Investigating Context-aware CTC for Pronunciation Assessment: Mitigating Peaky Behavior and Context Independency Assumption

Jiun-Ting Li[1] **Bi-Cheng Yan**[2] **Tien-Hong Lo**[2]
Shih-Hsuan Chiu[3] **Fu-An Chao**[2] **Berlin Chen**[2]

[1]Advanced Technology Laboratory, Chunghwa Telecom Co., Ltd., Taiwan
[2]National Taiwan Normal University, Taipei, Taiwan
[3]National Taiwan University, Taipei, Taiwan
jtli@cht.com.tw

Abstract

Automatic pronunciation assessment (APA) provides L2 learners with scalable and timely feedback on pronunciation proficiency in a target language, typically through goodness of pronunciation (GOP) features. GOP quantifies how well a pronounced phoneme matches the expected target sound by comparing acoustic features against the model's posterior probabilities. Traditional GOP relies on forced alignment to obtain these posteriors, but it suffers from acoustic-induced misalignments that degrade assessment reliability. Although the standard CTC-GOP approach bypasses forced alignment, it is limited by the inherent peaky behavior of CTC-based ASR models, which produces sparse posteriors and lacks stable temporal information. To address these issues in standard CTC, we propose a context-aware CTC framework incorporating output context dependency (OCD) in the CTC topology, along with label prior (LP) and maximum conditional entropy (EnCTC) regularization, to mitigate peakiness and produce more stable ASR logits suitable for GOP computation. Experiments on the speechocean762 corpus demonstrate that our best context-aware configurations achieve superior phoneme-level performance, outperforming the TDNN-F baseline and standard CTC in unified GOPT (phoneme PCC 0.641 vs. 0.612; word total PCC 0.582 vs. 0.549) while narrowing the gap in hierarchical HierCB scoring. These improvements widen the scoring margin between correct and mispronounced phonemes from 0.708 to 0.816 in GOPT. They also reveal that mitigating CTC peakiness and incorporating context dependency significantly enhance CTC-GOP stability and robustness, especially for alignment-free APA models.

1 Introduction

Goodness of pronunciation (GOP) (Witt and Young, 2000; Hu et al., 2015) aims to evaluate whether pronunciation is in the correct way (mispronunciation detection and diagnosis, MDD) (Parikh et al., 2025) and assess their pronunciation proficiency (automatic pronunciation assessment, APA) (Zhang et al., 2021; Chao et al., 2022; Gong et al., 2022; Do et al., 2023; Pei et al., 2024; Yan et al., 2024, 2025). The conventional GOP uses the hidden Markov model, operating force alignment to retrieve the time mapping with logits and the given phoneme sequence, then computing the similarity with aligned phonemes and logits (Zhang et al., 2021). The GOP score and feature have been widely adopted in MDD and APA tasks.

Despite the success, the conventional GOP has a major drawback. It depends on accurate audio-text alignments. Any misalignment degrades assessment quality. To solve this issue, an alignment-free method, named CTC-GOP (Cao et al., 2024), was proposed. CTC-GOP is based on connectionist temporal classification (CTC) (Graves et al., 2006)-based automatic speech recognition (ASR) models, utilizing path-wise decoding results to compute GOP scores and features. It does not need the force alignment procedure, avoiding the misalignment issues.

However, despite its significant contributions, CTC-GOP is not without its limitations. While traditional DNN-HMM systems are precise in timing but often too rigid to accommodate the disfluencies of L2 speech. In contrast, standard CTC is highly flexible but suffers from inherent peaky behavior, where the model strongly prioritizes what was said over when it occurred. Since accurate temporal information is essential for APA tasks, this remains a critical drawback. In detail, the CTC topology introduces a special blank token to act as a symbol boundary. During training, the ASR model over-identifies blank tokens, reinforcing this tendency until most frames are classified as "blank" token (Zeyer et al., 2021). Consequently, the probability distributions of non-peaky frames are underestimated, with optimal phoneme distributions restricted to only a few sharp spikes, making it

Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications, pages 21–33

difficult to handle pronunciation uncertainty in ambiguous pronunciation. While this peaky behavior and the resulting ambiguity in word boundaries do not hinder general ASR tasks, this lack of temporal density degrades the efficacy of CTC-GOP, which relies on stable frame-level posteriors.

Inspired by content-aware GOP (Shi et al., 2020), we explore two methods to alleviate peakiness and improve the accuracy of token onset and offset predictions: (1) a label prior (LP) to the CTC loss to penalize blank-heavy paths (Huang et al., 2024), and (2) maximum conditional entropy-based regularization for CTC (EnCTC), which can be conducted without frame-level supervision due to the lack of labels for phoneme boundaries (Chen et al., 2023). These adjustments aim to produce more informative, less peaky posteriors suitable for phonetic modeling. Furthermore, CTC primarily stems from the core assumptions of context independence. Although modern end-to-end (E2E) architectures, such as TDNN-F (Peddinti et al., 2015; Povey et al., 2018) and attention-based models (Baevski et al., 2020; Hsu et al., 2021), incorporate input context dependency (ICD) through contextualized representations, the standard CTC topology remains constrained by a strong conditional independence assumption. In natural speech, continuous phonemes exhibit significant coarticulation, suggesting that modeling dependencies between preceding and succeeding units may be superior to the standard independence model. To address this, we use context dependency (CD) symbols in CTC topology (Chorowski et al., 2019), referred to as output context dependency (OCD), to investigate the implications of the OCD impact on APA.

After these attempts to fix CTC, we utilize the resulting ASR models to compute CTC-GOP features for two distinct scoring architectures: the unified GOPT (Gong et al., 2022) and the hierarchical HierCB (Yan et al., 2025). Because GOPT maps phoneme-level CTC-GOP features directly to scores, serving as an alignment-free APA model. In contrast, HierCB incorporates additional inputs that require explicit time alignment (TA) information, such as phoneme-level acoustic embeddings and duration features, serving as an alignment-dependent APA model.

To sum up our contributions: (1) We develop a context-dependent CTC framework that incorporates OCD and regularization techniques to mitigate the inherent peaky behavior of standard CTC. (2) We systematically analyze how architectural choices, such as OCD, label priors, and stress markers, impact time alignment (TA) accuracy across multiple speech corpora. (3) We evaluate these improvements across both unigram and hierarchical APA architectures, demonstrating that mitigating CD is a factor for achieving trustworthy scoring results in alignment-dependent assessment models.

2 Methodology

2.1 CTC training with LP

ASR task aims to predict the most probable token sequence $\hat{\mathrm{Y}}$. Given a speech feature matrix X, whether it can be raw speech or a sequence of feature vectors, is conveyed to an audio encoder. It processes X into a sequence of contextualized hidden features. Subsequently, it can be applied to MLP (Popescu et al., 2009), or CD embedding layers (Chorowski et al., 2019), then culminating in a softmax activation function. This process yields the probability distribution over each token in $\hat{\mathrm{Y}}$. CTC models the posterior probability $P(\mathrm{Y}|\mathrm{X})$ by marginalizing all possible alignments $\mathcal{B}^{-1}(\mathrm{Y})$ between X and Y with the conditional independence assumption.

$$P(\mathrm{Y}|\mathrm{X}) = \sum_{\pi \in \mathcal{B}^{-1}(\mathrm{Y})} \prod_{t=1}^{T} p_t(\pi_t|\mathrm{X}), \tag{1}$$

where $\mathcal{B}(\pi)$ collapses an alignment sequence π, merging repeated neighbouring tokens and then removing any blank labels. $p_t(\pi_t|\mathrm{X})$ is the posterior probability of token at time t.

To reduce the number of blanks in the optimal alignment paths, we can apply unigram LPs on each token in π to penalise paths containing too many blank symbols.

$$P^{LP}(\mathrm{Y}|\mathrm{X}) = \sum_{\pi \in \mathcal{B}^{-1}(\mathrm{Y})} \prod_{t=1}^{T} \frac{p_t(\pi_t|\mathrm{X})}{P(\pi_t)^a}, \tag{2}$$

where the hyper-parameter $a \in \mathbb{R}$ is a scalar. When $a = 0$, this is exactly the standard CTC loss. If π has more blank tokens, it will have a larger prior $P(\pi_t)$, in other words, get a larger penalty in its posterior probability, so $\mathcal{B}^{-1}(\mathrm{Y})$, including the optimal one, will avoid blank tokens. Notably, the LPs are applied during training. And if using LP while training, there is no further need to apply LPs while decoding.

2.2 EnCTC

While LP ensures the ASR model is fair to rare symbols, EnCTC ensures the ASR model is cautious and doesn't overfit to its own predictions. We utilize EnCTC to prevent the ASR model from converging prematurely to a single, overconfident alignment path. This is achieved by adding a maximum conditional entropy term to the standard CTC objective:

$$\mathcal{L}_{\text{EnCTC}} = \mathcal{L}_{\text{CTC}} - bH(\pi|\mathrm{Y}, \mathrm{X}), \tag{3}$$

where b regulates the intensity of the regularization. The entropy $H(\cdot)$ measures the uncertainty across all feasible alignment paths. By maximizing this entropy, the ASR model is encouraged to distribute probability mass across multiple nearby paths, leading to less peaky posteriors and more informative frame-level distributions for GOP computation.

2.3 DWFST implementation for standardized CTC

To benefit from HMM's highly accurate time alignment (TA), and simultaneously imitate the CTC, which can be considered to have a one-state-HMM topology with a single self-loop. In literature, it can easily be done by defining a topology (Zhao and Bell, 2024), integrated into an E2E ASR framework with a differentiable weighted finite state transducer (DWFST) (Hannun et al., 2020). The posterior $P(\mathrm{Y}|\mathrm{X})$ can be rewritten as

$$P(\mathrm{Y}|\mathrm{X}) = \sum_{\pi \in \Pi(\mathrm{Y}; \mathbf{T} \circ \mathbf{L})} p(\pi|\mathrm{X}), \tag{4}$$

where $\mathbf{T}$ and $\mathbf{L}$ denote the topology and the lexicon finite-state transducer (FST), and $\Pi(\mathrm{Y}; \mathbf{T} \circ \mathbf{L})$ represents the set of all the token sequences, whose corresponding output sequence is Y, in the composed FST, $\mathbf{T} \circ \mathbf{L}$.

To compute the score of each probable π, we first need to intersect the decoding graph, which is $\mathbf{T} \circ (\mathbf{L} \circ \mathbf{G})$, where $\mathbf{G}$ is the target word sequence, with the emission FST constructed $\mathbf{E}$ from the ASR model's logit outputs, which is like

$$\sum_{\pi \in \Pi(\mathrm{Y}; \boldsymbol{T} \circ \boldsymbol{L})} p(\pi|\mathrm{X}) = \text{TotalScore}(\boldsymbol{E} \circ (\mathbf{T} \circ (\mathbf{L} \circ \mathbf{G}))). \tag{5}$$

2.4 Decoding with alignment graph built without lexicon

The Viterbi decoding does not guarantee the alignment output to be identical to the canonical phone sequence, even if the same word sequence is output. To solve this problem, we follow the idea in Zhang et al. (2021) to build the lexicon-to-grammar FST directly using the provided canonical phone sequence. We first create a linear FST in which input labels are the canonical phone sequences, and the output labels are the corresponding words and epsilons. The disambiguation symbol is added at the tail of LG to ensure the determinization of FST.

2.5 CTC training with context-dependent symbols

To address the limitations of context-independent targets, we transition from monophone to diphone units by following the idea in Chorowski et al. (2019). For a target sequence such as $[q, v, v]$, the corresponding extended transcript is mapped to $[\emptyset q, qv, vv]$, where $\emptyset q$ and qv are variant units for q unit, $\emptyset$ is the blank token, following the alignment constraints:

$$\emptyset^* \; \emptyset q \; \emptyset q^* \; \emptyset^* \; qv \; qv^* \; \emptyset^* \; vv \; vv^* \; \emptyset^*,$$

While this expansion captures local phonetic context, it significantly increases the output dimensionality and computational complexity. To mitigate this, we adopt a context-dependent embedding network to replace the standard final linear layer. This approach generates co-dependent prototype vectors for the diphone units, effectively reducing parameter overhead while enabling the ASR model to generalize across unseen phonetic contexts.

2.6 CTC-GOP computation

The definition of CTC-GOP adopts the approach of enumerating all possible transcription paths involving deletions and substitutions of specific tokens. We denote the canonical phoneme sequence as Y_c. In place of the original GOP-NN (Hu et al., 2015), this method first defines the log posterior probability (*lpp*) scalar as

$$lpp = \log p(\mathrm{Y_c}|\mathrm{X}). \tag{6}$$

The vector of log posterior ratios (**LPR**) for the ith phoneme in Y_c is then defined as

$$\mathbf{LPR}(i) = \log\left(\frac{p(\mathrm{Y_c}|\mathrm{X})}{p(\mathrm{Y}_{sdi}|\mathrm{X})}\right), \tag{7}$$

where the Y_{sdi} is the deletion and substitutions for the ith phoneme in Y_c. The features of the ith phoneme in Y_c is then defined to be:

$$\text{CTC-GOP}(i) = \{lpp, \mathbf{LPR}(i)\}. \quad (8)$$

Regarding the dimensionality of CTC-GOP(i), for a 39-phone set, the **LPR** vector consists of 40 dimensions (accounting for 38 possible substitutions, one deletion, and the canonical phoneme itself). Including the *lpp* scalar, the total feature vector length is 41.

We also describe the details for how to extract CTC-GOP features from a CTC-based ASR model in Appendix A. The pipeline consists of three stages: (1) reduction of the phone inventory from full phone sets to pure phones, (2) batched CTC loss computation over substitution and deletion hypotheses, and (3) construction of **LPR**.

3 Experimental Results

3.1 Corpus

We fine-tune the ASR models on the 960-hour LibriSpeech (Panayotov et al., 2015) corpus using a 72-phoneme inventory (including stress markers). To evaluate TA accuracy, we utilize two standard American English corpora with gold-standard phonetic alignments: TIMIT (Garofolo et al., 1993) (read speech) and Buckeye (Pitt et al., 2005) (conversational speech).

Our evaluation set includes 400 dev and 192 test samples from TIMIT, totaling 177,080 phonemes. For Buckeye, we evaluate 7,492 samples containing 858,386 phonemes. These datasets provide the verbatim timed transcriptions necessary to measure the impact of mitigated peaky behavior on boundary precision. Detailed hyperparameters and implementation are provided in Appendix B.

Both TIMIT (61 pure phones) and Buckeye (76 pure phones) were standardized to the CMU 39-pure phone units. This ensures a uniform phonetic label space across the three corpora, regardless of their original inventory sizes. Then, we add the stress markers back to their phoneme sequences L_{ref}. We first use the TDNN-F model pretrained on Librispeech [1] to transcribe the phoneme sequence L_{hyp}. Then, we apply a positional transfer function to map the possible stress markers from transcription to the converted non-stress labels, by identifying vowel indices V_{ref} and V_{hyp} and applying a conditional mapping strategy:

- Linear mapping: If $|V_{ref}| = |V_{hyp}|$, stress markers are transferred via a direct 1-to-1 ordinal correspondence in L_{ref} and L_{hyp}.
- Relative alignment: If vowel counts diverge, we apply a nearest-neighbor approach based on relative sequence position. For each V_j^{ref} in L_{ref}, the stress digit is borrowed from the V_j^{hyp} in L_{hyp} that minimizes the normalized distance: $\hat{i} = \arg\min_i \left| \frac{i}{|L_{\text{ref}}|} - \frac{j}{|L_{\text{hyp}}|} \right|$

This mechanism ensures that the hypothesis sequence maintains a rhythmic structure consistent with the reference, even in the presence of phoneme insertions or deletions.

In addition to investigating the effect on acoustic variants, we evaluate ASR models trained on two distinct phoneme inventories: a comprehensive 72-unit set and a reduced 39-unit set, to provide a controlled benchmark against the original set.

For APA evaluation, we conducted experiments on the speechocean762 corpus (Zhang et al., 2021), a publicly available dataset specifically designed for APA research. This dataset contains 5,000 English-speaking recordings spoken by 250 Mandarin L2 learners. Each of the training and test sets has 2,500 utterances, where annotates utterance-level scores (accuracy, stress, completeness, fluency, prosodic, total), word-level scores (accuracy, stress, total) and phoneme accuracy score.

3.2 Backbone ASR models

WavLM (Chen et al., 2022) improves upon wav2vec 2.0 (Baevski et al., 2020) and HuBERT (Hsu et al., 2021), enhancing the model's robustness for APA by incorporating gated relative position bias and speech denoising capabilities during pre-training. While it is similar to HuBERT, improving upon wav2vec 2.0 by predicting hidden unit clusters, WavLM's unique focus on simulating noisy and overlapped speech during training allows for more discriminative acoustic representations. These features help capture the small phonetic changes needed for accurate, high-resolution pronunciation scores.

3.3 Metrics

To access TA quality at the word- and phoneme-level, we adopt both absolute and marginal metrics

[1] https://kaldi-asr.org/models/m13

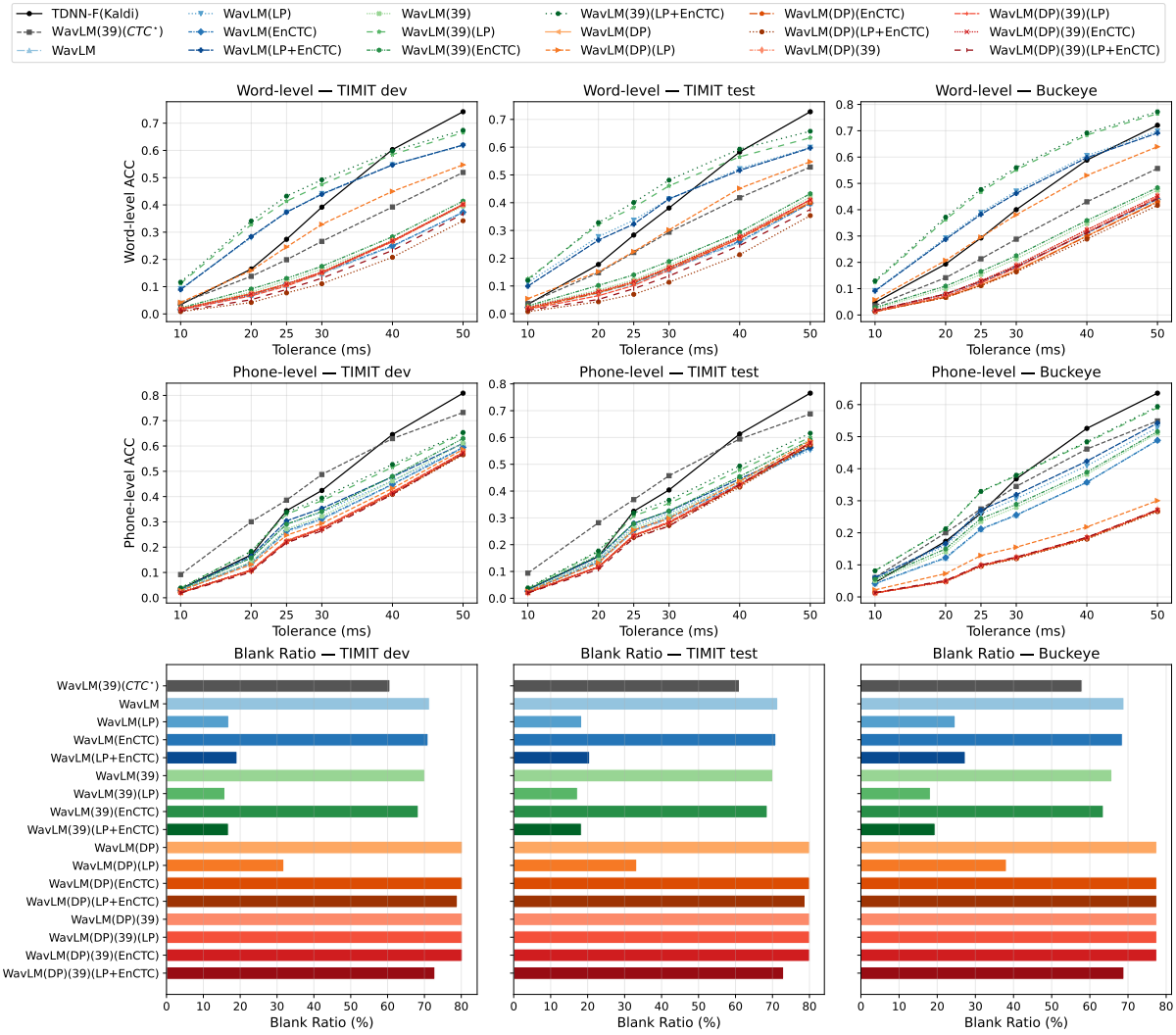

Figure 1: The time stamp error (TSE), alignment accuracy (ACC) and blank ratio (BR) figures. These ASR models test on TIMIT's dev and eval dataset and the whole Buckeye corpus.

(Zhao and Bell, 2024). For the absolute one, we use the time stamp error (TSE), which is defined as:

$$\text{TSE} = \frac{\sum_w |r_s - h_s| + |r_e - h_e|}{N_w}, \tag{9}$$

where the N_w represents the phoneme or word count in each utterance, and r_s, h_s, r_e, h_e indicate the reference, hypothesis' start time and end time, respectively. Another one is alignment accuracy (ACC) for a threshold τ milliseconds, $\tau = \{10, 20, 25, 30, 40, 50\}$, which is defined:

$$\text{ACC}(\tau) = \frac{\sum_w \mathbf{1}(r_s - \tau \leq h_s \cap h_e \leq r_e + \tau)}{N_w}, \tag{10}$$

for each phoneme or word that satisfies the condition, $\mathbf{1}(\star)$ outputs 1, otherwise 0.

To estimate the efficacy of those methods in mitigating CTC's peaky behavior, we also count the blank ratio (BR) to understand whether it decreases with those methods. BR is defined as

$$BR = \frac{\sum_u N_b}{\sum_u N}, \tag{11}$$

where N_b denotes the number of occurrences of the blank state within an evaluation utterance, and N represents the total number of frames in that utterance. The summation $\sum_u$ is executed across all evaluation utterances. This evaluation metric allows us to quantify the proportion of frames that are classified as blank states by our models.

To evaluate the APA model's performance, the Pearson correlation coefficient (PCC) is adopted, while the mean squared error (MSE) is only used for the phoneme accuracy.

ASR Models	Phoneme Score		Word Score (PCC)			Utterance Score (PCC)				
	MSE ↓	PCC ↑	Acc. ↑	Stress ↑	Total ↑	Acc. ↑	Comp. ↑	Fluency ↑	Prosodic ↑	Total ↑
TDNN-F(Kaldi) (Gong et al., 2022)	0.085 ±0.001	0.612 ±0.003	0.533 ±0.004	0.291 ±0.030	0.549 ±0.002	0.714 ±0.004	0.155 ±0.039	0.753 ±0.008	0.760 ±0.006	0.742 ±0.005
WavLM(39)($CTC^{\star}$) (Cao et al., 2024, 2026)	0.102 ±0.003	0.484 ±0.019	0.443 ±0.014	0.112 ±0.021	0.449 ±0.019	0.710 ±0.002	0.142 ±0.090	0.661 ±0.004	0.699 ±0.002	0.719 ±0.002
WavLM (DWFST-CTC)										
-	0.086 ±0.002	0.612 ±0.014	0.554 ±0.014	**0.306** ±**0.027**	0.573 ±0.013	**0.742** ±**0.006**	0.155 ±0.048	0.745 ±0.005	0.749 ±0.004	**0.762** ±**0.006**
(LP)	0.123 ±0.001	0.335 ±0.005	0.347 ±0.002	0.032 ±0.005	0.356 ±0.000	0.621 ±0.008	-0.013 ±0.036	0.655 ±0.015	0.643 ±0.009	0.622 ±0.011
(EnCTC)	0.089 ±0.002	0.590 ±0.009	0.541 ±0.006	0.254 ±0.010	0.559 ±0.007	0.729 ±0.007	0.179 ±0.033	0.739 ±0.003	0.743 ±0.001	0.745 ±0.003
(LP+EnCTC)	0.120 ±0.001	0.352 ±0.012	0.365 ±0.006	0.043 ±0.025	0.379 ±0.006	0.641 ±0.008	-0.007 ±0.012	0.670 ±0.014	0.659 ±0.013	0.654 ±0.011
(39)	0.088 ±0.002	0.600 ±0.010	0.547 ±0.008	0.303 ±0.005	0.565 ±0.009	0.725 ±0.006	0.174 ±0.011	0.740 ±0.006	0.745 ±0.003	0.749 ±0.006
(39)(LP)	0.125 ±0.000	0.319 ±0.000	0.336 ±0.004	0.022 ±0.002	0.347 ±0.001	0.594 ±0.006	0.033 ±0.033	0.621 ±0.013	0.615 ±0.010	0.601 ±0.008
(39)(EnCTC)	0.122 ±0.000	0.325 ±0.001	0.336 ±0.002	0.016 ±0.013	0.347 ±0.003	0.607 ±0.003	0.022 ±0.014	0.641 ±0.006	0.635 ±0.002	0.619 ±0.006
(39)(LP+EnCTC)	0.089 ±0.002	0.595 ±0.012	0.542 ±0.011	0.299 ±0.021	0.560 ±0.011	0.728 ±0.008	**0.230** ±**0.046**	0.736 ±0.005	0.744 ±0.004	0.747 ±0.006
(DP)	0.084 ±0.001	0.621 ±0.004	0.549 ±0.004	0.210 ±0.012	0.567 ±0.003	0.724 ±0.002	0.219 ±0.050	0.720 ±0.005	0.732 ±0.003	0.739 ±0.004
(DP)(LP)	0.106 ±0.001	0.486 ±0.012	0.444 ±0.012	0.134 ±0.026	0.463 ±0.010	0.660 ±0.005	0.094 ±0.019	0.659 ±0.004	0.664 ±0.001	0.675 ±0.004
(DP)(EnCTC)	0.083 ±0.001	0.626 ±0.003	0.553 ±0.004	0.227 ±0.010	0.571 ±0.003	0.727 ±0.004	0.228 ±0.077	0.726 ±0.009	0.739 ±0.006	0.745 ±0.002
(DP)(LP+EnCTC)	0.096 ±0.002	0.558 ±0.006	0.511 ±0.005	0.159 ±0.022	0.527 ±0.004	0.704 ±0.005	0.061 ±0.046	0.702 ±0.008	0.711 ±0.014	0.718 ±0.007
(DP)(39)	**0.081** ±**0.001**	**0.641** ±**0.005**	**0.565** ±**0.005**	0.261 ±0.018	**0.582** ±**0.004**	0.728 ±0.002	0.220 ±0.088	0.726 ±0.003	0.736 ±0.002	0.747 ±0.003
(DP)(39)(LP)	0.081 ±0.001	0.640 ±0.006	0.564 ±0.006	0.262 ±0.016	0.582 ±0.006	0.729 ±0.003	0.208 ±0.091	0.727 ±0.005	0.735 ±0.003	0.748 ±0.004
(DP)(39)(EnCTC)	0.081 ±0.001	0.641 ±0.006	0.565 ±0.005	0.260 ±0.019	0.582 ±0.004	0.728 ±0.002	0.217 ±0.086	0.726 ±0.003	0.736 ±0.002	0.747 ±0.003
(DP)(39)(LP+EnCTC)	0.081 ±0.001	0.637 ±0.006	0.562 ±0.005	0.262 ±0.023	0.580 ±0.005	0.726 ±0.002	0.217 ±0.082	0.724 ±0.003	0.732 ±0.003	0.745 ±0.003

Table 1: Evaluation results for the GOPT model. $CTC^{\star}$ denotes the standard CTC baseline. Results are reported as the mean and standard deviation (±) across five seeds. Bold and underlined text indicate performance exceeding the baseline, with bold representing the best result for each aspect. Models labeled (39) utilize the CMU 39-phone set, while (DP) refers to the proposed diphone units. (LP) and (EnCTC) represent label prior and conditional maximum entropy regularizations, respectively. Shaded rows distinguish models utilizing content-aware CTC.

3.4 Results

We compare our proposed context-dependent CTC methods with the TDNN-F (Kaldi) and standard CTC to evaluate their efficacy on Time Alignment (TA) accuracy and the resulting series of GOP features on the APA models. The TDNN-F baseline is well-known for its outstanding ability in audio-text alignment (Rousso et al., 2024), providing stable posteriors that serve as a robust benchmark for APA tasks. It outperforms all the other methods in the HierCB model, as shown in Table 2. But in GOPT shown in Table 1, WavLM(DP)(39) and WavLM(DP)(39) series outperform the TDNN-F (Kaldi) and standard CTC, and show the best results in many phone- and word-aspects. This observation shows that the content-aware CTC can really help access pronunciation at fine-grained levels. HierCB relies on external hard alignments which may not benefit from the soft posterior density. GOPT is alignment-free and looks at the global feature sequence; it benefits more from the enriched acoustic information in our dense posteriors.

3.4.1 Phone-set Granularity: 39 Phones vs. Full Phoneme Set

We investigated whether the granularity of the units affects alignment and GOP quality. As shown in Figure 1, ASR models using the CMU 39-phone

ASR Models	Phoneme Score		Word Score (PCC)			Utterance Score (PCC)				
	MSE ↓	PCC ↑	Acc. ↑	Stress ↑	Total ↑	Acc. ↑	Comp. ↑	Fluency ↑	Prosodic ↑	Total ↑
TDNN-F(Kaldi) (Yan et al., 2025; Li et al., 2025)	0.078 ±0.001	0.660 ±0.002	0.608 ±0.011	0.385 ±0.045	0.625 ±0.011	0.757 ±0.006	0.685 ±0.153	0.835 ±0.004	0.826 ±0.003	0.787 ±0.004
WavLM(39)($CTC^{\star}$) (Cao et al., 2024, 2026)	0.084 ±0.001	0.624 ±0.002	0.573 ±0.006	0.218 ±0.022	0.589 ±0.006	0.739 ±0.002	0.317 ±0.026	0.737 ±0.004	0.738 ±0.003	0.756 ±0.002
WavLM (DWFST-CTC)										
-	0.088 ±0.001	0.600 ±0.005	0.574 ±0.003	0.217 ±0.011	0.589 ±0.003	0.749 ±0.006	0.272 ±0.033	0.744 ±0.006	0.745 ±0.007	0.765 ±0.004
(LP)	0.114 ±0.002	0.432 ±0.007	0.430 ±0.006	0.097 ±0.008	0.442 ±0.006	0.635 ±0.006	0.099 ±0.016	0.665 ±0.011	0.653 ±0.009	0.645 ±0.008
(EnCTC)	0.088 ±0.002	0.602 ±0.006	0.566 ±0.008	0.203 ±0.020	0.581 ±0.009	0.743 ±0.006	0.248 ±0.046	0.738 ±0.005	0.740 ±0.005	0.762 ±0.004
(LP+EnCTC)	0.105 ±0.002	0.501 ±0.015	0.487 ±0.008	0.153 ±0.011	0.500 ±0.009	0.708 ±0.004	0.285 ±0.066	0.753 ±0.005	0.742 ±0.005	0.731 ±0.003
(39)	0.088 ±0.002	0.600 ±0.010	0.584 ±0.015	0.195 ±0.017	0.599 ±0.014	0.751 ±0.002	0.281 ±0.055	0.742 ±0.005	0.745 ±0.003	0.767 ±0.002
(39)(LP)	0.105 ±0.002	0.497 ±0.009	0.487 ±0.006	0.132 ±0.010	0.502 ±0.006	0.727 ±0.002	0.148 ±0.032	0.753 ±0.004	0.749 ±0.004	0.742 ±0.002
(39)(EnCTC)	0.104 ±0.001	0.494 ±0.010	0.490 ±0.008	0.143 ±0.011	0.505 ±0.007	0.727 ±0.003	0.185 ±0.035	0.749 ±0.004	0.741 ±0.004	0.743 ±0.003
(39)(LP+EnCTC)	0.087 ±0.002	0.611 ±0.007	0.570 ±0.012	0.267 ±0.006	0.586 ±0.011	0.742 ±0.008	0.427 ±0.049	0.773 ±0.004	0.773 ±0.005	0.761 ±0.007
(DP)	0.085 ±0.001	0.620 ±0.003	0.588 ±0.009	0.264 ±0.008	0.603 ±0.008	0.733 ±0.007	0.288 ±0.070	0.725 ±0.005	0.728 ±0.005	0.749 ±0.005
(DP)(LP)	0.107 ±0.002	0.491 ±0.012	0.466 ±0.014	0.153 ±0.013	0.479 ±0.012	0.708 ±0.007	0.306 ±0.047	0.743 ±0.005	0.737 ±0.007	0.727 ±0.006
(DP)(EnCTC)	0.084 ±0.001	0.623 ±0.002	0.584 ±0.002	0.254 ±0.006	0.598 ±0.002	0.741 ±0.004	0.383 ±0.031	0.730 ±0.005	0.734 ±0.005	0.754 ±0.003
(DP)(LP+EnCTC)	0.101 ±0.001	0.524 ±0.008	0.519 ±0.003	0.150 ±0.004	0.530 ±0.003	0.701 ±0.007	0.176 ±0.013	0.707 ±0.006	0.709 ±0.006	0.706 ±0.007
(DP)(39)	0.082 ±0.001	0.634 ±0.005	0.597 ±0.006	0.242 ±0.016	0.612 ±0.006	0.742 ±0.009	0.337 ±0.046	0.720 ±0.007	0.726 ±0.008	0.755 ±0.009
(DP)(39)(LP)	0.082 ±0.001	0.636 ±0.003	0.604 ±0.007	0.259 ±0.006	0.619 ±0.008	0.750 ±0.004	0.331 ±0.035	0.740 ±0.009	0.742 ±0.007	0.766 ±0.003
(DP)(39)(EnCTC)	0.083 ±0.002	0.632 ±0.004	0.590 ±0.010	0.232 ±0.016	0.605 ±0.009	0.748 ±0.006	0.261 ±0.033	0.737 ±0.009	0.743 ±0.010	0.764 ±0.008
(DP)(39)(LP+EnCTC)	0.084 ±0.001	0.628 ±0.003	0.593 ±0.009	0.231 ±0.012	0.608 ±0.008	0.736 ±0.006	0.287 ±0.039	0.725 ±0.015	0.730 ±0.014	0.751 ±0.008

Table 2: Evaluation results for HierCB (Yan et al., 2025) model.

set (pure phones) consistently outperformed their full-phoneme (stress-marked) counterparts across nearly all metrics. On TIMIT, WavLM(39)(LP) achieved a word TSE of 5.9% on the test set, compared to 6.6% for the standard WavLM(LP). This performance gain is likely due to the reduced confusion between acoustically similar phonemes, which simplifies the alignment task for the CTC loss.

3.4.2 Comparison of CTC Peaky Mitigation Methods

We evaluated two primary methods for mitigating the peaky behavior and overconfidence in CTC models, LP and EnCTC, respectively. LP ensures the ASR model remains fair to rare phonetic symbols by normalizing the posteriors. Our results indicate that while standard WavLM-CTC lags behind the Kaldi baseline in word TSE (9.3% vs. 9.0% on Buckeye), the integration of LP significantly closes this gap. As shown in Figure 1, WavLM(LP) achieves a word TSE of 7.3% on Buckeye, outperforming the Kaldi baseline by approximately 2% absolute. This suggests that accounting for the prior distribution of phones effectively mitigates the "lazy" alignment typical of standard CTC, leading to more precise boundary estimation.

Specifically, LP ensures the ASR model remains fair to rare phonetic symbols by normalizing the posteriors. This yields substantial gains in high-precision scenarios; for instance, the word ACC-50 on Buckeye increases from 44% (Standard CTC) to 69% with the addition of LP and phoneme level from 48% to 52%, shown in Table 3 and Table 4. In contrast, while EnCTC acts as a regularizer to ensure the ASR model remains cautious and avoids

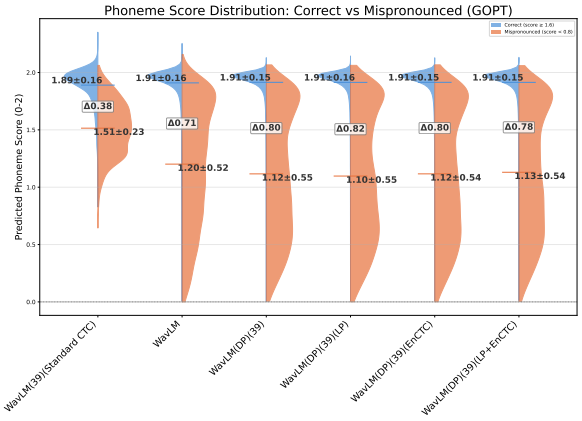

Figure 2: Distribution of GOPT phoneme scores for correct vs. mispronounced speech across various CTC-based ASR backbones.

overfitting to its own predictions, we observed that EnCTC is less effective than LP at reducing the BR.

Finally, we conducted experiments on the combination of both methods (LP+EnCTC). This hybrid approach yielded the best overall performance across all metrics. On the Buckeye dataset, WavLM(39)(LP+EnCTC) achieved the highest word ACC-50 score of 77.2%. These findings confirm that simultaneously addressing data bias (via LP) and model overconfidence (via EnCTC) is essential for achieving high-fidelity alignment suitable for downstream APA tasks.

3.4.3 DP vs. Regularizers

A notable observation is that LP still collapses on content-independent ASR models. Empirically, LP forces the model to predict rare phones more often. In a standard (monophone) CTC model, this might lead to insertion errors of rare phones into the blank spaces, destroying the alignment. However, in DP+39, the output space is more context-sensitive. The DP targets provide a stronger acoustic constraint that prevents LP from hallucinating rare symbols, instead using that prior to sharpen the transitions between phones.

3.4.4 Relational Studies: Alignment vs. APA Performance

We observed a clear correlation between TA accuracy and the PCC of the APA models. As shown in the comparison between Buckeye and TIMIT results, models with lower TSE (better alignment) consistently produced GOP features that led to higher word and utterance-level PCC in APA. For instance, the transition from standard WavLM-CTC to WavLM(39)(LP+EnCTC) reduced phone TSE from 26.3% to 25.2% on Buckeye, which directly contributed to more reliable GOP feature extraction.

ASR Models	Buckeye / TIMIT dev / TIMIT test		
	P-TSE↓	P-ACC@25↑	P-ACC@50↑
TDNN-F(Kaldi)	26.3/5.2/6.1	26.1/34.4/32.5	63.6/80.9/76.5
WavLM(39)($CTC^{\star}$)	26.1/5.6/6.7	27.4/**38.6**/**36.8**	54.9/73.2/68.8
WavLM (DWFST-CTC)			
-	26.3/7.1/8.2	21.0/25.9/24.9	48.9/59.2/56.0
(LP)	26.1/7.2/8.4	26.0/29.1/26.9	52.8/59.2/55.3
(EnCTC)	26.3/7.0/8.1	21.2/26.5/25.4	48.8/59.5/56.2
(LP+EnCTC)	26.0/7.0/8.1	27.1/30.4/28.0	54.2/61.0/56.3
(39)	26.0/6.9/8.0	23.5/27.1/26.2	51.1/61.1/57.6
(39)(LP)	25.3/6.5/7.6	32.8/33.1/30.7	59.0/64.9/60.1
(39)(EnCTC)	26.2/6.6/7.6	24.4/29.1/27.9	51.6/63.0/59.1
(39)(LP+EnCTC)	**25.2**/6.4/7.5	**32.9**/33.8/31.8	59.4/65.4/61.6
(DP)	38.2/6.8/6.9	9.7/22.4/23.5	26.9/57.4/57.9
(DP)(LP)	38.3/7.0/7.2	12.9/24.7/25.5	30.0/58.7/58.8
(DP)(EnCTC)	38.1/6.9/6.9	9.7/22.1/23.3	26.9/56.8/57.5
(DP)(LP+EnCTC)	38.5/7.0/7.1	9.5/22.3/22.7	26.6/56.5/57.2
(DP)(39)	38.2/6.8/7.0	9.9/22.1/23.3	27.1/57.4/58.0
(DP)(39)(LP)	38.2/6.8/7.0	9.8/22.6/23.5	27.0/57.5/58.1
(DP)(39)(EnCTC)	38.2/6.8/7.0	9.9/22.1/23.3	27.2/57.4/58.0
(DP)(39)(LP+EnCTC)	39.1/7.1/7.2	10.0/21.8/22.4	27.2/56.8/58.0

Table 3: A simplified result shows the TSE and ACC metrics on Buckeye corpus, TIMIT dev and test datasets, which only select the phoneme level (P-) and the tolerance at 25ms (@25) and 50ms (@50).

3.5 Scoring Results: Correct vs. Incorrect

We use the alignment-free APA model GOPT, which relies solely on CTC-GOP features, to test if our ASR changes actually help tell the difference between good and bad speech. As shown in Figure 2, the standard WavLM-CTC baseline has a scoring gap of 0.708. However, our best setup, WavLM(DP)(39)(LP), widens this gap to 0.816.

This 15% improvement in the scoring margin is a big deal for learners. While correct speech (score$\geq$ 1.6) gets high marks in all models, our new method is much better at catching mistakes. By fixing the peaky issue in CTC, the ASR model becomes more sensitive to the small acoustic errors that learners make. This makes the final scores more reliable and easier for a teacher to trust.

4 Discussion and Conclusion

Our results show that standard CTC's "blank" dominance and peaky behavior are major barriers to stable pronunciation scoring. We demonstrate that context-aware CTC, which combines peaky mitigation and OCD targets, overcomes these limits. Our WavLM(DP)(39) series with LP or EnCTC has

ASR Models	Buckeye / TIMIT dev / TIMIT test		
	W-TSE↓	W-ACC@25↑	W-ACC@50↑
TDNN-F(Kaldi)	9.3/6.2/6.3	29.3/27.4/28.3	72.1/74.2/72.8
WavLM(39)($CTC^{\star}$)	9.0/7.9/7.8	21.3/19.9/22.1	55.7/51.9/52.8
WavLM (DWFST-CTC)			
-	10.7/9.7/9.7	12.7/10.5/10.8	44.2/37.3/39.7
(LP)	7.3/6.5/6.6	38.9/37.4/33.7	69.8/61.8/60.0
(EnCTC)	10.7/9.7/9.7	12.7/11.1/11.0	43.9/37.3/39.8
(LP+EnCTC)	7.3/6.4/6.7	38.2/37.3/32.3	69.1/62.0/59.7
(39)	10.0/9.3/9.4	15.4/12.2/12.2	47.4/39.5/42.1
(39)(LP)	6.0/5.9/6.0	46.8/41.4/38.3	76.5/66.5/63.4
(39)(EnCTC)	9.9/9.1/9.0	16.5/13.0/14.0	48.4/41.4/43.2
(39)(LP+EnCTC)	**5.9/5.6/5.8**	**47.7/43.2/40.1**	**77.2**/67.4/65.7
(DP)	10.8/9.6/9.8	11.4/10.9/11.3	43.1/40.3/40.3
(DP)(LP)	7.9/7.7/7.9	29.6/24.6/22.4	63.9/54.7/54.8
(DP)(EnCTC)	10.9/9.7/9.8	11.3/10.8/11.2	42.9/40.0/39.9
(DP)(LP+EnCTC)	11.0/10.2/10.3	11.1/7.7/7.0	41.6/34.2/35.4
(DP)(39)	10.4/9.6/9.7	13.0/11.0/11.6	45.3/40.1/41.0
(DP)(39)(LP)	10.5/9.7/9.8	12.1/10.2/10.2	44.6/40.1/41.1
(DP)(39)(EnCTC)	10.4/9.6/9.7	13.0/11.0/11.6	45.3/40.2/41.0
(DP)(39)(LP+EnCTC)	10.6/9.8/10.0	12.7/9.0/9.0	44.2/36.7/37.5

Table 4: A simplified result shows the TSE and ACC on Buckeye corpus, TIMIT dev and test datasets, which only select the word level (W-) and the tolerance at 25ms (@25) and 50ms (@50).

now surpassed TDNN-F baseline. For APA, this soft and context-sensitive posterior provides a rich feature that captures the small acoustic details of a learner's mistakes.

In conclusion, the context-aware CTC framework fixes standard CTC's flaws for pronunciation tasks. We increased the scoring margin by 15% over the baseline. These findings prove that temporally dense and context-sensitive posteriors are essential for building reliable APA systems that provide stable, helpful feedback to learners.

5 Related Works

5.1 APA

APA systems evaluate second-language (L2) speech by providing fine-grained feedback across phoneme, word, and utterance levels (Gong et al., 2022; Yan et al., 2025). Most modern frameworks follow a two-stage pipeline: extracting segmental-level features and predicting scores through either unified (Gong et al., 2022; Chao et al., 2022) or hierarchical (Yan et al., 2025; Chao et al., 2023; Do et al., 2023) neural architectures. While unified models assess all granularities in a single decision, hierarchical models aggregate information from phoneme-level features up to the utterance level.

A fundamental building block in these models is the GOP feature, which measures the acoustic deviation between learner speech and native references (Gong et al., 2022). While early systems relied on traditional ASR-derived GOP, current state-of-the-art models increasingly leverage self-supervised learning (SSL) backends, such as Wav2vec 2.0 (Baevski et al., 2020), HuBERT (Hsu et al., 2021), and WavLM (Chen et al., 2022)—to extract robust contextual representations. These SSL models capture complex phonological patterns, significantly enhancing phoneme-level precision. In this work, we evaluate our proposed context-aware CTC-GOP features using both a unified architecture, GOPT (Gong et al., 2022), and a hierarchical model, HierCB (Yan et al., 2025).

Limitations

While the proposed method shows promise, several limitations remain. First, our evaluation is restricted to English as the target language with Mandarin L1 speakers. The efficacy of this approach for other language pairs and diverse L1 backgrounds requires further investigation.

Second, the transition from monophones to diphone units introduces a significant increase in the output layer dimensionality—from 39 to 39^2 possible combinations. This expansion increases computational overhead and may lead to data sparsity issues, as certain diphone transitions occur infrequently in training corpora.

Finally, our methodology is specifically designed to address the architectural constraints of CTC. Consequently, the adopted diphone-based context modeling may not be directly applicable or beneficial to non-CTC-based ASR frameworks, such as Transducer models.

Enthical Impact

We utilize publicly available and anonymized datasets (Librispeech, TIMIT, Buckeye, and speechocean762) for model training and evaluation. Our methodology focuses on improving speech technology for educational purposes and does not involve human subjects or the collection of private data. All authors comply with the ACL Code of Ethics and the relevant code of conduct for this venue.

References

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations.

Advances in Neural Information Processing Systems (NeurIPS), 33:12449–12460.

Xinwei Cao, Zijian Fan, Torbjørn Svendsen, and Giampiero Salvi. 2024. A framework for phoneme-level pronunciation assessment using CTC. In *Proc. of Interspeech*, pages 302–306.

Xinwei Cao, Zijian Fan, Torbjørn Svendsen, and Giampiero Salvi. 2026. Segmentation-free goodness of pronunciation. *IEEE Transactions on Audio, Speech and Language Processing (TASLP)*, 34:796–807.

Fu-An Chao, Tien-Hong Lo, Tzu-I Wu, Yao-Ting Sung, and Berlin Chen. 2022. 3M: An effective multi-view, multi-granularity, and multi-aspect modeling approach to English pronunciation assessment. In *Proc. of IEEE Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 575–582.

Fu-An Chao, Tien-Hong Lo, Tzu-I Wu, Yao-Ting Sung, and Berlin Chen. 2023. A hierarchical context-aware modeling approach for multi-aspect and multi-granular pronunciation assessment. In *Proc. of Interspeech*, pages 974–978.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, and 1 others. 2022. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing (JSTSP)*, 16(6):1505–1518.

Xianzhao Chen, Yist Y. Lin, Kang Wang, Yi He, and Zejun Ma. 2023. Improving frame-level classifier for word timings with non-peaky CTC in end-to-end automatic speech recognition. In *Proc. of Interspeech*, pages 2908–2912.

Jan Chorowski, Adrian Łańcucki, Bartosz Kostka, and Michał Zapotoczny. 2019. Towards using context-dependent symbols in CTC without state-tying decision trees. In *Proc. of Interspeech*, pages 4385–4389.

Heejin Do, Yunsu Kim, and Gary Geunbae Lee. 2023. Hierarchical pronunciation assessment with multi-aspect attention. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

John S Garofolo, Lori F Lamel, William M Fisher, David S Pallett, Nancy L Dahlgren, Victor Zue, and Jonathan G Fiscus. 1993. TIMIT acoustic-phonetic continuous speech corpus.

Yuan Gong, Ziyi Chen, Iek-Heng Chu, Peng Chang, and James Glass. 2022. Transformer-based multi-aspect multi-granularity non-native English speaker pronunciation assessment. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7262–7266.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proc. of the international conference on Machine learning (ICML)*, pages 369–376.

Awni Hannun, Vineel Pratap, Jacob Kahn, and Wei-Ning Hsu. 2020. Differentiable weighted finite-state transducers. *arXiv preprint arXiv:2010.01003*.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, 29:3451–3460.

Wenping Hu, Yao Qian, Frank K. Soong, and Yong Wang. 2015. Improved mispronunciation detection with deep neural network trained acoustic models and transfer learning based logistic regression classifiers. *Speech Communication*, 67:154–166.

Ruizhe Huang, Xiaohui Zhang, Zhaoheng Ni, Li Sun, Moto Hira, Jeff Hwang, Vimal Manohar, Vineel Pratap, Matthew Wiesner, Shinji Watanabe, Daniel Povey, and Sanjeev Khudanpur. 2024. Less peaky and more accurate CTC forced alignment by label priors. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 11831–11835.

Jiun-Ting Li, Bi-Cheng Yan, Yi-Cheng Wang, and Berlin Chen. 2025. Multi-task pretraining for enhancing interpretable L2 pronunciation assessment. In *Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 531–536.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *Proc. of International Conference on Learning Representations (ICLR)*.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: An asr corpus based on public domain audio books. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210.

Aditya Kamlesh Parikh, Cristian Tejedor-Garcia, Catia Cucchiarini, and Helmer Strik. 2025. Evaluating logit-based GOP scores for mispronunciation detection. In *Interspeech 2025*, pages 2405–2409.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, and 1 others. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems (NeuIPS)*, 32.

Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur. 2015. A time delay neural network architecture for efficient modeling of long temporal contexts. In *Proc. of Interspeech*, pages 3214–3218.

Hao-Chen Pei, Hao Fang, Xin Luo, and Xin-Shun Xu. 2024. Gradformer: A framework for multi-aspect multi-granularity pronunciation assessment. *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, 32:554–563.

Mark A Pitt, Keith Johnson, Elizabeth Hume, Scott Kiesling, and William Raymond. 2005. The buckeye corpus of conversational speech: Labeling conventions and a test of transcriber reliability. *Speech Communication*, 45(1):89–95.

Marius-Constantin Popescu, Valentina E Balas, Liliana Perescu-Popescu, and Nikos Mastorakis. 2009. Multilayer perceptron and neural networks. *WSEAS Transactions on Circuits and Systems*, 8(7):579–588.

Daniel Povey, Gaofeng Cheng, Yiming Wang, Ke Li, Hainan Xu, Mahsa Yarmohammadi, and Sanjeev Khudanpur. 2018. Semi-orthogonal low-rank matrix factorization for deep neural networks. In *Proc. of Interspeech*, pages 3743–3747.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukáš Burget, Ondřej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlíček, Yanmin Qian, Petr Schwarz, Jan Silovský, Georg Stemmer, and Karel Veselý. 2011. The kaldi speech recognition toolkit. In *Proc. of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE Signal Processing Society.

Rotem Rousso, Eyal Cohen, Joseph Keshet, and Eleanor Chodroff. 2024. Tradition or innovation: A comparison of modern ASR methods for forced alignment. In *Proc. of Interspeech*, pages 1525–1529.

Jiatong Shi, Nan Huo, and Qin Jin. 2020. Context-aware goodness of pronunciation for computer-assisted pronunciation training. In *Proc. of Interspeech*, pages 3057–3061.

S.M Witt and S.J Young. 2000. Phone-level pronunciation scoring and assessment for interactive language learning. *Speech Communication*, 30(2):95–108.

Bi-Cheng Yan, Jiun-Ting Li, Yi-Cheng Wang, Hsin Wei Wang, Tien-Hong Lo, Yung-Chang Hsu, Wei-Cheng Chao, and Berlin Chen. 2024. An effective pronunciation assessment approach leveraging hierarchical transformers and pre-training strategies. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1737–1747.

Bi-Cheng Yan, Yi-Cheng Wang, Jiun-Ting Li, Meng-Shin Lin, Hsin-Wei Wang, Wei-Cheng Chao, and Berlin Chen. 2025. ConPCO: Preserving phoneme characteristics for automatic pronunciation assessment leveraging contrastive ordinal regularization. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Zengwei Yao, Liyong Guo, Xiaoyu Yang, Wei Kang, Fangjun Kuang, Yifan Yang, Zeyu Jin, Long Lin, and Daniel Povey. 2024. Zipformer: A faster and better encoder for automatic speech recognition. In *Proc. of International Conference on Learning Representations (ICLR)*.

Albert Zeyer, Ralf Schlüter, and Hermann Ney. 2021. Why does CTC result in peaky behavior? *arXiv preprint arXiv:2105.14849*.

Junbo Zhang, Zhiwen Zhang, Yongqing Wang, Zhiyong Yan, Qiong Song, Yukai Huang, Ke Li, Daniel Povey, and Yujun Wang. 2021. speechocean762: An open-source non-native English speech corpus for pronunciation assessment. In *Proc. of Interspeech*, pages 3710–3714.

Zeyu Zhao and Peter Bell. 2024. Advancing CTC models for better speech alignment: A topological approach. In *Proc. of IEEE Spoken Language Technology Workshop (SLT)*, pages 279–285.

A CTC-GOP implementation

A.1 Phone Inventory Reduction

When the acoustic model operates over a context-dependent phone set (e.g., with stress markers such as `AA0`, `AA1`, `AA2`), the log-probability matrix must be reduced to a pure-phone inventory before GOP computation. This mirrors the `ComputeLpps()` procedure in the speechocean762 official implementation [2] (Zhang et al., 2021) .

Given a log-probability matrix $\mathrm{M} \in \mathbb{R}^{T\times C}$, where T is the number of frames and C is the full phone inventory size, and a surjective mapping $\phi : \{0,\ldots,C-1\} \to \{0,\ldots,K-1\}$ from tokens to pure phones, we compute the reduced log-probability matrix $\mathrm{M}' \in \mathbb{R}^{T\times K}$ as:

$$\mathrm{M}'_{t,k} = \log \sum_{j:\,\phi(j)=k} \exp(\mathrm{M}_{t,j}) \qquad (12)$$

To ensure numerical stability, we apply the grouped log-sum-exp (log-space marginalization) trick. For each frame t and pure phone k, let $m_{t,k} = \max_{j:\,\phi(j)=k} M_{t,j}$. Then:

$$\mathrm{M}'_{t,k} = m_{t,k} + \log \sum_{j:\,\phi(j)=k} \exp\big(\mathrm{M}_{t,j} - m_{t,k}\big) \qquad (13)$$

If no token maps to a particular pure phone k at frame t (i.e., the set $\{j : \phi(j) = k\}$ is empty), we define $M'_{t,k} = -\infty$. This reduction is implemented efficiently using `scatter_reduce` with the `amax` operator for computing $m_{t,k}$, and `scatter_add_` for the shifted exponential summation, avoiding explicit loops over the vocabulary.

[2] https://github.com/kaldi-asr/kaldi/blob/master/src/bin/compute-gop.cc

A.2 GOP Feature Computation

Let $\mathrm{Y}_c = (s_1, s_2, \ldots, s_N)$ denote the canonical phone sequence for a given utterance, where N is the number of phones. We define the *LPP* of the canonical sequence as:

$$lpp = -\mathcal{L}_{\mathrm{CTC}}(\mathrm{X}, \mathrm{Y}_c) \tag{14}$$

where $\mathcal{L}_{\mathrm{CTC}}$ denotes the CTC loss (i.e., the negative log-likelihood under the CTC formulation). It is notable that the input posterior to the CTC loss could select log posterior rather than pure probability one (Cao et al., 2026) to avoid numerical underflow errors when multiplying probabilities over longer speech segments.

For each phone position $i \in \{1, \ldots, N\}$, we construct a set of hypothesis sequences through two operations:

- **Deletion**: Remove the phone at position i, yielding $\mathrm{Y}_c^{(i,\mathrm{del})} = (s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_N)$.
- **Substitution**: Replace the phone at position i with every phone $q \in \{1, \ldots, K-1\}$, yielding $\mathrm{Y}_c^{(i,q)} = (s_1, \ldots, s_{i-1}, q, s_{i+1}, \ldots, s_N)$.

This produces $N \times K$ hypothesis sequences in total (one deletion and $K-1$ substitutions per position). All hypothesis sequences are batched into a single padded tensor and evaluated simultaneously via PyTorch(Paszke et al., 2019)'s `F.ctc_loss` with `reduction='none'`, enabling efficient GPU-parallel computation.

The **LPR** for position i and hypothesis q is defined as:

$$\mathbf{LPR}_{i,q} = \mathcal{L}_{\mathrm{CTC}}(\mathrm{X}, \mathbf{s}^{(i,q)}) - lpp \tag{15}$$

This quantity measures how much less likely the modified sequence is compared to the canonical pronunciation. A positive **LPR** for the substitution with the canonical phone at position i indicates correct pronunciation; values near zero or negative suggest mispronunciation.

A.3 Feature Representation

For each phone position i, the extracted feature vector is:

$$\begin{aligned}\mathbf{f}_i = \big[\, &\phi(s_i),\ lpp,\ \mathrm{LPR}_{i,0},\\ &\mathrm{LPR}_{i,1},\ \ldots,\ \mathrm{LPR}_{i,K-1}\,\big]\end{aligned} \tag{16}$$

where $\phi(s_i)$ is the pure phone identity and $\mathrm{LPR}_{i,0}$ corresponds to the deletion hypothesis. The resulting feature matrix $\mathbf{F} \in \mathbb{R}^{N \times (K+2)}$ is stored per utterance for downstream mispronunciation detection or scoring.

A.4 CTC Forward Algorithm

For reference, the CTC forward log-probability is computed via the standard dynamic programming recursion. Let $\mathbf{s}' = (\emptyset, s_1, \emptyset, s_2, \emptyset, \ldots, \emptyset, s_N, \emptyset)$ be the blank-augmented label sequence of length $L = 2N + 1$. The forward variable $\alpha_{l,t}$ represents the log-probability of emitting the partial sequence $\mathbf{s}'_{1:l}$ over frames 1 through t:

$$\alpha_{l,t} = \log p(\mathbf{s}'_{1:l},\, \mathrm{X}_{1:t}) \tag{17}$$

Let $\Psi(t-1, \mathcal{L}) = \log \sum_{l' \in \mathcal{L}} \exp(\alpha_{l',t-1})$. The recursion proceeds as:

$$\alpha_{l,t} = \begin{cases} \Psi(t-1, \mathcal{A}(l)) + \log p(\mathbf{s}'_l \mid t), \\ \qquad\qquad \text{if } l \bmod 2 = 0 \\ \Psi(t-1, \mathcal{B}(l)) + \log p(s_{\lfloor l/2 \rfloor} \mid t), \\ \qquad\qquad \text{otherwise} \end{cases}$$

where $\mathcal{A}(l) = \{l,\, l-1\}$ for blank positions, and $\mathcal{B}(l) = \{l,\, l-1\}$ when $s_{\lfloor l/2 \rfloor} = s_{\lfloor l/2 \rfloor - 1}$ (repeated label), or $\{l,\, l-1,\, l-2\}$ otherwise. The total sequence log-probability is obtained as:

$$\log p(\mathbf{s} \mid \mathrm{X}) = \mathrm{logaddexp}\big(\alpha_{L,T},\ \alpha_{L-1,T}\big) \tag{18}$$

A.5 Implementation Details

The pipeline operates as follows: (a) a pre-trained CTC acoustic model produces frame-level log-probabilities from audio; (b) canonical phone sequences are derived either from a lexicon-based word-to-phone expansion or from an explicit phoneme-level transcript; (c) GOP features are computed per utterance and saved as NumPy arrays. The batched CTC evaluation avoids the $\mathcal{O}(N \times K)$ sequential forward passes that would be required by a naïve implementation, yielding substantial speedup on GPU hardware.

B ASR and APA models training configures

In this work, we conduct experiments to train the ASR models on the Librispeech (Panayotov et al., 2015) corpus, a 960-hour training set. We utilize `torchaudio.bundle` to implement the backbone,

selecting WAVLM_LARGE as its initialized model parameters [3]. For some control groups, such as WavLM(39) and WavLM(DP)(39), they are based on WavLM and WavLM(DP), respectively, and further finetuned on the train-clean 100 hours training subset. In detail, we freeze all the WavLM encoder layers, and finetune only the output layers. For the DP ones, the output layer was designed with a single feedforward layer following two CD embedding layers (Chorowski et al., 2019) [4]. The feature extractor component of the WavLM model remains unchanged. Our models use phonemes as the modeling units. For the standard CTC, we adopt the implementation of Cao et al. (2024) [5], but change the backbone SSL model from wav2vec2.0 to WavLM. For the ASR model optimization, we employ the AdamW (Loshchilov and Hutter, 2019) optimizer, with a learning rate initialized from 10^{-4}.

Throughout our experimental setup, we utilize Zhao and Bell (2024) [6] project to conduct ASR experiments, in which Kaldi (Povey et al., 2011) for data preparation and PyTorch (Paszke et al., 2019) for neural network training, with k2-fsa [7] serving as the backend for the DWFST framework. The computation of CTC-GOP features follows the idea in Cao et al. (2024) but is further optimized for its stability and the running speed.

For the comparative methods, LP and EnCTC, we adopt the implementations [8] [9], adjusting them to fit in the implementation of computing loss in the k2-fsa (Yao et al., 2024) [10] framework. For LP, we change to a smart strategy, accumulating its label prior to each mini-batch to achieve a gradual adjustment. The a for the label prior weight is set to 0.2, while the b for the conditional maximum entropy weight is set to 0.1.

For GOPT and HierCB models training, we adhere to the hyperparameters from Gong et al. (2022); Yan et al. (2025) for training of our APA model. To quantify epistemic uncertainty, we train with five different random seeds.

[3] https://docs.pytorch.org/audio/stable/generated/torchaudio.pipelines.WAVLM_LARGE.html

[4] https://github.com/chorowski-lab/pytorch-asr/blob/master/att_speech/modules/decoders/advanced_decoder.py

[5] https://github.com/frank613/CTC-based-GOP

[6] https://github.com/ZhaoZeyu1995/BenNevis

[7] https://github.com/k2-fsa/k2

[8] https://github.com/huangruizhe/audio/blob/aligner_label_priors/examples/asr/librispeech_alignment/loss.py

[9] https://github.com/liuhu-bigeye/enctc.crnn/blob/master/pytorch_ctc/ctc_ent.py

[10] https://github.com/k2-fsa/k2