Interspeech 2024
1-5 September 2024, Kos, Greece

Contextualized End-to-end Automatic Speech Recognition with Intermediate Biasing Loss

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

[1]Honda Research Institute Japan Co., Ltd., Saitama, Japan
[2]Carnegie Mellon University, Pittsburgh, USA

{shakeel.muhammad,yui.sudo}@jp.honda-ri.com, {yifanpen,swatanab}@andrew.cmu.edu

Abstract

Contextualized end-to-end automatic speech recognition has been an active research area, with recent efforts focusing on the implicit learning of contextual phrases based on the final loss objective. However, these approaches ignore the useful contextual knowledge encoded in the intermediate layers. We hypothesize that employing explicit biasing loss as an auxiliary task in the encoder intermediate layers may better align text tokens or audio frames with the desired objectives. Our proposed intermediate biasing loss brings more regularization and contextualization to the network. Our method outperforms a conventional contextual biasing baseline on the LibriSpeech corpus, achieving a relative improvement of 22.5% in biased word error rate (B-WER) and up to 44% compared to the non-contextual baseline with a biasing list size of 100. Moreover, employing RNN-transducer-driven joint decoding further reduces the unbiased word error rate (U-WER), resulting in a more robust network.

Index Terms: end-to-end speech recognition, CTC, RNN transducer, auxiliary learning, contextual biasing

1. Introduction

End-to-end (E2E) automatic speech recognition (ASR) [1, 2] has been a focus of study for its effectiveness in transcribing speech, resulting in substantial reductions in word error rates (WERs). These improvements are made possible by state-of-the-art architectures, including connectionist temporal classification (CTC) [3], recurrent neural network transducers (RNN-transducer) [4,5], and attention-based [6] models. Hybrid methods [7, 8] that combine these techniques, occasionally supplemented with auxiliary learning [9–11] such as intermediate CTC (InterCTC) loss [12], has further boosted performance. However, the effective integration of contextual information, such as specialized vocabulary (referred to as a biasing list), remains an open challenge and often results in out-of-context transcriptions. Thus, it is crucial for E2E ASR architectures to effectively incorporate context-specific prior knowledge to improve transcription accuracy in specific domains.

Several studies have addressed the challenge of incorporating contextual information by investigating modifications to existing architectures, leading to two broad categories of approaches: 1) Shallow fusion-based methods involve transforming bias phrases into weighted finite-state transducers (WFSTs) at word or subword boundaries [13–15], creating a contextual language model (LM) for rescoring. Alternatively, some methods [16–18] model the bias phrases during training, inserting user-defined bias phrases as WFST graphs at appropriate positions in the beam search. These methods either utilize separately trained user-dependent external language models (LMs) or bias phrase information to enhance the performance of user-defined bias phrases through on-the-fly rescoring of the initial ASR hypothesis. Despite their effectiveness, these methods have inherent limitations, such as potential under- or over-biasing due to a lack of joint optimization during training, and may require heuristics to adjust the weights of the contextual LMs. 2) In contrast, neural deep-biasing methods [19–24] directly or indirectly integrate bias phrases with audio features in the intermediate layers [25, 26], often employing attention mechanisms [27–33]. However, they require additional bias phrase information [34–39] either to the encoder or decoder during training or inference. In an alternative approach, auxiliary loss is applied either to the *last* layer of the encoder [32] or decoder [33] to explicitly supervise bias phrases. This improves contextualization by mapping higher-level abstract representations of the audio input to bias phrases. However, it may limit the model's ability to explicitly align bias phrases with the input audio, relying on bias phrase prediction from learned higher-level abstract representations while ignoring lower levels during training. To bridge this gap between input and output representations, and thus improve contextualization, an E2E ASR model should be trained to explicitly align bias phrases within the *intermediate* levels of audio abstraction. This would enable more effective extraction and learning of contextual representations.

In this work, we present a novel method to improve contextualization in E2E ASR models. Our approach proposes to involve an auxiliary task that directly aligns bias phrases with the intermediate audio representations and can be adopted to any E2E ASR architecture. Our key contributions are:

- We introduce a new auxiliary task, the intermediate biasing (IB) loss based on CTC, allowing for improved contextualization by utilizing information across the intermediate representations of the audio encoder.
- We show that the IB loss can be adapted to various architectures such as CTC, attention, and RNN-transducer. In this paper, we specifically introduce its application to CTC [12] and contextual RNN-transducer-based [23] models.
- We achieved substantial reductions in word/character error rates (WER/CER) and biased word/character error rates (B-WER/B-CER) on the Librispeech and in-house datasets compared to conventional approaches. Additionally, RNN-transducer-driven joint decoding with CTC is introduced that further minimizes the unbiased word/character error rates (U-WER/U-CER) to mitigate the effect of biasing on unbiased words/characters when the biasing list size is large.

2. Preliminary

We present an E2E ASR architecture consisting of an audio encoder and an InterCTC [12] loss within both CTC and RNN-transducer [4] formulations.

10.21437/Interspeech.2024-1257

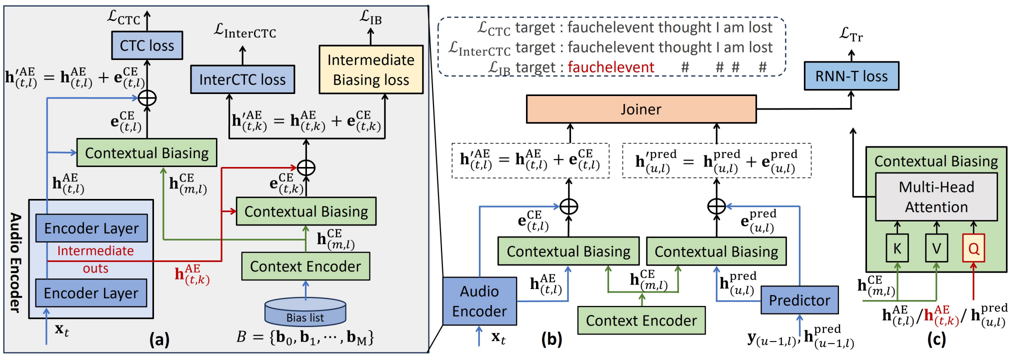
$\mathcal{L}_{\text{CTC}}$
CTC loss
$\mathbf{h}'^{\text{AE}}_{(t,l)} = \mathbf{h}^{\text{AE}}_{(t,l)} + \mathbf{e}^{\text{CE}}_{(t,l)}$
$\mathbf{e}^{\text{CE}}_{(t,l)}$
Contextual Biasing
$\mathbf{h}^{\text{AE}}_{(t,l)}$
$\mathbf{h}^{\text{CE}}_{(m,l)}$
Audio Encoder
Encoder Layer
Intermediate outs
Encoder Layer
$\mathbf{x}_t$
$\mathcal{L}_{\text{InterCTC}}$
InterCTC loss
$\mathcal{L}_{\text{IB}}$
Intermediate Biasing loss
$\mathbf{h}'^{\text{AE}}_{(t,k)} = \mathbf{h}^{\text{AE}}_{(t,k)} + \mathbf{e}^{\text{CE}}_{(t,k)}$
$\mathbf{e}^{\text{CE}}_{(t,k)}$
Contextual Biasing
$\mathbf{h}^{\text{CE}}_{(m,l)}$
$\mathbf{h}^{\text{AE}}_{(t,k)}$
Context Encoder
Bias list
$B = \{\mathbf{b}_0, \mathbf{b}_1, \cdots, \mathbf{b}_{\text{M}}\}$
(a)
$\mathcal{L}_{\text{CTC}}$ target : fauchelevent thought I am lost
$\mathcal{L}_{\text{InterCTC}}$ target : fauchelevent thought I am lost
$\mathcal{L}_{\text{IB}}$ target : fauchelevent # # # #
$\mathcal{L}_{\text{Tr}}$
RNN-T loss
Joiner
$\mathbf{h}'^{\text{AE}}_{(t,l)} = \mathbf{h}^{\text{AE}}_{(t,l)} + \mathbf{e}^{\text{CE}}_{(t,l)}$
$\mathbf{h}'^{\text{pred}}_{(u,l)} = \mathbf{h}^{\text{pred}}_{(u,l)} + \mathbf{e}^{\text{pred}}_{(u,l)}$
$\mathbf{e}^{\text{CE}}_{(t,l)}$
$\mathbf{e}^{\text{pred}}_{(u,l)}$
Contextual Biasing
Contextual Biasing
Audio Encoder
$\mathbf{h}^{\text{AE}}_{(t,l)}$
$\mathbf{h}^{\text{CE}}_{(m,l)}$
$\mathbf{h}^{\text{pred}}_{(u,l)}$
Predictor
Context Encoder
$\mathbf{x}_t$
(b)
$\mathbf{y}_{(u-1,l)}, \mathbf{h}^{\text{pred}}_{(u-1,l)}$
Contextual Biasing
Multi-Head Attention
K
V
Q
$\mathbf{h}^{\text{CE}}_{(m,l)}$
$\mathbf{h}^{\text{AE}}_{(t,l)} / \mathbf{h}^{\text{AE}}_{(t,k)} / \mathbf{h}^{\text{pred}}_{(u,l)}$
(c)

Figure 1: *Proposed model: (a) Audio encoder with the proposed intermediate biasing (IB) loss, (b) integrated with the InterCTC-based [12] contextual RNN-transducer [23], (c) employing a multi-head attention-based (MHA) contextual biasing (CB) module.*

2.1. Audio Encoder

The audio encoder $\text{AudioEnc}(\cdot)$ comprises l encoder layers, where l is a layer index from 1 to N. We leverage the conformer [40] architecture with Transformer and convolutional layers operating on a T-length input acoustic feature sequence $X = \{\mathbf{x}_t \in \mathbb{R}^D | t = 1, \cdots, T\}$, and transforms a subsampled $T'(< T)$-length feature sequence to a high-level representation $H^{\text{AE}}_{(N)} = \{\mathbf{h}^{\text{AE}}_{(t,l)} \in \mathbb{R}^S | t = 1, \cdots, T'\}$, given by:

$$\mathbf{h}^{\text{AE}}_{(t,l)} = \text{AudioEnc}(\mathbf{x}_t), \tag{1}$$

where $\mathbf{x}_t$ and $\mathbf{h}^{\text{AE}}_{(t,l)}$ are a D and S-dimensional acoustic features at frame t for layer l, respectively.

Given an input sequence and its corresponding U-length output sequence $Y = \{y_{(u,l)} \in \mathcal{V} | u = 1, \cdots, U\}$, where $y_{(u,l)}$ is an output token at position u in the vocabulary $\mathcal{V}$ for layer l, the audio encoder is trained using the CTC loss ($\mathcal{L}_{\text{CTC}}$) objective at the *last* (N) layer. The training process is optimized by minimizing the negative log-likelihood given by:

$$\mathcal{L}_{\text{CTC}} = -\log \text{P}_{\text{CTC}}(y_{(u,l)} | \mathbf{h}^{\text{AE}}_{(t,l)}). \tag{2}$$

2.2. InterCTC

The InterCTC loss employs auxiliary CTC losses to the intermediate hidden states of the audio encoder, in addition to the primary CTC loss applied at the *last* (N) layer, as described in Eq. (2). This regularizes the model parameters and is given by:

$$\mathcal{L}_{\text{InterCTC}} = \frac{1}{\mathcal{K}} \sum_{k \in \mathcal{K}} -\log \text{P}_{\text{CTC}}(y_{(u,k)} | \mathbf{h}^{\text{AE}}_{(t,k)}), \tag{3}$$

where $\mathcal{K} \subseteq \{1, \cdots, N-1\}$ represents the positions of the intermediate layers where InterCTC loss is applied. We optimize the audio encoder loss ($\mathcal{L}_{\text{AE}}$) by combining Eqs. (2) and (3) using a tunable hyper-parameter λ_{ic}:

$$\mathcal{L}_{\text{AE}} = (1 - \lambda_{\text{ic}})\mathcal{L}_{\text{CTC}} + \lambda_{\text{ic}}\mathcal{L}_{\text{InterCTC}}. \tag{4}$$

2.3. RNN-transducer

A RNN-transducer [4] typically consists of an audio encoder, a predictor network, and a joiner network. The audio encoder outputs a hidden state vector $\mathbf{h}^{\text{AE}}_{(t,l)}$ for layer l at frame t as described in Section 2.1. The predictor network, denoted as $\text{Predictor}(\cdot)$, takes the previous non-blank label $y_{(u-1,l)}$ as input in Eq. (5) and generates a vector representation $\mathbf{h}^{\text{pred}}_{(u,l)}$ at position u. The joiner network, denoted as $\text{Joiner}(\cdot)$, combines the output of the two networks using a linear layer and predicts the output token in the form of posteriors using Eq. (6):

$$\mathbf{h}^{\text{pred}}_{(u,l)} = \text{Predictor}(y_{(u-1,l)}, \mathbf{h}^{\text{pred}}_{(u-1,l)}), \tag{5}$$

$$\text{P}(y_{(u,l)} | \mathbf{x}_t, y_{(1:u-1,l)}) = \text{Softmax}(\text{Joiner}(\mathbf{h}^{\text{AE}}_{(t,l)}, \mathbf{h}^{\text{pred}}_{(u,l)})), \tag{6}$$

The RNN-transducer loss ($\mathcal{L}_{\text{Tr}}$) then updates the model parameters by minimizing the negative log-likelihood given by:

$$\mathcal{L}_{\text{Tr}} = -\log \text{P}_{\text{Tr}}(y_{(u,l)} | \mathbf{h}^{\text{AE}}_{(t,l)}). \tag{7}$$

3. Proposed architecture

Fig. 1 provides an overview of the proposed model. In this section, we explain the IB loss, context encoder, and a contextual biasing (CB) module (Fig. 1(c)) within the audio encoder (Fig. 1(a)) and decoder components (Fig. 1(b)) for contextualization.

3.1. Context Encoder

Given a list of M bias phrases $B = (\mathbf{b}_0, \mathbf{b}_1, \cdots, \mathbf{b}_M)$, each bias phrase $\mathbf{b}_m$, where $m \in M$, is processed by the context encoder using a BiLSTM model [23]. Here, $\mathbf{b}_0$ represents `<no_bias>` option of not using the context, which is similar to the `<blank>` token for CTC. Moreover, each bias phrase $\mathbf{b}_m \in \mathbb{R}^{L_{\max}}_{+}$ consists of a token sequence of length $L_{\max}$. These token sequences are obtained by zero-padding the extracted phrases to a fixed length $L_{\max}$ greater than zero (+), during the training phase. The context encoder in Eq. (8) then encodes each phrase into a S-dimensional fixed-size phrase vector embedding $H^{\text{CE}} = \{\mathbf{h}^{\text{CE}}_{(m,l)} \in \mathbb{R}^{(M+1) \times S} | m = 1, \cdots, M\}$, given by:

$$\mathbf{h}^{\text{CE}}_{(m,l)} = \text{BiLSTM}(\mathbf{b}_{(m,l)}). \tag{8}$$

The final hidden state of the BiLSTM is then fed to the CB module to obtain a context-aware hidden representation. Since bias list may not always be relevant, the introduction of the dummy `<no_bias>` token enables the model to learn when to enable or disable the biasing option during training and inference.

3.2. Contextual Biasing

Given the intermediate representations of the audio encoder $\mathbf{h}^{\text{AE}}_{(t,l)}$ in Eq. (1) and phrase embedding of the context encoder $\mathbf{h}^{\text{CE}}_{(m,l)}$ in Eq. (8), we insert a CB module employing cross-attention mechanism (Fig. 1(a)) to compute the attention scores $\mathbf{A}_{(t,l)} \in \mathbb{R}^{T' \times (M+1)}$ in Eq. (9) for layer l at a given frame t:

$$\mathbf{A}_{(t,l)} = \text{softmax}\left(\frac{\mathbf{h}^{\text{AE}}_{(t,l)} \mathbf{W}^{\text{Q}} (\mathbf{h}^{\text{CE}}_{(m,l)} \mathbf{W}^{\text{K}})^{\text{T}}}{\sqrt{\text{S}}} \right), \tag{9}$$

where $\mathbf{h}^{\text{AE}}_{(t,l)}$ serves as the attention query, $\mathbf{h}^{\text{CE}}_{(m,l)}$ acts as the keys and $\mathbf{W}^{\text{Q}}$, $\mathbf{W}^{\text{K}} \in \mathbb{R}^{S \times S}$ are the projection matrices that transform query and key. For simplicity, we define a single-head attention mechanism, however, this formulation extends to a multi-head attention (MHA) scenario.

$$\mathbf{e}^{\text{CE}}_{(t,l)} = \mathbf{A}_{(t,l)} \mathbf{h}^{\text{CE}}_{(m,l)} \mathbf{W}^{\mathbf{V}}; \quad \mathbf{h}'^{\text{AE}}_{(t,l)} = \mathbf{h}^{\text{AE}}_{(t,l)} + \mathbf{e}^{\text{CE}}_{(t,l)}. \tag{10}$$

We compute the biasing vector $\mathbf{e}^{\text{CE}}_{(t,l)}$ in Eq. (10) using a weighted sum of attention scores given the projection matrix $\mathbf{W}^{\mathbf{V}} \in \mathbb{R}^{S \times S}$ for value. We update the context-aware representation $\mathbf{h}'^{\text{AE}}_{(t,l)}$ (Fig. 1(c)) using the element wise addition between the hidden states of the audio encoder $\mathbf{h}^{\text{AE}}_{(t,l)}$ and the context encoder $\mathbf{e}^{\text{CE}}_{(t,l)}$.

3.3. Intermediate Biasing Loss

Prior studies [32, 33] show that using predicted posteriors and contextual phrases to compute CTC loss can improve contextual adaptation in E2E ASR models. However, their investigation is limited to the *last* (N) layer of the encoder and shows minimal impact on contextual recognition when presented with a large list of bias phrases. We address these limitations by adapting the loss function to the *intermediate* layers of the encoder.

In this work, we employ the IB loss to introduce phrase-level contextual modeling to the ASR model. It allows us to extract bias embeddings from the utterance. For example, given a reference bias phrase "fauchelevent" in the reference transcript "fauchelevent thought I am lost", we generate a training target for the IB loss "fauchelevent # # # #" by retaining the tokens of the target phrase and substituting the other tokens with a dummy token # (representing <no-bias> output). We compute the IB loss based on the CTC objective in Eq. (11) for the bias phrases $\mathbf{b}_{(m,k)}$ (Section 3.1).

$$\mathcal{L}_{\text{IB}} = \frac{1}{\mathcal{K}} \sum_{k\in\mathcal{K}} -\log \mathrm{P}_{\text{ctc}}(\mathbf{b}_{(m,k)} | \mathbf{h}^{'\text{AE}}_{(t,k)}). \quad (11)$$

We optimize the IB loss at a subset of layers $\mathcal{K}$ (Section 2.2) by obtaining the intermediate context-aware representations $\mathbf{h}^{'\text{AE}}_{(t,k)}$ processed via CB module using Eq. (10) for explicit alignment. We optimize the audio encoder loss using Eqs.(4) and (11) with tunable hyper-parameters λ_{ic} and λ_{ib}:

$$\mathcal{L}_{\text{AE}} = (1-\lambda_{\text{ic}})\mathcal{L}_{\text{CTC}} + \lambda_{\text{ic}}\mathcal{L}_{\text{InterCTC}} + \lambda_{\text{ib}}\mathcal{L}_{\text{IB}}. \quad (12)$$

The IB loss is a variant of the InterCTC loss (Section 2.2) with a significant difference, i.e., it minimizes the CTC objective using the bias phrase sequence instead of the actual target sequence. Additionally, the adapted formulation differs from the original in [32,33] in two ways. Firstly, the former applies to the intermediate hidden states $\mathbf{h}^{'\text{AE}}_{(t,k)}$ of the audio encoder, giving equal weightage to potential contextual cues across different layers, thereby enhancing the biasing process. In contrast, the architecture in [32, 33] only considers the higher-level abstractions of the audio encoder at the *last* (N) layer. Secondly, unlike in [33], where the authors do not modify the training targets for explicitly supervising the biasing task, our method provides explicit supervision. We force-align the bias phrase tokens with the frames to achieve a contextualized model. It ensures that all bias and non-bias frames contribute meaningfully to the loss.

3.4. Contextual RNN-transducer

We modify the original RNN-transducer [4] network and leverage the contextual variant from [23] for this study. Similar to [23], we incorporate two additional modules: a context encoder and a CB module as shown in Fig. 1(b). However, while [23] only trained the CB module as a contextual adapter, we advocate for training the entire model from scratch. It allows all newly integrated components (including the proposed IB loss) to co-adapt, potentially leading to improved contextual performance. Following the processing of the CB module using Eq. (10), we obtain the context-aware representations for the audio encoder $\mathbf{h}^{'\text{AE}}_{(t,l)}$ and the predictor network $\mathbf{h}^{'\text{pred}}_{(u,l)}$ at t and u, respectively. We integrate the CB module between both the encoder-joiner and predictor-joiner networks and update the joiner in Eq. (6), enabling the model to predict the bias phrases within the input speech and is given by:

$$\mathrm{P}(y_u|\mathbf{x}_t, y_{(1:u-1,l)}, \mathbf{h}^{\text{CE}}_{(m,l)}) = \text{Softmax}(\text{Joiner}(\mathbf{h}^{'\text{AE}}_{(t,l)}, \mathbf{h}^{'\text{pred}}_{(u,l)})), \quad (13)$$

We optimize the RNN-transducer loss using Eq. (7) given by:

$$\mathcal{L}_{\text{Tr}} = -\log \mathrm{P}_{\text{Tr}}(y_{(u,l)} | \mathbf{h}^{'\text{AE}}_{(t,l)}, \mathbf{h}^{\text{CE}}_{(m,l)}), \quad (14)$$

Finally, the overall training objective is the weighted sum of Eq. (12) and Eq. (14) with λ_{ae} being the tunable hyper-parameter:

$$\mathcal{L} = \lambda_{\text{ae}}\mathcal{L}_{\text{AE}} + (1-\lambda_{\text{ae}})\mathcal{L}_{\text{Tr}}. \quad (15)$$

3.5. RNN-transducer-driven joint decoding

We adopt the RNN-transducer-driven joint decoding with CTC from [8] to mitigate the degradation of U-WER when the bias list size is $M = 1000$ or large. In our proposed joint decoding we employ the contextual RNN-transducer decoder as the primary decoder and additionally account for CTC prefix scoring as proposed in [7] to compute the CTC likelihoods. Subsequently, the hypotheses are scored by combining CTC and RNN-transducer decoders. The CTC score is added with a weight factor μ_{ctc} to the RNN-transducer having a weight factor of μ_{tr}. Top k_{beam} hypotheses are retained for the next time frame based on the obtained joint score with the main beam size.

4. Experiments

Dataset and evaluation metrics. We evaluate our method on the LibriSpeech corpus (LS-960h) [41] (960 hours of English speech), a common speech recognition benchmark. We dynamically generate a biasing list during training, including randomly selected bias phrases (Section 3.1) shared across utterances within the batch. This strategy promotes robustness to out-of-vocabulary (OOV) words. At testing, biasing lists are composed of rare words from the target utterance and distractors from [18]. We evaluate contextual biasing performance using WER, U-WER, and B-WER criterion, as established in [18]. Our goal is to minimize WER and B-WER without substantially increasing U-WER, even with a growing number of distractors.

Experimental setup. We develop baseline and proposed model architectures using the ESPnet2 toolkit [42] to present the proposed method's adaptability to InterCTC-based [12] contextual RNN-transducer [23] models. The input features consists of 80-dimensional Mel filterbanks (window size 512 samples, hop length 160 samples), with SpecAugment applied for data augmentation. The audio encoder comprises two convolutional layers (stride two), a 256-dimensional linear projection, followed by 12 conformer layers (1024 linear units) with layer normalization. Attention layers within the audio encoder utilized four MHA modules (dimension 256). The prediction network contains a single LSTM layer (256 hidden units), while the joint network has one layer (size 320). The context encoder includes an embedding layer (size 256), a two-layer BiLSTM (size 256), and a linear projection layer. The CB module employs MHA (embedding size 256, 4 attention heads) and two linear layers projecting input/output to match embedding size and subsequent layer (size 256). Optimal training weights are determined empirically as $\lambda_{\text{ae}} = 0.3$, $\lambda_{\text{ic}} = 0.66$, and $\lambda_{\text{ib}} = 0.03$. During training, we uniformly extract 0 to 2 bias phrases for each utterance (M_{utt}) of $L_{\text{max}} = 10$ token lengths per batch resulting in a total of M bias phrases ($M_{\text{utt}} \times n_{\text{batch}}$). The InterCTC model is trained for 150 epochs while the RNN-transducer model for 70 epochs (limited by loss divergence), both using a 0.0015 learning rate, 25000 warmup steps, and the Adam optimizer.

5. Results

We first study the effect of the IB loss on the audio encoder and report our findings in Table 1.1. The IB loss, when adapted to an InterCTC-based model, achieves promising results on the LS-960h test sets. We have the following observations from these results: First, the random bias phrase extraction during training provides an augmentation effect, enabling the model to generate more generalizable representations. Second, the integration of

Table 1: *LS-960h test sets results following the test time biasing list selection in Section 4 with different sizes M. Reported metrics are in the following format: WER (U-WER/B-WER).* ***Bold:*** *the proposed method outperforms the baselines.* ***Underlined:*** *the best result.*

ID	Model Name	test-clean				test-other			
		M=0 (no-bias)	M=100	M=500	M=1000	M=0 (no-bias)	M=100	M=500	M=1000
A1	InterCTC-based RNN-T [12, 23] (non-contextual baseline)	3.68 (2.18/15.83)	3.68 (2.18/15.83)	3.68 (2.18/15.83)	3.68 (2.18/15.83)	8.56 (5.88/32.11)	8.56 (5.88/32.11)	8.56 (5.88/32.11)	8.56 (5.88/32.11)
A2	CPPNet-based RNN-T[1] [32] (contextual baseline)	3.93 (2.52/15.36)	3.28 (2.28/11.42)	3.52 (2.20/14.16)	3.67 (2.64/15.13)	9.12 (6.66/30.82)	7.97 (6.05/24.79)	8.30 (5.96/28.86)	8.41 (5.96/29.98)
B1	Intermediate Biasing (proposed) ($\mathcal{K}$ = 2,4,6,8,10)	**3.07** (**1.84/13.07**)	**2.47** (**1.69/8.85**)	**2.88** (**1.80/11.63**)	**3.04** (**1.88/12.58**)	**7.59** (**5.29/27.81**)	**6.66** (**4.99/21.36**)	**7.30** (**5.18/26.00**)	**7.55** (**5.26/27.72**)
B2	B1 + joint decoding (proposed)	**2.85** (**1.60/12.96**)	**2.31** (**1.53/8.60**)	**2.80** (**1.62/11.23**)	**2.84** (**1.66/12.44**)	**7.18** (**4.80/28.09**)	**6.40** (**4.72/21.18**)	**7.10** (**4.91/26.30**)	**7.34** (**5.00/27.93**)

the context encoder via the IB loss facilitates explicit alignment of general and context-specific bias embeddings due to the rules of CTC (Section 2.1) and the advantages of the cross-attention mechanism (Section 3.2). This helps the encoder to understand the broader context and focus on the most relevant features.

Table 1.1: *Effect of the IB loss on audio encoder-only architecture.* ***Bold:*** *the proposed method outperforms the baselines.*

Model	test-clean		test-other	
	M=100	M=1000	M=100	M=1000
InterCTC [12] (non-contextual baseline)	3.88 (2.23/17.32)	3.88 (2.23/17.32)	9.33 (6.29/36.06)	9.33 (6.29/36.06)
CTC-based CPPNet[1] [32] (contextual baseline)	3.78 (2.15/17.04)	3.92 (2.35/18.34)	8.57 (5.63/34.35)	8.90 (5.7/37.30)
Intermediate Biasing (proposed) ($\mathcal{K}$ = 2,4,6,8,10)	**2.85** (**1.63/12.79**)	**3.32** (**1.94/14.44**)	**7.22** (**4.84/28.11**)	**8.23** (**5.49/32.24**)

In Table 1, we further compare our proposed method with the baseline non-contextual model (A1). Furthermore, for a fair comparison, we reproduced the contextual phrase prediction network (CPPNet) based on RNN-transducer (A2) [32] following their experimental settings. However, it did not achieve a similar performance as described in their work.[1] Table 1 shows the proposed model (B1) reduces the WER from 3.68 to 2.47 (32.9% relatively), U-WER from 2.18 to 1.69 (22.5% relatively) and B-WER from 15.83 to 8.85 (44% relatively) on test-clean set compared to A1 model when $M = 100$. In addition, compared to the A2 baseline, our model also achieves substantial improvement in WER, U-WER, and B-WER. Although the proposed model becomes less effective as we increase the biasing list size $M = 1000$, it still improves the performance from the contextual baseline [32].

Effect of joint decoding. In Table 1, we investigate the impact of joint decoding on overall WER and U-WER when using a biasing list size of $M = 1000$. We employ a beam size k_{beam} of 10 and decoder weights μ_{ctc}=0.2 and μ_{tr}=0.8 to perform joint decoding. Results show a substantial improvement in both metrics when employing RNN-transducer-driven joint decoding (B1vsB2). This indicates that joint decoding benefits from both the contextualized audio encoder and the predictor network. Importantly, joint decoding successfully improves U-WER from 1.88 to 1.66 (11.7% relatively) when $M = 1000$, while maintaining similar performance on B-WER. This finding suggests that joint decoding effectively mitigates the performance degradation in U-WER typically seen in contextual ASR systems with increasing the biasing list size.

Effect of the IB loss in different layers. We investigate the effect of the IB loss in intermediate layers by employing it to three different combinations as shown in Table 3. We empirically find that employing the IB loss in alternative intermediate layer was most effective as it shows more improvement in WER, U-WER and B-WER. The observed performance gains suggest that the optimal placement of the IB loss is a critical factor in maximizing its effectiveness. This findings verify our main claim that model achieves and retains stronger contextualization ability if introduced early in the encoder layers with better alignment with the input audio.

Table 3: *Effect of the IB loss on different intermediate encoder layers using LS-960h test sets.* ***Bold:*** *the proposed method outperforms the baseline.* ***Underlined:*** *the best result.*

Model	test-clean		test-other	
	M=100	M=1000	M=100	M=1000
InterCTC-based RNN-T (non-contextual baseline) [12, 23]	3.68 (2.18/15.83)	3.68 (2.18/15.83)	8.56 (5.88/32.11)	8.56 (5.88/32.11)
Intermediate Biasing ($\mathcal{K}$ = 6)	**2.79** (**1.86/10.31**)	**3.12** (**1.95/12.65**)	**6.80** (**5.00/22.61**)	**7.51** (**5.29/26.95**)
Intermediate Biasing ($\mathcal{K}$ = 4,8)	**2.65** (**1.78/9.72**)	**3.14** (**1.85/13.62**)	**6.58** (**4.82/22.00**)	**7.61** (**5.20/28.80**)
Intermediate Biasing ($\mathcal{K}$ = 2,4,6,8,10)	**2.47** (**1.69/8.85**)	**3.04** (**1.88/12.58**)	**6.66** (**4.99/21.36**)	**7.55** (**5.26/27.72**)

Validation on Japanese dataset. Table 4 presents results from our in-house dataset. We collected 93 hours of spoken Japanese across diverse scenarios (e.g., meetings and morning assemblies). We combined this proprietary dataset with 581 hours from the Corpus of Spontaneous Japanese [43] and 181 hours from the Advanced Telecommunications Research Institute International (ATR) speech database [44]. We developed the proposed model following the experimental setup outlined in Section 4. Similar to Table 1, our method outperforms the non-contextual baseline using a biasing list of 203 phrases (including names and technical vocabulary). The model shows improvement in B-CER from 23.65 to 19.31 (18.4% relatively), with slight decreases in CER and U-CER. Joint decoding further reduces B-CER substantially to 16.93 (28.4% relatively) while also improving CER and U-CER performance. The results confirm the method's effectiveness across linguistically very different English and Japanese datasets.

Table 4: *Results obtained on our in-house dataset*

Model	CER	U-CER	B-CER
InterCTC-RNN-T (non-contextual baseline) [12, 23]	10.15	8.34	23.65
Intermediate Biasing ($\mathcal{K}$ = 2,4,6,8,10) (proposed)	10.22	9.00	**19.31**
+ joint decoding	**9.28**	**8.23**	**16.93**

6. Conclusion

In this work, we present a contextualized E2E ASR framework by leveraging intermediate representations within the encoder, enabling it to co-adapt to any E2E ASR architecture. A series of experiments with public and industrial data shows the effectiveness and robustness of auxiliary IB loss in improving the ASR accuracy and contextualization ability of the model without using an external language model. Compared to the alternative formulation [32, 33], the proposed method is integrated into the intermediate layers of the encoder and outperforms on WER, U-WER and B-WER. Furthermore, we analyze the performance with an increasing biasing list size and propose joint decoding to mitigate the degradation of U-WER when using a large biasing list.

[1] Non-availability of source code and training pipeline limits our ability to identify potential shortcomings in our reproduced model.

7. References

[1] J. Li *et al.*, "Recent Advances in End-to-end Automatic Speech Recognition," *APSIPA Transactions on Signal and Information Processing*, vol. 11, no. 1, 2022.

[2] R. Prabhavalkar, T. Hori, T. N. Sainath, R. Schlüter, and S. Watanabe, "End-to-end Speech Recognition: A Survey," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 32, pp. 325–351, 2024.

[3] A. Graves, S. Fernández *et al.*, "Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks," in *Proc. ICML*, 2006, pp. 369–376.

[4] A. Graves and N. Jaitly, "Towards End-to-end Speech Recognition with Recurrent Neural Networks," in *Proc. ICML*, 2014, pp. 1764–1772.

[5] A. Graves, A. rahman Mohamed, and G. E. Hinton, "Speech Recognition with Deep Recurrent Neural Networks," *Proc. ICASSP*, pp. 6645–6649, 2013.

[6] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-Based Models for Speech Recognition," in *Proc. NeurIPS*, vol. 28, 2015.

[7] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, "Hybrid CTC/attention Architecture for End-to-end Speech Recognition," *IEEE J. Selected Topics Signal Process.*, vol. 11, no. 8, pp. 1240–1253, 2017.

[8] 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, pp. 3312–3316.

[9] R. Sanabria and F. Metze, "Hierarchical Multitask Learning with CTC," in *Proc. SLT*, 2018, pp. 485–490.

[10] A. Tjandra, C. Liu *et al.*, "DEJA-VU: Double Feature Presentation and Iterated Loss in Deep Transformer Networks," in *Proc. ICASSP*, 2020, pp. 6899–6903.

[11] C. Liu, F. Zhang *et al.*, "Improving RNN Transducer Based ASR with Auxiliary Tasks," in *Proc. SLT*, 2021, pp. 172–179.

[12] J. Lee and S. Watanabe, "Intermediate Loss Regularization for CTC-Based Speech Recognition," in *Proc. ICASSP*, 2021, pp. 6224–6228.

[13] A. Bruguier, F. Peng, and F. Beaufays, "Learning Personalized Pronunciations for Contact Name Recognition," in *Proc. Interspeech*, 2016, pp. 3096–3100.

[14] I. Williams, A. Kannan, P. S. Aleksic, D. Rybach, and T. N. Sainath, "Contextual Speech Recognition in End-to-end Neural Network Systems using Beam Search," in *Proc. Interspeech*, 2018, pp. 2227–2231.

[15] D. Zhao, T. N. Sainath, D. Rybach, P. Rondon, D. Bhatia, B. Li, and R. Pang, "Shallow-Fusion End-to-end Contextual Biasing," in *Proc. Interspeech*, 2019, pp. 1418–1422.

[16] R. Huang, O. Abdel-Hamid, X. Li, and G. Evermann, "Class LM and Word Mapping for Contextual Biasing in End-to-end ASR," in *Proc. Interspeech*, 2020, pp. 4348–4351.

[17] D. Le, G. Keren, J. Chan *et al.*, "Deep Shallow Fusion for RNN-T Personalization," in *Proc. SLT*, 2020, pp. 251–257.

[18] D. Le, M. Jain, G. Keren *et al.*, "Contextualized Streaming End-to-end Speech Recognition with Trie-Based Deep Biasing and Shallow Fusion," in *Proc. Interspeech*, 2021, pp. 1772–1776.

[19] G. Pundak, T. N. Sainath, R. Prabhavalkar, A. Kannan, and D. Zhao, "Deep Context: End-to-end Contextual Speech Recognition," in *Proc. SLT*, 2018, pp. 418–425.

[20] M. Jain, G. Keren *et al.*, "Contextual RNN-T For Open Domain ASR," in *Proc. Interspeech*, 2020, pp. 11–15.

[21] F. J. Chang, J. Liu, M. H. Radfar *et al.*, "Context-Aware Transformer Transducer for Speech Recognition," in *Proc. ASRU*, 2021, pp. 503–510.

[22] F. Yu, H. Luo, P. Guo *et al.*, "Boundary and Context Aware Training for CIF-Based Non-Autoregressive End-to-end ASR," in *Proc. ASRU*, 2021, pp. 328–334.

[23] K. M. Sathyendra, T. Muniyappa, F.-J. Chang, J. Liu, J. Su *et al.*, "Contextual Adapters for Personalized Speech Recognition in Neural Transducers," in *Proc. ICASSP*, 2022, pp. 8537–8541.

[24] M. Han, L. Dong *et al.*, "Improving End-to-end Contextual Speech Recognition with Fine-Grained Contextual Knowledge Selection," in *Proc. ICASSP*, 2022, pp. 8532–8536.

[25] R. Pandey *et al.*, "Procter: Pronunciation-Aware Contextual Adapter For Personalized Speech Recognition In Neural Transducers," in *Proc. ICASSP*, 2023, pp. 1–5.

[26] Z. Wu *et al.*, "Dual-Mode NAM: Effective Top-K Context Injection for End-to-end ASR," in *Proc. Interspeech*, 2023, pp. 221–225.

[27] D. Kulshreshtha, S. Dingliwal *et al.*, "Multilingual Contextual Adapters To Improve Custom Word Recognition In Low-resource Languages," in *Proc. Interspeech*, 2023, pp. 3302–3306.

[28] T. Xu, Z. Yang, K. Huang, P. Guo *et al.*, "Adaptive Contextual Biasing for Transducer Based Streaming Speech Recognition," in *Proc. Interspeech*, 2023, pp. 1668–1672.

[29] G. Sun, C. Zhang, and P. C. Woodland, "Tree-Constrained Pointer Generator for End-to-end Contextual Speech Recognition," in *Proc. ASRU*, 2021, pp. 780–787.

[30] G. Sun *et al.*, "Minimising Biasing Word Errors for Contextual ASR With the Tree-Constrained Pointer Generator," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 31, pp. 345–354, 2023.

[31] A. Alexandridis, K. M. Sathyendra, G. P. Strimel, F.-J. Chang, A. Rastrow, N. Susanj, and A. Mouchtaris, "Gated Contextual Adapters for Selective Contextual Biasing in Neural Transducers," in *Proc. ICASSP*, 2023, pp. 1–5.

[32] K. Huang, A. Zhang, Z. Yang, P. Guo *et al.*, "Contextualized End-to-end Speech Recognition with Contextual Phrase Prediction Network," in *Proc. Interspeech*, 2023, pp. 4933–4937.

[33] J. Tang, K. Kim, S. Shon, F. Wu, P. Sridhar, and S. Watanabe, "Improving ASR Contextual Biasing with Guided Attention," in *Proc. ICASSP*, 2024.

[34] X. Wang, Y. Liu, J. Li *et al.*, "Towards Contextual Spelling Correction for Customization of End-to-end Speech Recognition Systems," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 30, pp. 3089–3097, 2022.

[35] X. Wang, Y. Liu, J. Li, and S. Zhao, "Improving Contextual Spelling Correction by External Acoustics Attention and Semantic Aware Data Augmentation," in *Proc. ICASSP*, 2023, pp. 1–5.

[36] M. Han, L. Dong, S. Zhou, and B. Xu, "CIF-Based Collaborative Decoding for End-to-end Contextual Speech Recognition," in *Proc. ICASSP*, 2021, pp. 6528–6532.

[37] X. Fu, K. M. Sathyendra, A. Gandhe *et al.*, "Robust Acoustic and Semantic Contextual Biasing In Neural Transducers for Speech Recognition," in *Proc. ICASSP*, 2023, pp. 1–5.

[38] G. Sun, X. Zheng, C. Zhang, and P. C. Woodland, "Can Contextual Biasing Remain Effective with Whisper and GPT-2?" in *Proc. Interspeech*, 2023, pp. 1289–1293.

[39] Y. Sudo, M. Shakeel, Y. Fukumoto, Y. Peng, and S. Watanabe, "Contextualized Automatic Speech Recognition with Attention-Based Bias Phrase Boosted Beam Search," in *Proc. ICASSP*, 2024.

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

[41] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR Corpus Based on Public Domain Audio Books," in *Proc. ICASSP*, 2015, pp. 5206–5210.

[42] S. Watanabe *et al.*, "ESPnet: End-to-end Speech Processing Toolkit," in *Proc. Interspeech*, 2018, pp. 2207–2211.

[43] K. Maekawa, "Corpus of Spontaneous Japanese: Its Design and Evaluation," in *ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition*, 2003.

[44] A. Kurematsu *et al.*, "ATR Japanese Speech Database as a Tool of Speech Recognition and Synthesis," *Speech Communication*, vol. 9, no. 4, pp. 357–363, 1990.