30] A. Rousseau, P. Deléglise, and Y. Estève, "Enhancing the TED-LIUM corpus with selected data for language modeling and more TED talks," in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, May 2014, pp. 3935–3939.

[31] T. Moriya *et al.*, "Self-Distillation for Improving CTC-Transformer-Based ASR Systems," in *Proc. Interspeech 2020*, 2020, pp. 546–550.

[32] N. Chen *et al.*, "Align-Denoise: Single-Pass Non-Autoregressive Speech Recognition," in *Proc. Interspeech 2021*, 2021, pp. 3770–3774.

[33] L. Qian *et al.*, "Glancing transformer for non-autoregressive neural machine translation," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 1993–2003.

[34] A. Gulati *et al.*, "Conformer: Convolution-augmented Transformer for Speech Recognition," in *Proc. Interspeech 2020*, 2020, pp. 5036–5040.

[35] K. Kim *et al.*, "E-branchformer: Branchformer with enhanced merging for speech recognition," in *2022 IEEE Spoken Language Technology Workshop (SLT)*, IEEE, 2023, pp. 84–91.

[36] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[37] D. S. Park *et al.*, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in *Proc. Interspeech 2019*, 2019, pp. 2613–2617.

[38] S. Watanabe *et al.*, "Espnet: End-to-end speech processing toolkit," in *Proc. Interspeech 2018*, 2018, pp. 2207–2211.

[39] Z. Dai *et al.*, "Transformer-XL: Attentive language models beyond a fixed-length context," in *Proceedings of ACL*, 2019, pp. 2978–2988.

[40] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. International Conference on Learning Representations (ICLR)*, 2015.

[41] P. Guo *et al.*, "Recent developments on ESPnet toolkit boosted by Conformer," in *Proc. 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 5874–5878.

[42] F. Boyer *et al.*, "A study of transducer based end-to-end ASR with ESPnet: Architecture, auxiliary loss and decoding strategies," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, IEEE, 2021, pp. 16–23.

[43] S. Watanabe *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, Dec. 2017.