Mixture-of-Experts with Intermediate CTC Supervision for Accented Speech Recognition

Wonjun Lee[◊], **Hyounghun Kim**[◊,♦] **and Gary Geunbae Lee**[◊,♦]
[◊] Department of Computer Science and Engineering, POSTECH
[♦] Graduate School of Artificial Intelligence, POSTECH
{lee1jun, h.kim, gblee}@postech.ac.kr

Abstract

Accented speech remains a persistent challenge for automatic speech recognition (ASR), as most models are trained on data dominated by a few high-resource English varieties, leading to substantial performance degradation for other accents. Accent-agnostic approaches improve robustness yet struggle with heavily accented or unseen varieties, while accent-specific methods rely on limited and often noisy labels. We introduce MoE-CTC, a Mixture-of-Experts architecture with intermediate CTC supervision that jointly promotes expert specialization and generalization. During training, accent-aware routing encourages experts to capture accent-specific patterns, which gradually transitions to label-free routing for inference. Each expert is equipped with its own CTC head to align routing with transcription quality, and a routing-augmented loss further stabilizes optimization. Experiments on the MCV-ACCENT benchmark demonstrate consistent gains across both seen and unseen accents in low- and high-resource conditions, achieving up to 29.3% relative WER reduction over strong FastConformer baselines.

1 Introduction

Automatic speech recognition (ASR) has reached near-human accuracy on benchmarks such as LibriSpeech (Panayotov et al., 2015) and Switchboard, driven by large-scale pretraining and advanced architectures (Baevski et al., 2020; Rekesh et al., 2023). However, these advances do not generalize uniformly: performance drops markedly on accented speech, where acoustic and phonetic patterns differ from those dominant in training data. Because accented speech remains underrepresented in most corpora, current ASR systems under-perform for accented speakers, raising concerns of fairness and inclusivity (Shor et al., 2019; DHERAM et al., 2022; Graham and Roll, 2024).

Prior research has mainly followed two directions. **Accent-agnostic approaches** aim to learn robust, accent-invariant representations through self-supervised or adversarial objectives (Hsu et al., 2021; Das et al., 2021a). While models such as XLS-R (Babu et al., 2022) and Whisper (Radford et al., 2023) improve general robustness, their performance still degrades for heavily accented or unseen varieties, indicating that full invariance is insufficient to capture accent diversity (Zuluaga-Gomez et al., 2023).

Accent-specific approaches, on the other hand, leverage accent labels for explicit specialization via fine-tuning (Li et al., 2021), data augmentation (Do et al., 2024), or parameter-efficient modules such as adapters and accent embeddings (Tomanek et al., 2021; Jain et al., 2018). These methods improve performance for known accents but depend on accent labels even at inference, limiting scalability and generalization to unseen speakers.

A scalable alternative lies in **Mixture-of-Experts (MoE)**, which allows experts to specialize across domains. Since accent is an utterance-level attribute, sequence-level MoE provides a natural fit for accented ASR. Recent studies have explored MoE for multilingual or dialectal ASR (You et al., 2021; Jie et al., 2024; Bagat et al., 2025), yet existing methods either require accent labels at test time or fail to reliably route inputs to the appropriate experts.

To address these challenges, we propose MoE-CTC, a novel architecture that integrates MoE with expert-level intermediate CTC (Connectionist Temporal Classification) supervision. Our method learns accent-aware routing during training but transitions to accent-agnostic inference through a two-stage learning process. Each expert is equipped with an auxiliary CTC head, aligning routing decisions with transcription quality and improving optimization stability.

Experiments on the MCV-ACCENT bench-

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 26015–26027

mark (Prabhu et al., 2023) demonstrate that MoE-CTC consistently outperforms prior accent-robust baselines, achieving substantial WER reductions across both seen and unseen accents.

2 Related Work

2.1 Accented Speech Recognition

Accented speech has long been recognized as a major source of performance degradation in ASR systems. Analyses on corpora such as L2-ARCTIC (Zhao et al., 2018), CommonVoice (Ardila et al., 2020), and GLOBE (Wang et al., 2024) consistently show that models trained primarily on native speech exhibit substantially higher error rates for accented speakers (Zuluaga-Gomez et al., 2023; Shor et al., 2019). Even large-scale pretrained models such as XLS-R and Whisper reduce but do not eliminate the accent gap: Whisper shows higher WER on non-native accents (Graham and Roll, 2024), while XLS-R/Whisper performance degrades on African-accented datasets (Olatunji et al., 2023).

Accent-agnostic Approaches. One research strand aims to improve robustness without relying on accent labels. Self-supervised pretraining with wav2vec 2.0 (Baevski et al., 2020), HuBERT (Hsu et al., 2021), and related models has proven effective at learning accent-invariant representations. Similarly, large-scale multilingual pretraining (e.g., XLS-R, Whisper) improves cross-accent generalization. Other works adopt adversarial or domain generalization techniques—such as domain adversarial training (Sun et al., 2018; Das et al., 2021a), relabeling strategies (Hu et al., 2020), or coupled/contrastive learning (Unni et al., 2020)—to enforce invariance. While these methods improve robustness overall, they still under-perform on heavily accented or unseen varieties compared to accent-aware models.

Accent-specific Approaches. Another line of work explicitly incorporates accent supervision. Fine-tuning on accent-labeled corpora (Li et al., 2021) often boosts performance for the target accent but compromises generalization to unseen varieties. Data augmentation synthesizes accented speech via phoneme perturbations or accent conversion (Do et al., 2024), though gains are limited by coverage and realism. Parameter-efficient adaptations such as residual adapters (Tomanek et al., 2021), non-linear modules (Qian et al., 2022), and accent embeddings (Jain et al., 2018; Chen et al., 2015; Li et al., 2017; Viglino et al., 2019) offer lighter-weight alternatives. Rapid cross-accent adaptation has also been explored for low-resource settings (Rao and Sak, 2017; Winata et al., 2020). Recently, Bagat et al. (2025) organized accent-specific LoRA modules as a mixture of experts, showing that modular parameterization can balance specialization and scalability.

2.2 Mixture-of-Experts for Accent Adaptation

While accent-specific methods achieve strong specialization, they depend on labeled accent data and generalize poorly to unseen accents. MoE offers a scalable alternative: by selectively activating specialized subnetworks, MoE increases model capacity without proportional computational cost (Shazeer et al., 2017; Fedus et al., 2022). In ASR, MoE has been applied to multilingual and multi-domain recognition (You et al., 2021; Hu et al., 2023), where routing mechanisms distribute inputs across experts under diverse acoustic conditions.

A key challenge for accent-aware MoE is routing when accent labels are unavailable at inference. To address this, Prabhu et al. (2023, 2024) introduced beam search–based expert (codebook) selection, while Jie et al. (2024) proposed dialect-adaptive dynamic routing using feature–embedding combinations. Bagat et al. (2025) presented MAS-LoRA, where accent-specific LoRA modules act as experts; averages all LoRA experts equally when accent labels are unavailable.

In contrast, our MoE-CTC introduces accent-aware routing during early training to encourage expert specialization, then transitions to accent-agnostic training for inference. This avoids auxiliary routing strategies, requires no accent labels at test time, and enables effective generalization to unseen accents.

3 Background

Our proposed framework, MoE-CTC, is built upon a FastConformer (Rekesh et al., 2023) encoder with the CTC head for ASR, augmented with two key ideas: *intermediate CTC supervision* and the *MoE* architecture. Intermediate CTC provides auxiliary objectives that stabilize optimization and enhance representation learning, while MoE introduces capacity through expert specialization. In the following subsections, we review these two components to clarify the foundation of our approach.

3.1 Intermediate CTC Loss

Connectionist Temporal Classification (CTC) (Graves et al., 2006) is a widely adopted training criterion for non-autoregressive ASR models. By marginalizing over all valid frame-to-label alignments, CTC enables end-to-end training without the need for frame-level annotations.

Recent studies have shown that inserting intermediate CTC losses (Lee and Watanabe, 2021) at multiple encoder layers can improve training stability and accelerate convergence (Komatsu et al., 2022; Hojo et al., 2024). These auxiliary objectives encourage hidden representations at lower layers to be predictive of output units, thereby mitigating vanishing gradients and facilitating optimization.

Formally, let $h^{(\ell)}$ denote the encoder states at layer ℓ. An auxiliary CTC loss $\mathcal{L}^{(\ell)}_{\text{CTC}}$ is computed, and the overall training objective is

$$\mathcal{L} = \mathcal{L}^{(L)}_{\text{CTC}} + \sum_{\ell \in \mathcal{I}} \lambda_\ell \, \mathcal{L}^{(\ell)}_{\text{CTC}}, \tag{1}$$

where L is the final encoder layer, $\mathcal{I}$ is the set of intermediate layers with auxiliary supervision, and λ_ℓ are weighting coefficients.

3.2 Sequence-level Mixture-of-Experts

Mixture-of-Experts (MoE) (Shazeer et al., 2017; Fedus et al., 2022) augments deep models with sparsely activated expert networks. A gating function routes inputs to a subset of experts, enabling large capacity without proportional computation.

Conventional MoE performs token-level routing, where each audio frame in ASR may be directed to different experts, incurring high computation and switching overhead. In contrast, sequence-level MoE assigns an entire utterance to one or a few experts, reducing cost while preserving specialization. This is particularly suitable for accented speech, since accent is an utterance-level attribute.

Given encoder input $H \in \mathbb{R}^{B \times T \times D}$, where B is the batch size, T the number of audio frames, and D the hidden dimension, we mean-pool across time:

$$\bar{h} = \tfrac{1}{T} \sum_{t=1}^{T} H_{:,t,:}. \tag{2}$$

The routing network maps this pooled representation $\bar{h} \in \mathbb{R}^{B \times D}$ to logits $L \in \mathbb{R}^{B \times N}$, where N is the number of experts and $L_{i,j}$ is the logit for assigning sample i to expert j. The gating probabilities are then obtained via softmax:

$$g_{i,j} = \frac{\exp(L_{i,j})}{\sum_{k=1}^{N} \exp(L_{i,k})}$$

Finally, the MoE output for sample i is the weighted sum of expert outputs:

$$\textsc{MoE}(H_i) = \sum_{j=1}^{N} g_{i,j} \, \textsc{Expert}_j(H_i).$$

Here, each $\textsc{Expert}_j$ is implemented as a feed-forward block, which we describe in detail in the following subsection.

4 Method

4.1 Mixture-of-Experts Module

We extend the FastConformer encoder by inserting MoE modules between encoder layers (ℓ-th), rather than replacing the internal feed-forward networks as in Hu et al. (2023). Each MoE module consists of N parallel experts, where each expert is a two-layer feed-forward block with ReLU activation:

$$\textsc{Expert}_j(H) = \text{FFN}_{2,j}\big(\text{ReLU}(\text{FFN}_{1,j}(H))\big),$$

with $\text{FFN}_{1,j}, \text{FFN}_{2,j} \in \mathbb{R}^{D \times D}$ and D the hidden dimension.

The routing network maps the pooled encoder representation $\bar{h}_i$ of sample i (Eq. 2) to a distribution $g_i \in \mathbb{R}^N$ over N experts. To promote efficiency, we adopt top-K routing: only the K experts with the highest probabilities $g_{i,j}$ are selected, and their weights are renormalized:

$$\tilde{g}_{i,j} = \frac{g_{i,j}}{\sum_{k \in \mathcal{K}_i} g_{i,k}}, \quad j \in \mathcal{K}_i,$$

where $\mathcal{K}_i \subseteq \{1, \ldots, N\}$ denotes the selected top-K experts for sample i. The final MoE output for sample i is then the weighted combination:

$$\textsc{MoE}(H_i) = \sum_{j \in \mathcal{K}_i} \tilde{g}_{i,j} \, \textsc{Expert}_j(H_i).$$

4.2 Accent-Aware Routing

Building on the generic MoE module, we introduce **Accent-MoE**, which incorporates explicit accent supervision. During training, each utterance is associated with an accent label $a_i \in \{0, 1, \ldots, A-1\}$. We align each accent to a designated expert, encouraging balanced utilization and accent-specific specialization. For example, align j-th expert for the Scotland accent (a_i).

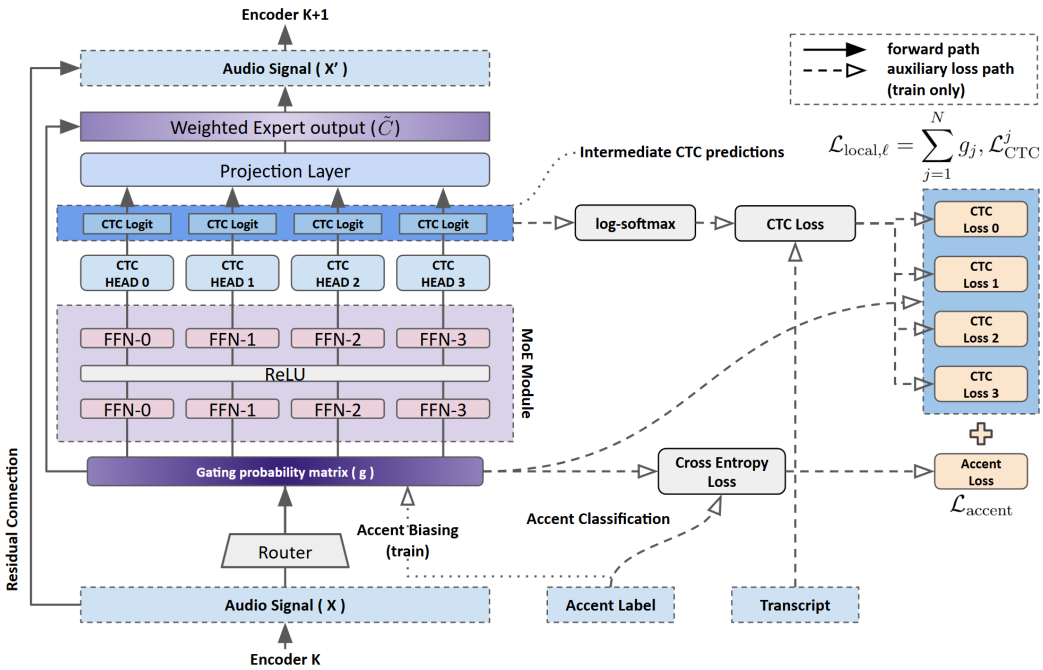

Figure 1: Overview of the proposed MoE-CTC architecture. The ℓ-th MoE module, inserted between encoder blocks, is illustrated with four experts (shown as an example). Each expert is equipped with an auxiliary CTC head, producing local supervision loss $\mathcal{L}_{\text{local},\ell}$. During training, the router incorporates accent-aware routing through Accent Biasing and Accent Classification loss ($\mathcal{L}_{\text{accent}}$).

Formally, let $L_{i,j}$ denote the gating logit for sample i and expert j, and a_i its accent index. We define the accent-biased logit:

$$\tilde{L}_{i,j} = L_{i,j} + \alpha \cdot \mathbf{1}[j = a_i], \tag{3}$$

where α is the bias strength and $\mathbf{1}[\cdot]$ is the indicator function. The final routing distribution is then

$$g_{i,j} = \frac{\exp(\tilde{L}_{i,j})}{\sum_{k=0}^{N-1} \exp(\tilde{L}_{i,k})}.$$

To enable the router to select appropriate experts even without accent labels at test time, the gating weights are also interpreted as accent logits and trained with an auxiliary loss

$$\mathcal{L}_{\text{accent}} = -\sum_{i} \log \frac{\exp(g_{i,a_i})}{\sum_{j=0}^{N-1} \exp(g_{i,j})}. \tag{4}$$

By applying the biasing term, we guide the model to route samples to accent-specific experts—so that each expert is primarily trained for its designated accent—while the auxiliary classification loss further regularizes the routing process.

However, explicit accent supervision alone may not align routing with recognition quality, so we introduce expert-level CTC supervision to directly couple expert selection with transcription accuracy.

4.3 Expert-Level CTC Supervision (MoE-CTC)

Inspired by intermediate CTC loss (Section 3.1), we extend this principle to ACCENT-MOE by equipping each expert with its own auxiliary CTC head, as illustrated in Figure 1. This design provides fine-grained supervision that guides both layer-wise representations and expert-level specialization, a strategy we denote as **MOE-CTC**.

Let $X \in \mathbb{R}^{B \times T \times D}$ denote the input to the MoE module, and $\text{EXPERT}_j(X)$ the output of the j-th expert. Each expert is equipped with a CTC head that maps hidden states into the vocabulary space:

$$\text{CTC-HEAD} : \mathbb{R}^{B \times T \times D} \rightarrow \mathbb{R}^{B \times T \times V},$$

where V is the vocabulary size of CTC-HEAD. Each expert-specific head yields an auxiliary CTC loss $\mathcal{L}_{\text{CTC}}^{(\ell,j)}$ at layer ℓ for expert j, evaluated against the ground-truth transcription.

Following the intermediate CTC self-conditioning strategy of Nozaki and Komatsu (2021), the logits from each expert's CTC head are projected back into the hidden dimension through a learnable linear mapping,

$$\text{PROJ} : \mathbb{R}^{B \times T \times V} \rightarrow \mathbb{R}^{B \times T \times D},$$

then combined according to the gating weights and added via a residual pathway:

$$
\begin{aligned}
H_j &= \textsc{Expert}_j(X), \\
C_j &= \textsc{Ctc-Head}_j(H_j), \\
P_j &= \textsc{Proj}(C_j), \\
\tilde{C} &= \sum_{j=1}^{N} g_j \, P_j \in \mathbb{R}^{B \times T \times D}, \\
X' &= X + \tilde{C}.
\end{aligned}
$$

Here, the auxiliary CTC losses are computed directly from C_j, while the residual pathway ensures that these CTC-informed signals are stably integrated back into the shared representation. In this way, expert behavior is aligned with transcription quality and CTC-informed feedback is propagated across layers.

Finally, to further couple expert performance with routing, we define a local routing-augmented objective:

$$
\mathcal{L}_{\text{local}} = \sum_{\ell=1}^{L} \sum_{j=1}^{N} g_j^{(\ell)} \, \mathcal{L}_{\text{CTC}}^{(\ell,j)} \tag{5}
$$

where $\mathcal{L}_{\text{CTC}}^{(\ell,j)}$ denotes the CTC loss from expert j in layer ℓ, and $g_j^{(\ell)}$ is its routing probability, analogous to the intermediate CTC objective (Eq. 1). By minimizing this loss, the router is encouraged to assign higher weights to experts that yield lower CTC loss, thereby improving routing quality and enhancing overall recognition performance.

The final training objective combines the global CTC loss, the accent-aware auxiliary loss, and the expert-level CTC supervision:

$$
\mathcal{L} = \mathcal{L}_{\text{CTC}} + \beta \, \mathcal{L}_{\text{local}} + \gamma \, \mathcal{L}_{\text{accent}}, \tag{6}
$$

where β and γ are weighting coefficients.

In this way, expert-level CTC supervision directly links routing to transcription quality, encouraging experts to specialize in improving ASR accuracy rather than only modeling accent distinctions.

4.4 Accent-Agnostic Training

To stabilize routing and improve generalization, we adopt a two-stage training strategy. The first stage focuses on specialization: both ACCENT-MOE and MOE-CTC are trained with accent-aware routing (Figure 1, Section 4.2), where the router is guided by an explicit biasing term and an auxiliary accent classification loss to encourage expert alignment with specific accents.

The second stage promotes generalization by removing these accent-specific signals, allowing the router to autonomously select experts. In this phase, ACCENT-MOE is optimized solely with the global CTC objective, while MOE-CTC continues to leverage both global and expert-level CTC supervision ($\mathcal{L}_{\text{local}}$). This staged design enables the model to first learn accent-specific expertise and then generalize to unseen accents by aligning expert selection with recognition quality.

5 Experimental Setup

5.1 Datasets

We conduct experiments on the **MCV-ACCENT** benchmark (Prabhu et al., 2023), which is derived from the English portion of the Mozilla CommonVoice corpus (Ardila et al., 2020). CommonVoice is a large-scale crowd sourced dataset covering diverse speakers and accents worldwide. The MCV-ACCENT split provides two training subsets (100h and 600h), along with corresponding development and test sets. Summary statistics are presented in Table 6 and Table 7 in the Appendix.

The training and development sets include five *seen* accents—Australia, Canada, England, Scotland, and the United States—while the test set introduces nine additional *unseen* accents to evaluate cross-accent generalization.

To simulate realistic training strategies, all models are first pretrained on 960 hours of English speech dataset **LIBRISPEECH-960H** (Panayotov et al., 2015) and then fine-tuned on MCV-ACCENT, following a common pipeline where large-scale standard English pretraining enhances downstream recognition of accented speech.

5.2 Models

All experiments are implemented using the NeMo framework (Kuchaiev et al., 2019) and trained on eight NVIDIA A100 GPUs (80GB each).

Baseline Models. We first build baseline FastConformer encoders of three sizes—*Small*, *Medium*, and *Large*—as summarized in Table 8. To ensure fair comparison with our MoE-based architectures, we also include intermediate FastConformer variants with approximately **46M** and **76M** parameters, matching the parameter scale of the MoE extensions.

Configuration			Seen Accent						Unseen Accent									
encoder size	model	parameter size	AUS	CAN	UK	SCT	US	Seen average	AFR	HKG	IND	IRL	MAL	NWZ	PHL	SGP	WLS	Unseen average
Small	FastConformer	12.78M	11.6	15.5	15.1	11.4	13.1	13.3	20.0	27.4	29.5	22.5	30.8	18.0	27.1	34.1	17.1	25.2
	Inter-CTC	12.78M	11.3	14.3	15.1	11.0	13.2	13.0	19.4	26.6	29.3	21.4	30.1	18.1	26.9	33.8	16.9	24.7
	MoE	13.72M	9.8	12.7	12.9	8.6	11.5	11.1	16.4	24.4	25.3	19.5	26.5	15.3	23.2	31.2	16.2	22.0
	ACCENT MOE	13.72M	8.5	11.2	12.2	7.5	10.8	10.0	16.0	23.8	24.6	18.9	26.8	14.8	22.9	30.5	15.8	21.6
	MOE-CTC	16.62M	8.2	10.5	11.4	6.6	10.1	9.4	15.7	22.8	23.2	18.3	25.5	13.7	22.5	29.6	15.2	20.7
Medium	FastConformer	26.39M	7.3	11.4	10.3	5.7	10.0	8.9	15.4	22.6	22.8	16.5	24.3	12.8	21.4	28.6	14.1	19.8
	Inter-CTC	26.39M	7.3	11.2	10.4	5.1	9.8	8.8	15.5	22.8	22.9	16.1	23.8	12.2	21.1	27.3	13.7	19.5
	MOE	28.37M	7.0	10.3	9.7	5.0	9.9	8.4	15.5	22.4	22.1	16.2	23.6	11.9	20.4	26.4	12.9	19.0
	ACCENT MOE	28.37M	6.3	9.3	8.5	4.7	8.0	7.4	13.8	21.3	20.0	15.5	21.3	11.1	19.9	24.9	10.7	17.6
	MOE-CTC	32.58M	5.9	8.7	8.4	4.3	7.2	6.9	12.9	19.4	18.3	14.8	21.1	10.7	18.9	24.2	7.9	16.5
Large	FastConformer	115.60M	5.5	9.4	7.8	3.9	7.7	6.9	13.3	19.8	18.5	15.6	23.6	10.8	19.2	24.2	10.4	17.3
	Inter-CTC	115.60M	5.6	9.2	7.9	4.1	7.5	6.9	12.9	19.5	17.9	15.3	23.8	10.5	18.3	23.3	9.9	16.8
	MOE	123.48M	5.3	8.8	7.3	4.0	6.8	6.4	12.5	19.9	17.8	14.5	19.5	8.8	17.4	20.6	7.5	15.4
	ACCENT MOE	123.48M	4.9	7.9	6.8	3.6	6.1	5.9	12.4	16.3	17.0	13.0	18.3	8.2	15.2	20.2	7.0	14.2
	MOE-CTC	131.90M	4.4	7.8	6.1	3.2	5.9	5.5	11.4	12.2	16.8	12.6	15.1	7.7	12.6	17.9	6.3	12.5

Table 1: WER (%) on the MCV-ACCENT-TEST set, comprising 5 seen and 9 unseen accents. All models are pretrained on LIBRISPEECH-960H and subsequently fine-tuned on MCV-ACCENT-100H. Highlighted cells indicate the best WER within each encoder size group.

MoE-Based Models. On top of the baseline Conformers, we additionally build three MoE variants: a standard **MoE** without accent supervision, an accent-aware **ACCENT-MOE**, and an expert-level supervised **MOE-CTC** (Table 9).

Each MoE model contains $L = 3$ MoE layers inserted at the 4th, 8th, and 12th encoder blocks, with $N = 5$ experts per layer matching the number of *seen accents*. Unless otherwise specified, all MoE variants share the same default settings: $K = 2$ for Top-K expert selection, accent prior strength $\alpha = 2$, local CTC loss coefficient $\beta = \frac{1}{2\cdot(L\times N)}$, and accent classification loss weight $\gamma = 0.1$.

Tokenizer and Decoding. We employ a 1024-token BPE vocabulary trained on MCV-ACCENT-100H. For decoding, we apply greedy CTC decoding without any external language model or beam search.

5.3 Training Configuration

All models are trained with a global batch size of 1024 using mixed-precision (FP16) training. During both LIBRISPEECH-960H pretraining and MCV-ACCENT fine-tuning, checkpoints are saved at the end of each epoch, and the best model is selected based on the lowest validation WER, with a maximum of 500 epochs. We employ the AdamW optimizer with an initial learning rate of 1×10^{-4} for pretraining and 1×10^{-5} for fine-tuning, using a cosine annealing schedule with warmup.

For ACCENT-MOE and MOE-CTC, we use a two-stage training strategy (Section 4.4). Models are first trained with accent-aware routing, and the best checkpoint from this stage is then fine-tuned for 20 additional epochs without accent supervision. The final model is selected based on the best validation WER from this second stage.

6 Results

6.1 Main Result

Table 1 presents the Word Error Rate (WER) of all models across seen and unseen accents under various encoder sizes (Table 8, 9). Across all settings, the proposed MOE-CTC achieves the lowest WER, demonstrating clear improvements over the FastConformer baselines. With the ***Small*** encoder, it yields relative WER reductions (WERR) of **29.3%** on seen and **17.3%** on unseen accents, and with the ***Large*** encoder, **20.3%** and **27.8%**, respectively. Notably, the gains on unseen accents increase with model capacity, suggesting that our approach generalizes more effectively as encoder size grows.

To further examine the effect of intermediate supervision, we introduce an *Inter-CTC* variant that adds intermediate CTC losses to the FastConformer baseline. This variant yields consistent yet modest improvements, confirming that intermediate supervision stabilizes training and enhances recognition quality.

Beyond this baseline, we compare the MOE and ACCENT-MOE variants, which share identical parameter sizes but differ in whether accent-aware supervision is applied. Across small, medium, and large encoders, ACCENT-MOE achieves average WERRs of 9.9%, 11.9%, and 7.8% on seen accents, and 1.8%, 7.4%, and 7.8% on unseen accents, re-

model	parameter	Seen ALL	Unseen ALL
Conformer (Gulati et al., 2020) †	43M	14.0	23.7
MTL (Zhang et al., 2021) †	43M	14.1	23.7
DAT (Das et al., 2021b) †	43M	14.0	23.4
Prabhu et al. (2023) †	46M	13.6	22.9
FastConformer	46.89M	13.8	23.7
MoE-CTC	46.91M	**12.7**	**22.3**

Table 2: Weighted average WER (%) on the MCV-ACCENT-TEST set. All models are trained on the **MCV-ACCENT-100H** training split. † indicates results reported by Prabhu et al. (2023).

model	parameter	Seen ALL	Unseen ALL
HuBERT (Hsu et al., 2021) †	74M	3.87	9.49
MTL (Zhang et al., 2021) †	74M	3.76	9.37
DAT (Das et al., 2021b) †	74M	3.83	9.30
Prabhu et al. (2024) †	76M	3.80	9.19
FastConformer	76.70M	3.53	9.52
MoE-CTC	76.26M	**3.10**	**8.65**

Table 3: Weighted average WER(%) on the MCV-ACCENT-TEST set. All models are trained on the **MCV-ACCENT-600H** training split. † indicates results reported by Prabhu et al. (2024).

spectively—highlighting the consistent advantage of accent-aware routing.

Building on these results, MoE-CTC demonstrates the most robust and generalizable performance, outperforming ACCENT-MoE by an additional **6.5%** WERR on seen and **7.5%** on unseen accents across all encoder sizes. These findings suggest that expert-level CTC supervision enables MoE-CTC to further enhance recognition accuracy across both seen and unseen accents.

To isolate the effect of model capacity, we control for parameter size in subsequent analyses. For fairness, we also compare MoE-CTC and FastConformer under equal parameter budgets, along with prior MCV-ACCENT benchmark results, as detailed in the following subsection.

6.2 Benchmark

To compare with prior studies on MCV-ACCENT, we evaluate two benchmark settings. The first uses a 46M-parameter model trained solely on the MCV-ACCENT-100H split without Librispeech pretraining, following Prabhu et al. (2023) (as shown in Table 2). The second employs a 76M-parameter model trained on the larger MCV-ACCENT-600H split, also without pretraining, following Prabhu et al. (2024) (Table 3).

The *Seen-ALL* and *Unseen-ALL* scores are calculated as weighted averages based on the number of samples per accent (Table 7), which differs from the unweighted averages reported in Table 1.

As shown in Table 2, removing LIBRISPEECH-960H pretraining leads to higher WER across all models. Our proposed MoE-CTC still outperforms other baselines and Prabhu et al. (2023), although the improvement is relatively modest, achieving a WERR of **6.6%** on seen and **2.6%** on unseen accents. When compared to our FastConformer baseline, the gains are slightly larger (8.0% on seen and 5.9% on unseen), but remain smaller than those observed in Table 1. This indicates that training with only 100 hours of data is insufficient to fully exploit the advantages of the proposed method.

In contrast, utilizing the larger MCV-ACCENT-600H training set yields a substantial performance improvement. As shown in Table 3, MoE-CTC outperforms previous best model (Prabhu et al., 2024) with larger WERR margins of **18.4%** on seen and **5.9%** on unseen accents.

7 Analysis

7.1 Effect of Accent-Agnostic Training

We further investigate the impact of accent-agnostic training (Section 4.4) for both ACCENT-MoE and MoE-CTC under the *Large* encoder configuration. Table 4 presents the results across different training stages.

For ACCENT-MoE, the accent-agnostic only training corresponds to the standard MoE baseline reported in Table 1. Introducing an initial accent-aware stage yields consistent gains on both seen and unseen accents (6.4 → 6.1 and 15.4 → 15.1, respectively). When followed by accent-agnostic fine-tuning, the model achieves further improvements, reducing WER to **5.9** (↓ 3.8%) on seen and **14.2** (↓ 6.0%) on unseen accents. These results confirm that accent-agnostic training helps the router generalize beyond explicit accent supervision.

A similar trend is observed in MoE-CTC, where incorporating accent-agnostic fine-tuning after accent-aware pretraining leads to a more pronounced improvement—achieving **5.5** (↓ 5.2%) on seen and **12.5** (↓ 12.0%) on unseen accents. This indicates that our expert-level CTC supervision is especially effective when combined with accent-agnostic training, allowing the model to retain accent sensitivity while improving generalization to unseen accents.

Model	Training Stage	Seen Avg.	Unseen Avg.
MoE	Accent-agnostic only	6.4	15.4
ACCENT-MOE	Accent-aware	6.1	15.1
	+Accent-agnostic	5.9 (↓ 3.8%)	14.2 (↓ 6.0%)
MOE-CTC	Accent-aware	5.8	14.2
	+Accent-agnostic	5.5 (↓ 5.2%)	12.5 (↓ 12.0%)

Table 4: Average WER(%) for seen and unseen accents under different training stages. Green text indicates WERR compared to the accent-aware stage.

7.2 Accent Routing Performance

While Zuluaga-Gomez et al. (2023) report 95% accuracy on a 16-way accent classification task using w2v2-XLSR (315M) trained on the CommonAccent dataset, their model is optimized solely for accent identification. In contrast, our router is jointly trained with ASR, making the task considerably more challenging. Under this multitask setting, the 76M MOE-CTC model (in Table 3), trained on 600 hours of data, attains 72% top-1 routing accuracy on the final MoE layer before accent-agnostic training. A similar observation was made by Prabhu et al. (2023), who found that auxiliary accent classifiers for expert selection offered limited benefit over inference-time beam search, suggesting that perfect accent discrimination is neither achievable nor necessary within ASR training.

After accent-agnostic fine-tuning, the router in MOE-CTC still preserves meaningful accent organization, achieving 66.2% top-1 accuracy (Figure 2). The confusion matrix reveals that Canadian speech is often routed to the U.S. expert and Scottish to the England expert, indicating that routing aligns with acoustically and geographically correlated accents even without explicit supervision. This adaptive redistribution promotes better generalization by guiding expert selection toward recognition-oriented specialization (Section 7.1).

7.3 Oracle Accent Routing Setting

To estimate the upper bound of our architecture under ideal conditions, we conduct an oracle experiment where ground-truth accent labels are provided at inference time. Specifically, we enforce hard expert routing by applying the biasing term (Eq. 3) with maximum strength ($\alpha = \infty$), ensuring that each accent exclusively activates its designated expert. As shown in Table 5, providing oracle labels yields the lowest WER of **5.2%** under accent-aware training and **5.3%** when combined with accent-agnostic fine-tuning, compared

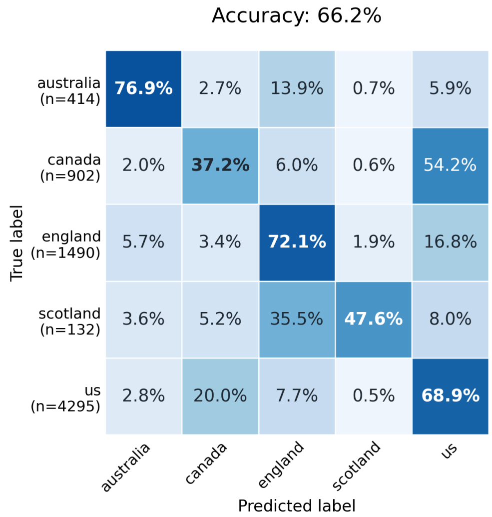

Figure 2: Matrix of router gating probabilities ($g_{i,j}$) at the final (*3rd*) MOE-CTC module of the 76M model trained on MCV-ACCENT-600H. Each row represents an accent, and the probabilities in each row sum to $\sum_{j=1}^{N=5} g_{i,j} = 1$.

model	Training Stage	Accent Label	Seen Average
MoE-CTC	Accent-aware	No	5.8 (5.82)
MoE-CTC	Both	No	5.5 (5.48)
MoE-CTC	Accent-aware	Yes	**5.2** (5.20)
MoE-CTC	Both	Yes	5.3 (5.32)

Table 5: WER(%) under the oracle accent routing setting with *Large* encoder. Parentheses denote values reported to two decimal places.

to **5.8%** and **5.5%** without label access. These results confirm that oracle routing achieves the best performance by leveraging accent-specific specialization, while accent-agnostic training slightly reduces specialization but improves generalization for label-free inference.

8 Conclusion

We presented MOE-CTC, a Mixture-of-Experts architecture enhanced with intermediate CTC supervision for accent-robust ASR. By combining expert-level CTC heads with a two-stage transition from accent-aware to accent-agnostic training, the model achieves relative WER reductions of up to 29.3% and 27.8% on seen and unseen accents, respectively, on the MCV-ACCENT benchmark. These results highlight that aligning expert routing with transcription quality substantially improves both robustness and generalization in accented speech recognition.

Limitations

While MoE-CTC demonstrates substantial improvements in accented speech recognition, its sequence-level routing assumes discrete accent boundaries, which may not generalize to mixed or code-switched speech. The model also relies on accent labels during early training, limiting applicability in unsupervised settings. Although sparsely activated, the additional MoE modules increase training cost and latency. Finally, experiments are limited to English accents on MCV-ACCENT; broader multilingual evaluation and analysis of expert interpretability remain future work.

Acknowlegment

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.RS-2019-II191906, Artificial Intelligence Graduate School Program(POSTECH)) (10%)

This research was supported by Culture, Sports and Tourism R&D Program through the Korea Creative Content Agency grant funded by the Ministry of Culture, Sports and Tourism in 2025 (Project Name: Development of an AI-Based Korean Diagnostic System for Efficient Korean Speaking Learning by Foreigners, Project Number: RS-2025-02413038, Contribution Rate: 45%)

This research was supported by Smart HealthCare for Police Officers Program(www.kipot.or.kr) through the Korea Institutes of Police Technology(KIPoT) funded by the Korean National Police Agency(KNPA, Korea)(No. RS-2022-PT000186) (45%)

References

Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. 2020. Common voice: A massively-multilingual speech corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4218–4222, Marseille, France. European Language Resources Association.

Arun Babu, Changhan Wang, Andros Tjandra, Kushal Lakhotia, Qiantong Xu, Naman Goyal, Kritika Singh, Patrick von Platen, Yatharth Saraf, Juan Pino, and 1 others. 2022. Xls-r: Self-supervised cross-lingual speech representation learning at scale. In *Proc. Interspeech 2022*, pages 2278–2282.

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*, 33:12449–12460.

R. Bagat, I. Illina, and E. Vincent. 2025. Mixture of LoRA experts for low-resourced multi-accent automatic speech recognition. In *Proceedings of Interspeech 2025*, pages 1143–1147.

Mingming Chen, Zhanlei Yang, Jizhong Liang, Yanpeng Li, and Wenju Liu. 2015. Improving deep neural networks based multi-accent mandarin speech recognition using i-vectors and accent-specific top layer. In *INTERSPEECH*.

Nilaksh Das, Sravan Bodapati, Monica Sunkara, Sundararajan Srinivasan, and Duen Horng Chau. 2021a. Best of both worlds: Robust accented speech recognition with adversarial transfer learning. In *INTERSPEECH*.

Nilaksh Das, Sravan Bodapati, Monica Sunkara, Sundararajan Srinivasan, and Duen Horng Chau. 2021b. Best of both worlds: Robust accented speech recognition with adversarial transfer learning. In *Proc. Interspeech 2021*, pages 1314–1318.

PRANAV DHERAM, Murugesan Ramakrishnan, Anirudh Raju, I-Fan Chen, Brian King, Katherine Powell, Melissa Saboowala, Karan Shetty, and Andreas Stolcke. 2022. Toward fairness in speech recognition: Discovery and mitigation of performance disparities. In *Proc. Interspeech 2022*, pages 1268–1272.

Cong-Thanh Do, Shuhei Imai, Rama Doddipatla, and Thomas Hain. 2024. Improving accented speech recognition using data augmentation based on unsupervised text-to-speech synthesis. In *2024 32nd European Signal Processing Conference (EUSIPCO)*, pages 136–140. IEEE.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39.

Calbert Graham and Nathan Roll. 2024. Evaluating openai's whisper asr: Performance analysis across diverse accents. *JASA Express Letters*.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376.

A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang. 2020. Conformer: Convolution-augmented transformer for speech recognition. In *Proceedings of Interspeech 2020*, pages 5036–5040.

K. Hojo, Y. Wakabayashi, K. Ohta, A. Ogawa, and N. Kitaoka. 2024. Boosting CTC-based ASR using inter-layer attention-based CTC loss. In *Proceedings of Interspeech 2024*, pages 2860–2864.

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*, 29:3451–3460.

Hu Hu, Xuesong Yang, Zeynab Raeesy, Jinxi Guo, Gokce Keskin, Harish Arsikere, Ariya Rastrow, Andreas Stolcke, and Roland Maas. 2020. Redat: Accent-invariant representation for end-to-end asr by domain adversarial training with relabeling. In *INTERSPEECH*.

Ke Hu, Bo Li, Tara Sainath, Yu Zhang, and Françoise Beaufays. 2023. Mixture-of-expert conformer for streaming multilingual asr. In *Proc. Interspeech 2023*, pages 3327–3331.

Abhinav Jain, Minali Upreti, and Preethi Jyothi. 2018. Improved accented speech recognition using accent embeddings and multi-task learning. In *Interspeech*, pages 2454–2458.

Zhou Jie, Gao Shengxiang, Yu Zhengtao, Dong Ling, and Wang Wenjun. 2024. Dialectmoe: An end-to-end multi-dialect speech recognition model with mixture-of-experts. In *Proceedings of the 23rd Chinese National Conference on Computational Linguistics (Volume 1: Main Conference)*, pages 1148–1159.

T. Komatsu, Y. Fujita, J. Lee, L. Lee, S. Watanabe, and Y. Kida. 2022. Better intermediates improve CTC inference. In *Proceedings of Interspeech 2022*, pages 4965–4969.

Oleksii Kuchaiev, Jason Li, Huyen Nguyen, Oleksii Hrinchuk, Ryan Leary, Boris Ginsburg, Samuel Kriman, Stanislav Beliaev, Vitaly Lavrukhin, Jack Cook, and 1 others. 2019. Nemo: a toolkit for building ai applications using neural modules. *arXiv preprint arXiv:1909.09577*.

Jaesong Lee and Shinji Watanabe. 2021. Intermediate loss regularization for ctc-based speech recognition. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6224–6228. IEEE.

Bo Li, Tara N Sainath, Khe Chai Sim, Michiel Bacchiani, Eugene Weinstein, Patrick Nguyen, Zhifeng Chen, Yonghui Wu, and Kanishka Rao. 2017. Multi-dialect speech recognition with a single sequence-to-sequence model. In *ICASSP*, pages 4749–4753.

Jialu Li, Vimal Manohar, Pooja Chitkara, Andros Tjandra, Michael Picheny, Frank Zhang, Xiaohui Zhang, and Yatharth Saraf. 2021. Accent-robust automatic speech recognition using supervised and unsupervised wav2vec embeddings. *Preprint*, arXiv:2110.03520.

Jumon Nozaki and Tatsuya Komatsu. 2021. Relaxing the conditional independence assumption of ctc-based asr by conditioning on intermediate predictions. In *Proc. Interspeech 2021*, pages 3735–3739.

T. Olatunji, T. Afonja, B. F. P. Dossou, and 1 others. 2023. Afrispeech-200: Pan-african accented speech dataset for asr. *Transactions of the Association for Computational Linguistics (TACL)*.

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

D. Prabhu, A. Gupta, O. Nitsure, P. Jyothi, and S. Ganapathy. 2024. Improving self-supervised pre-training using accent-specific codebooks. In *Proceedings of Interspeech 2024*, pages 2310–2314.

Darshan Prabhu, Preethi Jyothi, Sriram Ganapathy, and Vinit Unni. 2023. Accented speech recognition with accent-specific codebooks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7175–7188, Singapore. Association for Computational Linguistics.

Yanmin Qian, Xun Gong, and Houjun Huang. 2022. Layer-wise fast adaptation for end-to-end multi-accent speech recognition. *IEEE/ACM TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING*, 30.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *Proceedings of the 40th International Conference on Machine Learning*, pages 28492–28518.

Kanishka Rao and Haşim Sak. 2017. Multi-accent speech recognition with hierarchical grapheme based models. In *ICASSP*, pages 4815–4819.

Dima Rekesh, Nithin Rao Koluguri, Samuel Kriman, Somshubra Majumdar, Vahid Noroozi, He Huang, Oleksii Hrinchuk, Krishna Puvvada, Ankur Kumar, Jagadeesh Balam, and 1 others. 2023. Fast conformer with linearly scalable attention for efficient speech recognition. In *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1–8. IEEE.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*.

Joel Shor, Dotan Emanuel, Oran Lang, Omry Tuval, Michael Brenner, Julie Cattiau, Fernando Vieira, Maeve McNally, Taylor Charbonneau, Melissa Nollstadt, and 1 others. 2019. Personalizing asr for dysarthric and accented speech with limited data. In *Proc. Interspeech 2019*, pages 784–788.

Sining Sun, Ching-Feng Yeh, Mei-Yuh Hwang, Mari Ostendorf, and Lei Xie. 2018. Domain adversarial training for accented speech recognition. In *ICASSP*, pages 4854–4858.

Katrin Tomanek, Vicky Zayats, Dirk Padfield, Kara Vaillancourt, and Fadi Biadsy. 2021. Residual adapters for parameter-efficient ASR adaptation to atypical and accented speech. In *Proceedings of the 2021*

Conference on Empirical Methods in Natural Language Processing, pages 6751–6760, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Vinit Unni, Nitish Joshi, and Preethi Jyothi. 2020. Coupled training of sequence-to-sequence models for accented speech recognition. In *ICASSP*, pages 8254–8258.

Thibault Viglino, Petr Motlíček, and Milos Cernak. 2019. End-to-end accented speech recognition. In *INTERSPEECH*.

Wenbin Wang, Yang Song, and Sanjay Jha. 2024. Globe: A high-quality english corpus with global accents for zero-shot speaker adaptive text-to-speech. In *Proc. Interspeech 2024*, pages 1365–1369.

Genta Indra Winata, Samuel Cahyawijaya, Zihan Liu, Zhaojiang Lin, Andrea Madotto, Peng Xu, and Pascale Fung. 2020. Learning fast adaptation on cross-accented speech recognition. *arXiv preprint arXiv:2003.01901*.

Zhao You, Shulin Feng, Dan Su, and Dong Yu. 2021. Speechmoe: Scaling to large acoustic models with dynamic routing mixture of experts. In *Proc. Interspeech 2021*, pages 2077–2081.

Jicheng Zhang, Yizhou Peng, Van Tung Pham, Haihua Xu, Hao Huang, and Eng Siong Chng. 2021. E2e-based multi-task learning approach to joint speech and accent recognition. In *Proc. Interspeech 2021*, pages 1519–1523.

G. Zhao, S. Sonsaat, A. Silpachai, I. Lucic, E. Chukharev-Hudilainen, J. Levis, and R. Gutierrez-Osuna. 2018. L2-ARCTIC: A non-native english speech corpus. In *Proceedings of Interspeech 2018*, pages 2783–2787.

Juan Zuluaga-Gomez, Sara Ahmed, Danielius Visockas, and Cem Subakan. 2023. Commonaccent: Exploring large acoustic pretrained models for accent classification based on common voice. In *Proc. Interspeech 2023*, pages 5291–5295.

A Appendix

A.1 MCV-ACCENT Dataset

For English text normalization, we adopt the whisper-normalizer[1] toolkit to ensure consistent text preprocessing across both the training (Table 6) and evaluation (Table 7) sets.

A.2 Model Configuration Details

Table 8 summarizes the configurations of the FastConformer baselines, including the total number of parameters, encoder layers, hidden dimension, and multi-head attention heads. The corresponding configurations for MOE-CTC are presented in Table 9. The *Small*, *Medium*, and *Large* variants share identical FastConformer settings as in Table 8, with additional MoE modules inserted between encoder blocks. For the 46M and 76M variants, the base FastConformer encoders are slightly downsized—by adjusting the number of layers and hidden dimensions—to ensure comparable total parameter counts after adding the MoE layers.

[1] https://kurianbenoy.github.io/whisper_normalizer/

Accent	MCV-ACCENT-100H		MCV-ACCENT-600H	
	Count	Duration (h)	Count	Duration (h)
US	46,385	64.08	289,177	399.88
England	14,590	19.51	90,081	119.93
Australia	5,018	6.95	31,355	43.56
Canada	4,815	6.79	29,436	41.13
Scotland	1,669	2.69	10,123	16.21

Table 6: Comparison of accent distribution and duration in MCV-ACCENT-100H and MCV-ACCENT-600H subsets from Prabhu et al. (2023).

Accent	MCV-ACCENT-DEV		MCV-ACCENT-TEST	
	Count	Duration (h)	Count	Duration (h)
US	7,050	8.32	4,295	4.87
England	2,775	3.22	1,490	1.65
Australia	3,004	4.33	414	0.46
Canada	966	1.16	902	1.21
Scotland	233	0.23	132	0.16
New Zealand	–	–	1,620	2.11
Ireland	–	–	1,423	1.94
African	–	–	1,213	1.71
Philippines	–	–	631	0.90
Indian	–	–	492	0.58
Singapore	–	–	477	0.64
Hongkong	–	–	409	0.52
Malaysia	–	–	262	0.39
Wales	–	–	219	0.27

Table 7: Statistics of validation (MCV-ACCENT-DEV) and test (MCV-ACCENT-TEST) sets.

Name	Params	Layers	d_model	Heads
Small	12.78M	16	176	4
Medium	26.39M	16	256	4
46M	46.89M	18	324	4
76M	76.70M	18	416	4
Large	115.60M	18	512	8

Table 8: FastConformer configurations with parameter size, depth, hidden dimension (d_{model}), and multi-head attention heads.

Name	Params	Layers	d_model	Heads
Small	16.62M (12.78M)	16	176	4
Medium	32.58M (26.39M)	16	256	4
46M	**46.91M** (38.85M)	**16**	**312**	4
76M	**76.26M** (65.51M)	**18**	**384**	4
Large	131.90M (115.60M)	18	512	8

Table 9: MoE-CTC model configurations. Notation follows Table 8. Values in parentheses indicate the base FastConformer parameters, excluding MoE-CTC modules.

Model	Number of Experts	Parameter size	Seen average	Unseen average
ACCENT-MOE	5	123.48M	5.86	14.18
ACCENT-MOE	8	129.89M	**5.83** (↓ 0.5%)	14.12 (↓ 0.4%)
MOE-CTC	5	131.90M	5.48	12.51
MOE-CTC	8	141.37M	**5.39** (↓ 1.6%)	**12.28** (↓ 1.9%)

Table 10: WER(%) comparison between 5- and 8-expert variants. Green text denotes relative WER reduction (WERR) of 8-expert over 5-expert models.

A.3 Adding Spare Experts

By default, we set the number of experts to 5, matching the number of seen accents for accent-aware routing. In the *Large* configuration, we further increase the capacity by adding three additional experts, resulting in a total of 8 experts. During training, these extra experts are randomly assigned to approximately 20% of the training samples from all seen accents, encouraging exposure to diverse acoustic patterns without explicit bias. As shown in Table 10, increasing the number of experts consistently improves recognition performance. For ACCENT-MOE, expanding from 5 to 8 experts reduces WER from 5.86 to 5.83 (**0.5%** relative) on seen accents and from 14.18 to 14.12 (**0.4%**) on unseen accents. The improvement is more pronounced for MOE-CTC, achieving a WER reduction from 5.48 to 5.39 (**1.6%**) on seen and from 12.51 to 12.28 (**1.9%**) on unseen accents. These consistent gains indicate that the additional experts act as global experts, learning accent-invariant representations that complement accent-specific specialization and enhance overall generalization.

A.4 CTC-HEAD sharing

MOE-CTC (Figure 1) equips each expert with its own CTC-HEAD, enabling expert-specific alignment but also increasing the overall parameter count. To examine more lightweight alternatives, we explore several CTC-HEAD sharing strategies.

Model	CTC-Head Sharing	Parameter size	Seen average	Unseen average
MOE-CTC	Full separation	131.90M	**5.48**	**12.51**
	Layer-wise sharing	125.59M	5.60 (↑2.2%)	13.07 (↑4.5%)
	Global sharing	124.01M	5.92 (↑8.0%)	14.52 (↑16.1%)
ACCENT-MOE	No Head	123.48M	5.86	14.18

Table 11: WER(%) comparison of different CTC head sharing strategies in MoE-CTC. Red text indicates relative WER degradation compared to the full separation baseline.

As summarized in Table 11, *Full separation* denotes the default configuration where each expert has an independent head. In the *Layer-wise sharing* setup, a single CTC-HEAD is shared among all experts within the same MoE layer, while *Global sharing* removes intermediate heads entirely, using only the final global CTC head for all experts. Layer-wise sharing slightly degrades WER on both seen (+2.2%) and unseen (+4.5%) accents, yet it remains competitive and far superior to ACCENT-MOE while reducing parameters by about 5%. In contrast, global sharing results in a more substantial degradation (+8.0% on seen, +16.1% on unseen), even worse than ACCENT-MOE with no expert-specific head. We hypothesize that each MoE layer produces representations of varying transcription quality—earlier layers generate coarse alignments, while deeper layers refine them—thus separate heads at each layer help preserve alignment fidelity and stabilize expert specialization.

A.5 MOE Module Placement

We use three MOE modules by default, balancing modest parameter growth with sufficient capacity for expert specialization. All experiments in this subsection are conducted with the *Large* MOE-CTC encoder, which has 131.90M parameters. Relative to the 115.60M-parameter FastConformer baseline, adding these modules results in an overhead of about 14% (Table 9). Based on pilot experiments, we insert the MOE modules after encoder layers 4, 8, and 12, thereby introducing specialization at the early, middle, and late stages of the 16–18-layer FastConformer encoder.

We further examine alternative placements in a small ablation study, summarized in Table 12. Placing all three MOE modules in the early layers (2, 4, 6), middle layers (6, 8, 10), or late layers (10, 12, 14) is consistently less effective than the balanced 4–8–12 configuration. Among the single-region variants, middle-layer insertion yields the best generalization to both seen and unseen ac-

MoE insertion	**Indices**	**Seen average**	**Unseen average**
Early	(2, 4, 6)	6.0	14.2
Middle	(6, 8, 10)	5.6	12.7
Late	(10, 12, 14)	5.6	13.0
Balanced	**(4, 8, 12)**	**5.5**	**12.5**
Last†	(4, 8, **18**)	6.2	15.5

Table 12: Ablation of MOE insertion positions in MOE-CTC. All variants use the *Large* encoder configuration (131.90M parameters), with three MOE modules differing only in their insertion layers. † indicates that the last MOE module is placed at the final encoder layer.

cents, whereas the balanced configuration performs best overall. As shown in Table 12, moving the last MOE module to the final encoder layer further degrades performance, suggesting that several conventional encoder blocks should remain after the last MOE module to support robust decoding. These findings motivate our default placement at layers 4, 8, and 12.