Studies on Context-Aware Speech Recognition and Multimodal Spoken Language Understanding

Jiajun He

Abstract

End-to-end (E2E) automatic speech recognition (ASR) has made tremendous progress in recent years, fueled by advances in deep neural networks, attention mechanisms, and large-scale pretraining. Modern ASR systems, ranging from Transformer-based architectures to foundation speech-language models that couple speech encoders with large language models (LLMs), now achieve near-human performance on frequent words in open-domain benchmarks. Despite this success, their accuracy deteriorates substantially on long-tailed rare and domain-specific words. These words often include named entities such as personal names, locations, and organizations, as well as technical terms and domain-specific expressions that are underrepresented in training corpora. Although infrequent, they typically carry crucial semantic content in real-world applications. Misrecognition of a drug name in a medical dictation system or failure to capture a company name in a meeting transcription can severely undermine the utility of ASR outputs.

To address this issue, this dissertation provides a systematic study of rare-word robustness in ASR and develops three complementary solution families: context-aware ASR post-correction, E2E contextual ASR, and LLM-based ASR for complex acoustic and contextual scenarios.

First, we introduce a family of error detection and context-aware error correction models that operate as lightweight post-processors on ASR hypotheses. The key idea is to localize likely error positions and apply targeted correction using external contextual knowledge such as lists of named entities and technical terms, thus avoiding full-sentence redecoding. To further handle homophones, we design a phoneme-augmented multimodal fusion variant that injects phonetic cues to disambiguate orthographically distinct but phonetically similar candidates. We also propose a retention probability mechanism to filter low-confidence edits, reducing overdetection and preserving correct tokens. Experiments across five datasets show that this approach consistently reduces biased word error rate while maintaining competitive inference latency.

Second, we move contextualization inside the recognizer by designing rare-word-aware E2E models. The proposed phoneme-augmented robust contextual ASR with contrastive entity disambiguation (PARCO) addresses two fundamental limitations of prior biasing methods: weak phonetic discrimination for domain entities and incomplete multi-token entity retrieval caused by token-level biasing. PARCO integrates phoneme-aware encoding, entity-level supervision, hierarchical entity filtering, and a contrastive objective that separates confusable entities under uncertainty. Under 5,000 distractors, PARCO achieves strong improvements on both Chinese and English benchmarks, and shows robust generalization to out-of-domain corpora.

Third, we explore LLM-based ASR to handle realistic conditions with overlapping speakers and large biasing lists. We propose optimized finetuning strategies and a two-stage rare-word filtering pipeline that

prunes large biasing lists and injects the survivors into LLM prompts. Even under a biasing size of 5,000, the system still achieves remarkable performance on LibriMix and AMI, surpassing traditional contextual biasing approaches in complex acoustic scenes.

Beyond transcription accuracy, this thesis further investigates the impact of ASR on downstream understanding tasks. For multimodal speech emotion recognition, we develop fusion architectures that jointly leverage acoustic cues and ASR text while explicitly modeling ASR uncertainty. We integrate auxiliary ASR error detection and correction modules with multimodal fusion, and introduce adversarial and contrastive strategies to learn disentangled and modality-invariant affective representations. For speech-driven multimodal video moment retrieval, we propose a pointer-based reading framework that combines an audio–visual encoder for coarse video/moment granularity with a 2D pointer module for sharp temporal boundary localization. Experiments show substantial improvements on benchmark datasets such as IEMOCAP, MELD, and HiREST.

The empirical scope of this thesis spans a wide range of benchmark datasets, including ATIS and SNIPS for spoken language understanding (SLU), DATA2, LibriSpeech, and PRLVS for rare-word and open-domain ASR, and AISHELL-1 and THCHS-30 for Mandarin ASR. To evaluate robustness in multi-talker conditions, we employ LibriMix and AMI, which focus on overlapping and conversational meeting speech. For affective interaction, we use IEMOCAP and MELD, which provide multimodal annotations for emotion recognition. Finally, to explore cross-modal retrieval, we adopt the HiREST dataset for speech-driven video moment retrieval. Together, these datasets cover diverse application domains such as task-oriented dialogue, audiobook reading, domain-specific terminology, English and Mandarin ASR, conversational meetings with speaker overlap, multimodal affective understanding, and speech-to-video semantic alignment. Results reveal consistent trends: post-correction with phoneme-aware cues complements E2E contextualization; entity-level supervision and contrastive disambiguation are critical in homophone- and distractor-rich settings; LLMs, when paired with principled bias filtering, extend contextual ASR to multi-talker scenarios; and explicitly modeling ASR errors within downstream fusion prevents catastrophic error propagation and yields robust gains.

In summary, this dissertation advances contextual ASR for long-tailed vocabularies and demonstrates how such advances translate to resilient speech-driven applications. The methods are architecture-agnostic and compatible with contemporary E2E systems, offering a practical route to deploy rare-word-aware ASR and multimodal SLU in real-world settings.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background and Motivation

Automatic speech recognition (ASR) refers to the process of transcribing spoken language into written text in an automatic manner, as shown in Fig. 1.1. In recent years, significant advances have been achieved through the development of end-to-end (E2E) architectures, such as connectionist temporal classification (CTC), recurrent neural network transducer (RNN-T), and attention-based encoder–decoder (AED) models, together with the emergence of large-scale pretrained models [2]. These innovations have facilitated the practical deployment of ASR in diverse applications, ranging from intelligent personal assistants and transcription services to multimodal speech–language understanding systems. Nevertheless, despite these advances, ASR continues to face several long-standing challenges that limit its robustness and generalization in real-world scenarios.

One fundamental problem is the accurate recognition of rare and domain-specific words, such as named entities, technical terms, or personal names. These words are often underrepresented or absent in training corpora, leading to high substitution or deletion error rates [3]. Errors on such tokens are particularly harmful because they typically carry the most salient semantic information, and their misrecognition propagates downstream, negatively affecting applications such as summarization, named entity recognition, or dialogue understanding [4, 5]. Addressing this issue requires methods that can leverage contextual knowledge, handle phonetic ambiguities, and remain robust across domains and noise conditions.

A second line of challenges arises in multi-talker and conversational ASR. In natural communication scenarios such as meetings or customer service calls, multiple speakers may overlap, switch rapidly, or introduce out-of-domain terminology. Classical approaches, such as permutation-invariant training (PIT) [6] or serialized output training (SOT) [7], attempt to resolve speaker assignment am-

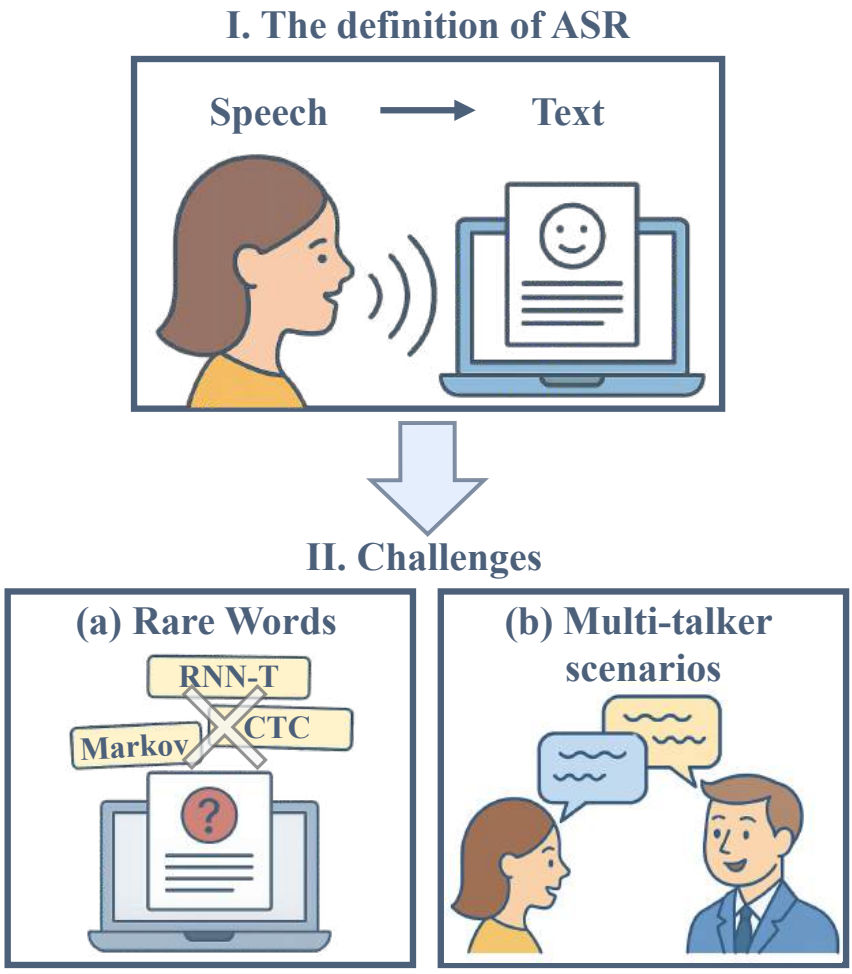


Figure 1.1: Definition and Challenges of ASR

biguities but struggle with long-context dependencies and rare-word recognition. These limitations highlight the need for ASR systems that not only separate overlapping speech but also integrate contextual biasing mechanisms to ensure accurate recognition of critical terms.

Equally important is the impact of ASR errors on spoken language understanding (SLU) tasks. Many downstream tasks rely directly on ASR transcripts as inputs, including speech emotion recognition (SER) and video moment retrieval (VMR), as shown in Fig. 1.2. In SER, emotionally salient words misrecognized by ASR can severely degrade emotion classification performance [8, 9]. In VMR, language queries derived from ASR transcripts are used to align with multimodal video content; ASR inaccuracies, especially on rare or descriptive terms, reduce retrieval precision [10, 11].

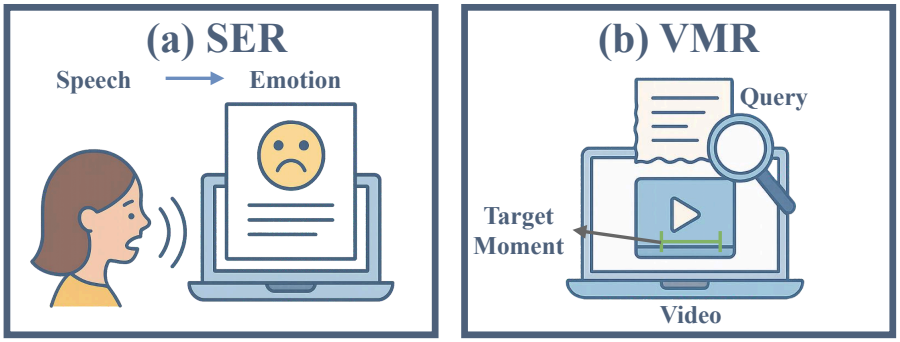


Figure 1.2: Definition of SER and VMR

Consequently, improving the robustness of ASR outputs and developing multimodal methods resilient to recognition errors are both essential for advancing SLU.

Taken together, these challenges motivate research on context-aware and multimodal speech recognition and understanding. The integration of contextual biasing, multimodal fusion, and advanced training strategies provides a promising pathway toward systems that are not only accurate at the ASR level but also robust in downstream tasks. This dissertation is driven by the need to bridge these gaps and to design unified frameworks that advance rare-word recognition, improve robustness in multi-talker scenarios, and enhance performance in multimodal understanding tasks such as SER and VMR.

1.2 Problem Definition

This dissertation focuses on the challenge of rare-word recognition in ASR and its cascading effects on downstream tasks. Rare words refer to lexical items that appear infrequently in training corpora but are semantically critical, such as named entities, technical jargon, or emerging terms. Accurately recognizing these words is essential to ensure that ASR outputs preserve the meaning of spoken content in real-world applications, including meeting transcription, dialogue systems, and multimodal understanding.

The difficulty arises primarily from two factors. First, data sparsity prevents ASR models from learning robust representations of rare words, as they occur too infrequently in training sets. Second, phonetic ambiguity causes rare words to be easily confused with acoustically similar frequent words, resulting in substitution errors. These challenges are further exacerbated under noisy conditions, domain shifts, or overlapping multi-talker scenarios, making rare-word recognition a fundamental bottleneck in ASR.

Beyond ASR itself, recognition errors significantly affect downstream applications. For multi-

modal SER, errors in ASR transcripts distort semantic cues and degrade emotion classification accuracy. To address this issue, auxiliary tasks such as ASR error detection (AED) and ASR error correction (AEC) can be integrated into SER frameworks, improving semantic coherence and robustness against transcription noise. Similarly, in multimodal VMR, ASR provides the textual modality that complements video features, and its quality directly influences the alignment between natural language queries and candidate video segments.

In summary, this dissertation addresses the challenge of rare-word recognition in ASR and investigates its broader implications for multimodal speech and language understanding.

1.3 Research Objectives

The primary goal of this dissertation is to enhance the robustness of ASR systems with respect to rare and domain-specific words, and to examine how ASR performance interacts with downstream multimodal and speech-driven tasks. In particular, this work investigates (1) how the quality of ASR transcripts affects the reliability of downstream SER, where transcription errors can distort semantic cues and degrade emotional inference, and (2) whether integrating speech-based representations derived from ASR can further improve the performance of VMR by providing complementary auditory context. The specific objectives are:

- To design postprocessing approaches that detect and correct errors in ASR outputs by leveraging contextual and phonetic information.
- To develop E2E contextual ASR models that integrate rare-word biasing mechanisms directly into the recognition process.
- To explore the use of LLMs for multi-talker and contextual ASR, combining speech encoders with prompt-based LLM adaptation.
- To examine the impact of ASR on downstream tasks, particularly multimodal SER and speech-driven multimodal VMR.

1.4 Overview of Proposed Approaches

To achieve these objectives, we propose three complementary research directions. First, we introduce context-aware ASR post-correction frameworks that detect and correct rare-word errors using external knowledge and phoneme-level cues. Second, we design E2E contextual ASR methods, such as

the phoneme-augmented robust contextual ASR model, which incorporates entity-level supervision, contrastive learning, and hierarchical biasing. Third, we investigate LLM-based ASR approaches that unify multi-talker recognition and contextual biasing through prompt-driven adaptation and two-stage rare-word filtering. Finally, we extend ASR to downstream tasks by integrating ASR error modeling into multimodal SER and by leveraging ASR transcripts for cross-modal grounding in VMR.

1.5 Contributions of the Thesis

The contributions of this dissertation lie in advancing multimodal and context-aware speech and language understanding through a unified line of research that starts from ASR itself and extends to downstream SLU tasks. The main contributions can be summarized as follows:

- **Advancing context-aware ASR correction and biasing.** We design two complementary frameworks to address the vulnerability of ASR to recognition errors and domain-specific terminology. The first, PMF-CEC, introduces phoneme-augmented multimodal fusion and error-specific selective decoding to effectively correct recognition errors in ASR transcripts. The second, PARCO, develops an E2E contextual ASR approach that integrates phoneme-augmented representations with contrastive entity disambiguation, enabling robust rare-word recognition in dynamic and open-domain contexts.

- **Integrating LLMs for multi-talker contextual ASR.** Building upon the foundations of correction and contextual biasing, we propose CMT-LLM, which leverages LLMs to jointly address multi-talker overlap and contextual rare-word retrieval. Through SOT and prompt-based biasing, combined with two-stage rare-word filtering, CMT-LLM demonstrates how LLMs can unify speech recognition with advanced contextual reasoning in complex conversational scenarios.

- **Extending ASR to multimodal downstream tasks.** We further show that ASR transcripts, even when noisy, are valuable for downstream spoken language understanding tasks. In multimodal SER, we propose M^{4}SER, which incorporates modality-specific and modality-invariant representations, auxiliary tasks of ASR error detection and correction, and advanced training strategies such as adversarial and contrastive learning. In multimodal VMR, we develop 2DP-2MRC, a novel machine reading comprehension-inspired framework that exploits video, audio, and text-based query information to achieve fine-grained temporal localization.

- **A unified perspective linking ASR and spoken language understanding.** Taken together, these contributions demonstrate a consistent research thread: from improving ASR accuracy and robustness through error correction, contextual biasing, and LLM integration, to enabling more reliable

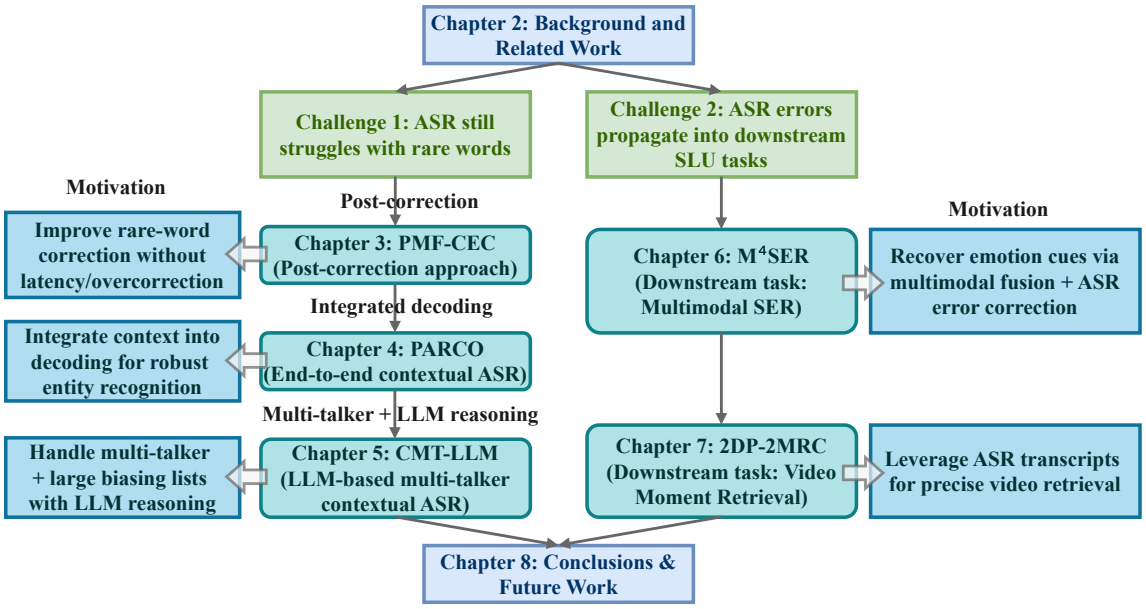


Figure 1.3: Thesis roadmap

multimodal downstream tasks that depend on ASR transcripts. This unified perspective provides not only methodological innovations but also practical insights into how ASR and spoken language understanding can be co-designed for real-world multimodal applications.

1.6 Organization of the Thesis

The remainder of this dissertation is organized as follows. Fig. 1.3 shows the overall roadmap of this thesis. Chapter 2 provides background knowledge and a survey of related work. This chapter first reviews the fundamentals of ASR, including acoustic modeling, language modeling, and E2E frameworks. It then introduces contextual modeling approaches for improving ASR robustness and accuracy. Finally, it outlines related research in spoken language understanding, including SER and VMR, which constitute the downstream tasks studied in later chapters. Chapter 3 presents context-aware ASR post-correction methods, with a particular focus on the PMF-CEC framework. This chapter introduces phoneme-augmented multimodal fusion and selective error correction mechanisms to mitigate ASR errors, especially in handling rare words and homophones. Chapter 4 proposes an E2E contextual ASR framework, PARCO. The chapter discusses phoneme-augmented robust contextual modeling and contrastive entity disambiguation, demonstrating how entity-level supervision can improve recognition in dynamic vocabulary and domain-shift scenarios. Chapter 5 investigates LLM-based ASR methods for multi-talker and contextual recognition. The CMT-LLM framework is presented,

combining SOT with prompt-based decoding and two-stage rare-word filtering, thereby addressing overlapping speech and contextual biasing challenges. Chapter 6 extends ASR research to downstream SLU, focusing on multimodal SER. This chapter introduces the M^{4}SER framework, which integrates multimodal fusion, auxiliary tasks for ASR error detection and correction, and advanced training strategies such as adversarial and contrastive learning, achieving state-of-the-art (SOTA) performance. Chapter 7 explores multimodal VMR, where video content is complemented by ASR transcripts of the corresponding audio track. The 2DP-2MRC framework is proposed, which draws inspiration from machine reading comprehension by employing an AV-Encoder, pointer module, and two-dimensional probability encoder for precise temporal boundary detection. Finally, Chapter 8 concludes the dissertation by summarizing the main contributions and findings. This chapter also outlines potential directions for future work, including the integration of visual modality in ASR, disentangled representation learning, and the application of large multimodal foundation models for unified speech and language understanding.

Chapter 2

Background and Related Work of Automatic Speech Recognition and Understanding

Speech is one of the most natural and fundamental modes of human communication, carrying not only linguistic content but also prosody, emotion, and intention. With the rapid spread of ubiquitous devices, intelligent assistants, and multimodal interactive systems, enabling machines to accurately perceive and understand spoken language has become an indispensable goal [12]. ASR lies at the core of this endeavor, serving as the fundamental interface that bridges human speech with computational understanding. Accurate and robust ASR systems benefit a wide spectrum of applications, from voice assistants and real-time translation to accessibility technologies, and increasingly act as a prerequisite for multimodal intelligence.

Formally, ASR can be described as the task of mapping an acoustic input sequence into a corresponding label sequence. Let $S = (s_1, s_2, \ldots, s_M), \quad s_t \in \mathbb{R}^D$, denote the input acoustic feature sequence of length M, where s_t represents the D-dimensional feature vector at frame t (e.g., Mel-filterbank features or self-supervised embeddings). The output is a label sequence $T = (t_1, t_2, \ldots, t_N)$, $t_n \in \mathscr{V}$, where N is the length of the sequence, $\mathscr{V}$ is the vocabulary (phonemes, characters, or words), and $\mathscr{V}^*$ denotes the set of all sequences over $\mathscr{V}$. The recognition problem is then defined as

$$\hat{T} = \arg\max_{T \in \mathscr{V}^*} p(T \mid S), \tag{2.1}$$

i.e., finding the most probable output sequence $\hat{T}$ given the acoustic input S. In this definition, $p(T \mid S)$ denotes the conditional probability distribution that ASR models aim to estimate.

The development of ASR has gone through several important stages. Early systems were dom-

inated by the Gaussian Mixture Model–Hidden Markov Model (GMM–HMM) framework, where GMMs modeled acoustic likelihoods and HMMs captured temporal dynamics [13]. With the advent of deep learning, Deep Neural Network–HMM (DNN–HMM) hybrid models replaced GMMs with more powerful neural architectures, significantly boosting recognition accuracy [14]. To eliminate the need for handcrafted pipelines involving acoustic models, pronunciation dictionaries, and external language models, the research community turned toward E2E approaches that directly optimize $p(T|S)$. Among them, three paradigms stand out: (i) Connectionist Temporal Classification (CTC), which introduces blank symbols and assumes monotonic alignments between speech frames and labels; (ii) RNN-Transducer (RNN-T), which integrates an encoder, a prediction network, and a joint network, and has become the de facto standard for streaming ASR; and (iii) Attention-based Encoder–Decoder (AED), e.g., Listen, Attend and Spell, which employs attention mechanisms to learn soft alignments and predict each token conditioned on the entire input sequence. More recently, Transformer and Conformer architectures have further strengthened modeling of long-range dependencies and local structures, enabling SOTA performance across benchmarks [15, 16].

In parallel, the field has entered the era of pretraining and foundation models. Self-supervised learning (SSL) techniques such as wav2vec 2.0, HuBERT, and WavLM leverage massive amounts of unlabeled speech to learn general-purpose representations, greatly improving recognition in low-resource and noisy environments. Building upon this, large-scale foundation models such as Whisper have demonstrated unprecedented robustness across domains and languages. Furthermore, the integration of ASR with LLMs is propelling the field toward SLU, where systems not only transcribe speech but also perform higher-level tasks such as summarization, entity recognition, and spoken question answering [17–21].

Beyond transcription, ASR plays a pivotal role as an upstream module for multimodal understanding tasks. In SER, the accuracy of transcripts directly influences the modeling of emotion-related cues embedded in language, where errors in recognizing key words may flip the perceived sentiment. In multimodal VMR, ASR provides temporal anchors for aligning spoken queries with video segments; for example, retrieving "the moment when the character says 'I'm leaving' " requires reliable transcripts as semantic keys. Hence, modern ASR research emphasizes not only lowering word error rates (WER) but also improving contextual awareness, rare-word preservation, domain robustness, and synergy with large models to better support downstream multimodal applications.

This dual perspective—advancing ASR methodologies themselves and enhancing their effectiveness as foundations for downstream multimodal tasks—motivates the research presented in this thesis.

2.1 Historical Development of ASR

The research on ASR can be traced back to the mid-20th century [22]. The earliest attempts were largely experimental, relying on simple filter banks to analyze the energy of speech signals and visualize trajectories for rough transcription. These prototypes were extremely limited, often constrained to digit or vowel recognition and heavily dependent on specific speakers. Nevertheless, they provided the first evidence that machines could, in principle, interpret human speech [23].

In the 1950s, advances in hardware and signal processing enabled more sophisticated systems [24]. At Bell Labs, a recognizer capable of identifying ten digits by matching filter-bank outputs to handcrafted templates was developed in 1952, often regarded as the first complete speech recognition system. Shortly after, researchers built voice-activated typewriters and speaker-independent vowel recognizers [25]. Although these systems could only handle isolated words or phonemes, they marked a transition from proof-of-concept demonstrations to practical prototypes.

The 1960s saw international expansion of speech recognition research. In Japan, several laboratories designed task-specific hardware-based recognizers, such as vowel, phoneme, and digit recognizers [25, 26]. Notably, early work at Kyoto University by Sakai and Doshita proposed an automatic recognition system for speech sounds, aiming at a phonetic typewriter that converts spoken utterances into written symbols, while explicitly addressing speaker and dialect variability [27]. A particularly notable contribution was the introduction of speech segmentation, where utterances were divided into smaller segments for recognition. This concept foreshadowed the development of continuous speech recognition systems [28–30].

A significant leap occurred in the 1970s, when mathematical and algorithmic methods were introduced. Linear predictive coding and dynamic programming provided systematic tools for feature modeling and sequence matching, enabling substantial improvements in speaker-dependent and small-vocabulary tasks [31, 32]. With these advancements, researchers began extending recognition systems to speaker-independent scenarios and large-vocabulary continuous speech recognition (LVCSR). However, increasing variability in speakers, contextual dependencies within speech, and vocabulary growth posed serious challenges that template-based and simple statistical methods could not address [13, 33].

This limitation motivated the adoption of probabilistic sequence modeling approaches, most notably HMMs, for speech recognition. Before the widespread adoption of continuous-density HMMs, early HMM-based ASR systems relied on vector quantization to convert continuous acoustic features into discrete symbols, resulting in discrete HMMs with multinomial output distributions [34]. Although these models enabled the first practical applications of HMMs in speech recognition, the coarse quantization introduced significant information loss and limited modeling accuracy. By the

mid-1980s, HMMs had become the dominant framework for ASR. HMMs modeled the temporal dynamics of speech through probabilistic state transitions, while GMMs approximated the acoustic observation likelihoods. The resulting GMM–HMM pipeline became the standard paradigm for LVCSR, exemplified by the Carnegie Mellon University SPHINX system, which represented a milestone in speech recognition [13]. Through the 1990s and early 2000s, further refinements were introduced, including discriminative training criteria such as maximum mutual information and minimum phone error, as well as speaker adaptation techniques like MLLR and fMLLR. These developments improved robustness, but even with such enhancements, the GMM–HMM architecture struggled in spontaneous conversations, noisy environments, and real-world applications, where its performance approached a practical ceiling.

A paradigm shift emerged with the advent of deep learning. From 2010 onwards, Deep Neural Networks (DNNs) began replacing GMMs as acoustic models within the HMM framework, yielding DNN–HMM hybrid systems that achieved substantial performance gains. Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) were subsequently adopted to capture local spectral patterns and long-range temporal dependencies, respectively. A landmark development came in 2011 when researchers at Microsoft proposed context-dependent DNN–HMM (CD-DNN-HMM) systems, which significantly outperformed traditional approaches on LVCSR tasks and firmly established deep learning as the new foundation of ASR [14].

Building on this success, research rapidly shifted toward E2E modeling, which directly estimates the conditional probability $p(T|S)$ without the need for explicit alignment or separate acoustic, pronunciation, and language models. Three major paradigms emerged: (i) CTC, which assumes monotonic alignment and introduces blank tokens to handle variable-length mappings; (ii) RNN-T, which integrates encoder, prediction, and joint networks, supporting online streaming recognition and achieving widespread adoption in industrial applications; and (iii) AED, such as Listen, Attend and Spell (LAS), which employs attention mechanisms to learn flexible alignments and predicts each symbol conditioned on the entire input sequence [15, 16].

More recently, the Transformer architecture has revolutionized ASR by providing superior modeling of long-range dependencies, while the Conformer further enhanced performance by combining convolutional modules with self-attention to capture both local and global structures. These architectures, often combined with CTC/AED joint training or RNN-T, now constitute the backbone of SOTA ASR systems.

In parallel, the field has entered the era of self-supervised pretraining and foundation models. SSL approaches such as wav2vec 2.0, HuBERT, and WavLM leverage massive amounts of unlabeled speech to learn universal acoustic representations, dramatically improving robustness in noisy and low-resource conditions. On top of SSL, large-scale foundation models like Whisper demonstrated

unprecedented cross-lingual generalization and resilience to domain shifts, setting new benchmarks for robustness and scalability [2, 17–20].

Finally, the integration of ASR with LLMs is reshaping the field [21]. By leveraging the reasoning and contextual capabilities of LLMs, ASR systems are evolving into SLU frameworks, where speech is not only transcribed but also semantically interpreted for tasks such as summarization, entity extraction, spoken question answering, and multimodal retrieval. This trajectory marks the transition of ASR from a narrowly defined recognition technology into a central component of multimodal and semantic intelligence.

2.2 HMM-Based ASR Models

For decades, the HMM framework represented the dominant paradigm in ASR and was widely recognized as the state of the art before the advent of deep learning. The strength of HMM lies in its ability to capture the temporal dynamics of speech through probabilistic modeling, while handling uncertainty and variability in sequential signals. A typical HMM-based ASR system is composed of three components: the acoustic model, the pronunciation model, and the language model [13, 14].

The acoustic model describes the mapping between acoustic features and phonetic units (e.g., phonemes or sub-phonemes). The pronunciation model establishes the correspondence between phonetic units and written symbols, typically via a manually constructed pronunciation lexicon. The language model captures the statistical regularities of word sequences from large text corpora, thereby producing syntactically plausible transcriptions. These three modules are optimized separately but work jointly to complete the mapping from speech to text.

From a probabilistic perspective, the fundamental objective of ASR is to estimate the conditional distribution $p(T \mid S)$ and to select the optimal output sequence based on this distribution. Within the HMM framework, this probability can be decomposed by introducing a hidden state sequence H. Let $S = (s_1, s_2, \ldots, s_M)$ denote the observed acoustic feature sequence, where m indexes the time frame. Under the conditional independence assumption, the acoustic likelihood conditioned on a given hidden state sequence $H = (h_1, \ldots, h_M)$ can be written as

$$p(S \mid H) = \prod_{m=1}^{M} p(s_m \mid h_m). \tag{2.2}$$

The overall likelihood under an HMM is obtained by marginalizing over all possible hidden state sequences,

$$p(S \mid T) = \sum_{H} p(S \mid H)\, p(H \mid T). \tag{2.3}$$

In early systems, $p(s_m|h_m)$ was typically estimated by GMMs, giving rise to the classical HMM–GMM architecture. This approach dominated both academia and industry during the 1990s and early 2000s. However, GMMs were limited in their ability to model high-dimensional and complex acoustic distributions. With the emergence of deep learning, DNNs were introduced to replace GMMs, leading to the HMM–DNN framework. By estimating state posterior probabilities and converting them to likelihoods, DNNs significantly enhanced the modeling power of acoustic models, and HMM–DNN soon surpassed HMM–GMM to become the mainstream paradigm.

For language modeling, the most common form was the n-gram Markov model:

$$p(T) = \prod_{n=1}^{N} p(t_n|t_{1:n-1}), \tag{2.4}$$

where the conditional probability is truncated to a fixed history length to reduce computational complexity.

Despite their success, HMM-based models face inherent limitations. First, the modular nature of the system makes global optimization difficult: the acoustic, pronunciation, and language models are trained separately with distinct criteria, which may not align with overall recognition accuracy. Second, the conditional independence assumption within HMMs does not reflect the true characteristics of speech, thereby limiting the ability to capture long-range dependencies and contextual correlations. These drawbacks become especially problematic in noisy conditions and spontaneous conversational speech.

In summary, although HMM-based methods dominated ASR research and applications for decades, their structural limitations ultimately motivated the shift toward E2E deep learning approaches. The development of HMM–GMM and HMM–DNN not only provided a solid foundation for ASR, but also paved the way for subsequent models such as CTC, RNN-T, and attention-based architectures.

2.3 E2E ASR Models

In recent years, E2E ASR approaches have become increasingly dominant. The central idea is to employ a single neural network model that directly learns the mapping between acoustic observation sequences and target word sequences. Formally, given an input acoustic feature sequence $S = (s_1, s_2, \ldots, s_M)$, where T denotes the number of frames, the system aims to predict an output word sequence $T = (t_1, t_2, \ldots, t_N)$, where N is the number of output tokens. The model parameters are denoted by θ, and training seeks to estimate the optimal parameter set $\bar{\theta}$ that maximizes the posterior

probability of T given S:

$$\bar{\theta} = \arg\max_{\theta} P(T|S;\theta). \tag{2.5}$$

Depending on the alignment strategy, E2E models can generally be divided into two categories: frame-synchronous and label-synchronous.

- **Frame-synchronous approaches:** These predict an output label for each input acoustic frame, and then marginalize or compress the frame-level predictions into a token sequence. Representative methods include:
  - **CTC** [35], which introduces a special blank symbol `<blank>` and marginalizes over all possible frame-to-label alignments.
  - **RNN-T** [36], which extends CTC by incorporating a prediction network that conditions on both acoustic features and the history of emitted symbols.
- **Label-synchronous approaches:** These directly generate output tokens one at a time, without requiring strict frame-level alignment. Let $T = (t_1, t_2, \ldots, t_N)$ denote the output label sequence, where $t_i \in \mathscr{V}$ represents the i-th output token (e.g., a phoneme, character, or word), and N is the output length. Its conditional probability is factorized in an autoregressive manner:

$$P(T|S;\theta) = \prod_{i=1}^{N} P(t_i \mid t_{1:i-1}, S;\theta). \tag{2.6}$$

The attention mechanism was first proposed in neural machine translation [37] and soon introduced into speech recognition [38–40]. Unlike translation, where alignment is often non-monotonic, ASR typically exhibits a roughly monotonic alignment between input frames and output symbols. This property has motivated the development of specialized attention mechanisms such as location-aware attention, monotonic attention, and their variants, which better capture the temporal structure of speech.

In the following sections, we provide an overview of three representative E2E ASR paradigms, namely CTC, RNN-T, and AED. Each of these approaches embodies a distinct modeling philosophy for sequence-to-sequence learning in speech recognition, and together they form the methodological foundation upon which subsequent advances in context-aware and multimodal ASR have been developed.

2.3.1 Modelling Units in E2E ASR

A central design choice in E2E ASR systems is the definition of the modelling unit, i.e., the basic symbol predicted by the decoder. Different languages exhibit different orthographic and phonological structures, and therefore the optimal unit often depends on the target language as well as the amount of available training data.

For **English**, three common options are phones, graphemes, and word pieces. Phone-based systems require an external lexicon and linguistic expertise, which increases the system complexity. In contrast, grapheme-based systems directly operate on characters such as "a", "b", or "c", thereby eliminating the need for a hand-crafted lexicon. With sufficient training data, grapheme-based encoder-decoder systems have been shown to match or even surpass phonetic systems in performance. However, graphemes can be ambiguous because their pronunciation is often context-dependent (e.g., the letter "c" may correspond to /s/ or /k/). To address this issue, word pieces—subword units automatically learned from text corpora—are widely adopted. Word pieces balance between vocabulary coverage and sequence length: frequent words are represented as whole units, while rare words are decomposed into smaller subword fragments. Algorithms such as Byte Pair Encoding (BPE) and unigram language model word pieces are typically used to construct such vocabularies.

For **Chinese**, modelling units are treated differently due to the logographic writing system. Each Chinese character usually corresponds to a morpheme and can be directly mapped to a syllable (e.g., "学" $\rightarrow$ /xue2/). Therefore, characters are often used as the basic recognition units in E2E ASR. This design has two advantages: first, the character set is relatively limited in size (a few thousand symbols), and second, each character carries semantic meaning, which aligns well with human interpretation. However, Chinese has no explicit word boundary markers in its writing system, so directly using characters means that downstream language models must handle word segmentation implicitly. Recently, word-piece models have also been applied to Chinese ASR, where frequent multi-character words (e.g., "学生 (Student)") are encoded as a single unit while preserving the ability to decompose rare words.

For **Japanese**, multiple writing systems coexist: Kanji (logographic characters), Hiragana and Katakana (syllabaries). A common design is to use characters from all three scripts as the modelling units. Compared with Chinese, Japanese characters have a stronger phonetic component because Hiragana and Katakana directly represent syllables. This reduces ambiguity in grapheme-to-phoneme mapping. However, similar to Chinese, Japanese text does not include explicit word delimiters, which complicates the use of word-level units. Subword models such as BPE are again effective, since they can learn units that correspond to frequent morphological or lexical segments, while still allowing decomposition for unseen words.

To summarise, the choice of modelling units differs across languages. English systems benefit from word pieces to resolve grapheme ambiguity and vocabulary size issues. Chinese systems commonly adopt characters, but increasingly incorporate subword models to capture frequent multi-character words. Japanese systems usually integrate Kanji and syllabaries as recognition units, complemented by word-piece approaches for better coverage. This cross-linguistic perspective highlights that the design of modelling units is not universal, and effective ASR systems must consider language-specific writing and phonological characteristics.

2.3.2 CTC-Based Model

In conventional HMM–DNN hybrid systems, the role of DNNs remained constrained. DNNs were typically used only to estimate the posterior probabilities of HMM states, thereby replacing GMMs in acoustic modeling. While this substitution improved accuracy, the temporal dependencies of speech were still modeled by HMM state transitions, leaving the overall system a combination of local neural modeling and global HMM sequence modeling. Such a design limited the full potential of neural networks in ASR.

To overcome this limitation, researchers attempted to replace HMMs with RNNs or CNNs to model temporal dependencies directly. However, these approaches encountered the fundamental alignment problem: training requires a clear mapping between input acoustic frames and output label sequences, yet such frame-level alignments are difficult or impossible to obtain in real-world speech data. Since the loss functions of RNNs and CNNs are usually defined at each time step, the absence of explicit alignments prevented straightforward E2E training.

To address this challenge, Graves et al. [35] proposed the CTC. The key innovation of CTC lies in introducing a special **blank** symbol, $\varnothing$, and allowing repeated labels, thereby transforming the alignment problem into a summation over all valid alignment paths. Given a target label sequence, CTC computes its probability by summing over the probabilities of all possible alignments, enabling training directly from unsegmented input–output pairs without requiring frame-level annotations.

CTC brings two major breakthroughs for E2E ASR:

1. **Resolution of the alignment problem:** CTC eliminates the need for manual segmentation and alignment of training data. This allows neural networks to model temporal dynamics directly, significantly enhancing their role in LVCSR.

2. **Direct transcription output:** Unlike traditional models that predict phonemes or other intermediate units, CTC enables networks to output target transcriptions directly by leveraging the $\varnothing$ symbol. This greatly simplifies both the model design and training pipeline.

By addressing these two challenges, CTC enables a single neural network to map acoustic feature sequences directly to label sequences, achieving truly E2E ASR.

Path Probability Modeling

The overall architecture of the CTC model is illustrated in Fig. 2.1(a). Let the input acoustic sequence be $\mathbf{S} = (s_1, \cdots, s_M)$ of length M. The encoder transforms this sequence into a series of feature vectors $\mathbf{H} = (h_1, \cdots, h_M)$, where each $f_t \in \mathbb{R}^{|\mathscr{V}|+1}$, with $|\mathscr{V}|$ denoting the vocabulary size and the additional dimension corresponding to the blank symbol. After a softmax operation, each frame produces a probability distribution over $\mathscr{V} \cup \{\varnothing\}$.

A path $\pi = (\pi_1, \cdots, \pi_M)$ is defined as a sequence of labels of length M, where each $\pi_m \in \mathscr{V} \cup \{\varnothing\}$. The probability of a path is given by:

$$p(\pi|\mathbf{S}) = \prod_{m=1}^{M} y_m^{\pi_m}, \tag{2.7}$$

where $y_m^{\pi_m}$ is the probability of emitting label π_m at time step m. Intuitively, a path represents a potential alignment between input frames and output symbols.

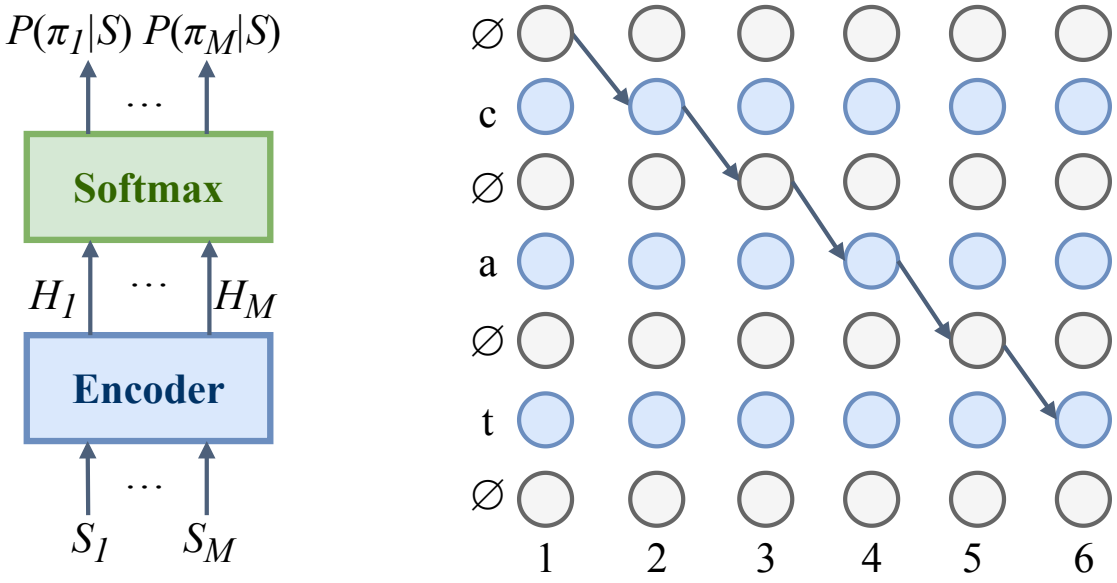
Path-to-Label Aggregation

Since paths always have length M, while target label sequences $\mathbf{T} = (t_1, \cdots, t_N)$ are typically much shorter, CTC employs a many-to-one mapping function $B(\pi)$ to collapse paths into valid label sequences:

1. **Merge consecutive identical labels:** For example, "$c\varnothing aa\varnothing t$" and "$c\varnothing a\varnothing tt$" are both reduced to "$c\varnothing a\varnothing t$".

2. **Remove blank symbols:** The blank symbol is discarded during aggregation, e.g., "$\varnothing c\varnothing a\varnothing t$" is reduced to "*cat*".

Through this process, multiple different paths can correspond to the same target sequence. An example of path alignment is shown in Fig. 2.1(b). The horizontal axis corresponds to input frames $t = (1, \ldots, 6)$, while the vertical axis enumerates the possible output symbols, including characters and the blank symbol $\varnothing$. The highlighted path illustrates one valid alignment that collapses to the target label sequence "cat".



(a) CTC model (b) A path example "∅c∅a∅t"

Figure 2.1: Illustration of the CTC framework. (a) The encoder–softmax architecture, where the encoder maps acoustic features **S** to hidden representations and the softmax layer outputs frame-level label distributions. (b) An example alignment path "$\varnothing c \varnothing a \varnothing t$" for the target word "cat". The horizontal axis denotes input frames, and the vertical axis lists the output symbols, including characters and the blank symbol ∅.

Sequence Probability Computation

The probability of a label sequence is defined as the sum of probabilities of all paths that map to it:

$$p(\mathbf{T}|\mathbf{S}) = \sum_{\pi \in B^{-1}(\mathbf{T})} p(\pi|\mathbf{S}), \tag{2.8}$$

where $B^{-1}(\mathbf{T})$ denotes the set of all paths that collapse to **T**. Training thus maximizes the likelihood of correct label sequences without requiring explicit alignments.

Characteristics and Significance

CTC relies on a conditional independence assumption, where frame-level predictions are assumed to be independent given the input sequence. As a result, the encoder effectively functions as a purely acoustic model, with no inherent language modeling capability. Nevertheless, the blank symbol and path summation mechanisms make CTC a powerful framework: it enables E2E training, removes the

need for manual alignments, and provides the foundation for more advanced sequence-to-sequence approaches such as RNN-T and AED models.

2.3.3 RNN-Transducer E2E ASR Model

Although the CTC framework has made a significant contribution to E2E ASR, it suffers from two intrinsic limitations. First, CTC assumes conditional independence among output symbols given the input, which prevents it from capturing dependencies within the output sequence. As a result, a CTC-trained network essentially functions as an acoustic model without integrated language modeling capability. Second, CTC constrains the output sequence length to be no longer than the number of input frames, making it unsuitable when the target transcription is longer than the input. These issues, particularly the lack of language modeling, substantially limit recognition accuracy. To address these challenges, Graves proposed the RNN-T in [36], which unifies input–output alignment and sequence dependency modeling within a single framework. In principle, RNN-T can map an arbitrary input sequence to a variable-length output sequence while jointly modeling both acoustic evidence and linguistic context.

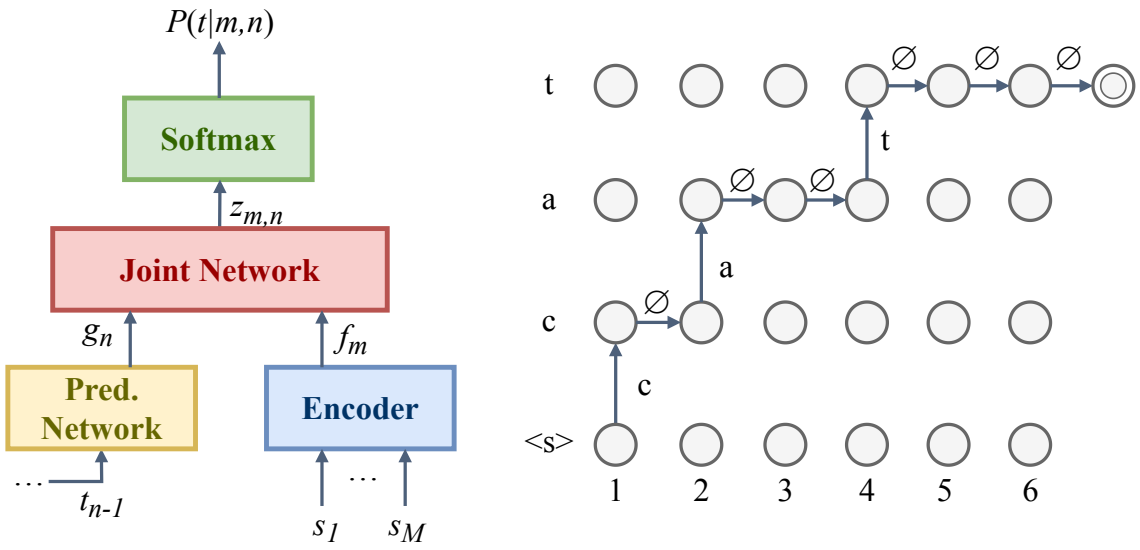
Model Architecture and Core Mechanism

Similar to CTC, RNN-T introduces a blank symbol to handle alignment and computes the probability of a target sequence by summing over all possible alignment paths. However, its path construction and probability computation are more flexible, allowing the model to overcome CTC's constraints. As shown in Fig. 2.2(a), a standard RNN-T consists of three interconnected subnetworks:

Encoder. Given an input acoustic sequence $S = (s_1, s_2, \ldots, s_M)$, where s_m denotes the acoustic feature vector at frame m and M is the input length, the encoder maps S into hidden representations $F = (f_1, f_2, \ldots, f_M)$, with each $f_m \in \mathbb{R}^{|\mathscr{V}|+1}$. Here, $|\mathscr{V}|$ is the vocabulary size and the additional dimension corresponds to the blank symbol $\varnothing$. This component essentially serves as the acoustic model.

Prediction Network. The prediction network models dependencies within the output sequence by conditioning on previously generated labels. At output position n, given the previous label t_{n-1}, the hidden state h_n and output vector g_n are computed as

$$h_n = \mathscr{H}(W_{ih} t_{n-1} + W_{hh} h_{n-1} + b_h), \quad g_n = W_{ho} h_n + b_o, \tag{2.9}$$



(a) CTC model (b) A path example "c∅a∅∅t∅∅∅"

Figure 2.2: Illustration of the RNN-T framework. (a) The overall RNN-T model. (b) An example alignment path "$c\varnothing a\varnothing\varnothing t\varnothing\varnothing\varnothing$" for the target word "cat". The horizontal axis denotes input frames, and the vertical axis lists the output symbols.

where h_n denotes the hidden state at step n, W_{ih}, W_{hh}, W_{ho} are weight matrices, and b_h, b_o are bias terms. The resulting $g_n \in \mathbb{R}^{|\mathscr{V}|+1}$ encodes linguistic context, making this module functionally analogous to a language model.

Joint Network. At each alignment state (m,n), the joint network fuses the acoustic representation f_m and the prediction representation g_n by projecting them into a shared space:

$$h_{m,n}^{joint} = \tanh(Af_m + Bg_n + b), \tag{2.10}$$

where A and B are projection matrices and b is a bias vector. This joint representation is then transformed into logits

$$z_{m,n} = Dh_{m,n}^{joint} + d, \tag{2.11}$$

which are normalized with a softmax to produce a probability distribution over the extended vocabulary $\mathscr{V}' = \mathscr{V} \cup \{\varnothing\}$:

$$P(k|m,n) = \frac{\exp(z_{m,n}^k)}{\sum_{k' \in \mathscr{V}'} \exp(z_{m,n}^{k'})}, \quad k \in \mathscr{V}'. \tag{2.12}$$

Thus, at state (m,n), the model outputs the probability of emitting either a label $k \in \mathscr{V}$ or the blank symbol $\varnothing$, thereby integrating acoustic and linguistic information.

Alignment and Decoding

The alignment process in RNN-T can be visualized as searching paths in a two-dimensional trellis, as shown in Fig. 2.2(b). At state (m,n):

- If the model predicts a label other than the blank, the system moves vertically to $(m,n+1)$ while keeping the same acoustic frame m.
- If the model predicts a blank, the system moves horizontally to $(m+1,n)$ while keeping the output position unchanged.

During training, the probabilities of all valid paths are summed using the forward–backward algorithm, which eliminates the need for explicit frame-level alignments. Hence, RNN-T is regarded as a soft alignment method.

During inference, the model reads input frames sequentially:

- When a label is emitted, the model stays at the current frame to allow multiple labels per frame.
- When a blank is emitted, the model advances to the next frame.

This mechanism naturally enables RNN-T to handle cases where the output sequence is longer than the input sequence.

Advantages and Significance

RNN-T offers several notable advantages:

- **Variable-length mapping:** It supports output sequences longer or shorter than the input sequence.
- **Language modeling capability:** The prediction network explicitly conditions on previous outputs, providing integrated language modeling within the E2E system.
- **Acoustic–linguistic fusion:** The joint network combines acoustic and linguistic information in the probability computation, enabling unified optimization.

By combining these strengths, RNN-T not only inherits the alignment flexibility of CTC but also overcomes its limitations, establishing itself as one of the mainstream frameworks for large-scale E2E ASR.

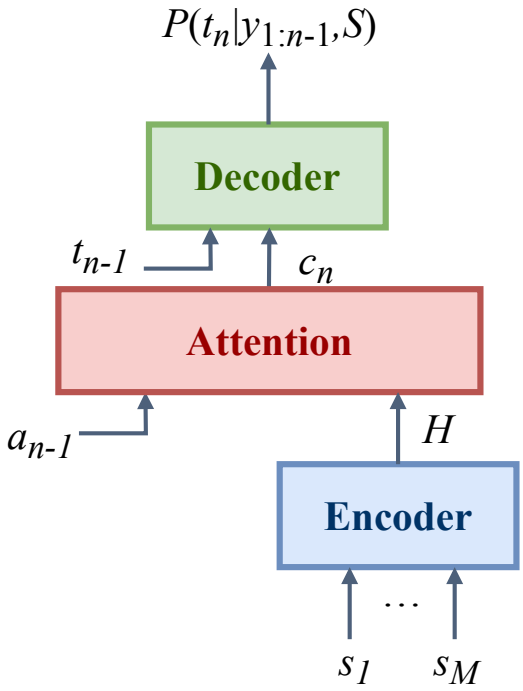


Figure 2.3: Illustration of the AED framework.

2.3.4 Attention-Based Encoder-Decoder (AED) Model

RNN AED Model

The AED model with RNNs represents one of the most influential paradigms in E2E ASR. Before the introduction of attention mechanisms, early E2E ASR systems adopted simple RNN encoder–decoder architectures that directly mapped input acoustic sequences to output label sequences. However, the lack of an explicit alignment mechanism made such models difficult to train and limited their performance, especially for long input sequences. The incorporation of attention addressed this issue by enabling flexible and data-driven alignment between input frames and output tokens. Unlike CTC or RNN-T models, which rely on summing over alignment paths, the AED model integrates alignment and decoding into a single framework through the attention mechanism. This design enables the model to learn a "soft" alignment between input acoustic features and output tokens, thus eliminating the need for explicit frame-level annotations.

Model Architecture. The RNN-AED model consists of three main components: the encoder, the attention mechanism, and the decoder, as shown in Fig. 2.3.

Encoder. Let the input acoustic sequence be denoted as $\mathbf{S} = (s_1, s_2, \ldots, s_M), \quad s_m \in \mathbb{R}^d$, where M is the number of frames and d is the feature dimension. The encoder, typically realized by a stack of RNN layers such as LSTMs or GRUs, transforms $\mathbf{S}$ into a sequence of hidden states $\mathbf{H} = (h_1, h_2, \ldots, h_M), \quad h_m \in \mathbb{R}^D$, where D is the hidden dimension. If no subsampling is applied, the sequence length of $\mathbf{H}$ equals that of $\mathbf{S}$.

Attention Mechanism. At decoding step n, the attention mechanism computes a set of scores by comparing a query vector derived from the previous decoder hidden state with key vectors derived from the encoder hidden states. Specifically, the decoder hidden state d_{n-1} serves as the query, while each encoder hidden state h_m acts as both the key and the value. Formally, the unnormalized attention score is given by

$$\hat{a}_{nm} = \phi(d_{n-1})^\top \psi(h_m), \tag{2.13}$$

where d_{n-1} denotes the decoder hidden state from the previous step, while $\phi(\cdot)$ and $\psi(\cdot)$ are nonlinear transformations (typically multilayer perceptrons). After softmax normalization, the attention weights are

$$\mathbf{a}_n = \text{Softmax}([\hat{a}_{n1}, \ldots, \hat{a}_{nM}]), \tag{2.14}$$

with $\sum_{m=1}^{M} a_{nm} = 1$. The attention weights $\mathbf{a}_n$ define a probability distribution over input frames.

Using these weights, the encoder outputs are aggregated into a context vector:

$$c_n = \sum_{m=1}^{M} a_{nm} h_m, \tag{2.15}$$

which summarizes the most relevant acoustic information from $\mathbf{H}$ for predicting the n-th output token.

Decoder. The decoder functions as a conditional language model. At step n, it takes as input: (i) the previous output symbol t_{n-1}, (ii) the previous decoder hidden state d_{n-1}, and (iii) the current context vector c_n. The decoder then updates its hidden state and predicts a probability distribution over the vocabulary $\mathscr{V}$:

$$P(t_n | t_{1:n-1}, \mathbf{S}; \theta) = \text{Decoder}(t_{n-1}, d_{n-1}, c_n), \tag{2.16}$$

where $y_n \in \mathscr{V}$ and θ denotes the model parameters.

Training and Inference. During training, the model is typically optimized with the teacher forcing strategy, where the ground-truth history $t_{1:n-1}$ is fed into the decoder to maximize the likelihood of the correct output sequence. During inference, decoding starts from a special start-of-sentence symbol $\langle s \rangle$ and continues until the end-of-sentence symbol $\langle /s \rangle$ is generated. Common inference strategies include greedy decoding, which selects the most probable token at each step, and beam

search, which maintains multiple candidate sequences to improve global accuracy.

Properties and Significance. The RNN-AED model has several key advantages:

- **Soft alignment:** the attention mechanism learns a probabilistic alignment between input frames and output tokens, avoiding the need for manual segmentation.
- **E2E optimization:** encoder, decoder, and attention are trained jointly under a unified objective.
- **Implicit language modeling:** the decoder conditions on previous outputs $t_{1:n-1}$, effectively embedding a language model within the E2E system.
- **Variable-length mapping:** the model flexibly generates output sequences longer or shorter than the input sequence, making it well-suited to real-world ASR scenarios.

In summary, the RNN-AED model integrates acoustic modeling, alignment, and language modeling into a unified framework, providing a strong foundation for modern E2E ASR.

Transformer AED Model

Although the RNN-based AED framework has achieved notable progress in alignment learning and sequence modeling, it still suffers from several inherent limitations:

1. **Attention redundancy:** for long input utterances, the encoder generates excessively long hidden representations, which not only increase the computational overhead of the attention mechanism but also introduce irrelevant information.
2. **Training instability:** in long-sequence modeling, the attention distribution may suffer from alignment drift, leading to unstable training and degraded convergence behavior.

To address these challenges, researchers have proposed Transformer-based AED models, which replace recurrent encoders with self-attention mechanisms. By leveraging the global dependency modeling capability of self-attention, Transformer AED models substantially improve the efficiency and stability of long-sequence modeling, and have become the dominant paradigm in recent E2E ASR research. In the Transformer AED, the encoder maps the input acoustic feature sequence $\mathbf{S} = (s_1, \ldots, s_M)$ into hidden representations $\mathbf{T} = (t_1, \ldots, t_N)$ using stacked multi-head self-attention (MHSA) and position-wise feed-forward layers, as presented in Fig. 2.4. Unlike recurrent encoders, each representation h_n is contextualized by attending to all positions in the input sequence, thereby modeling global dependencies.

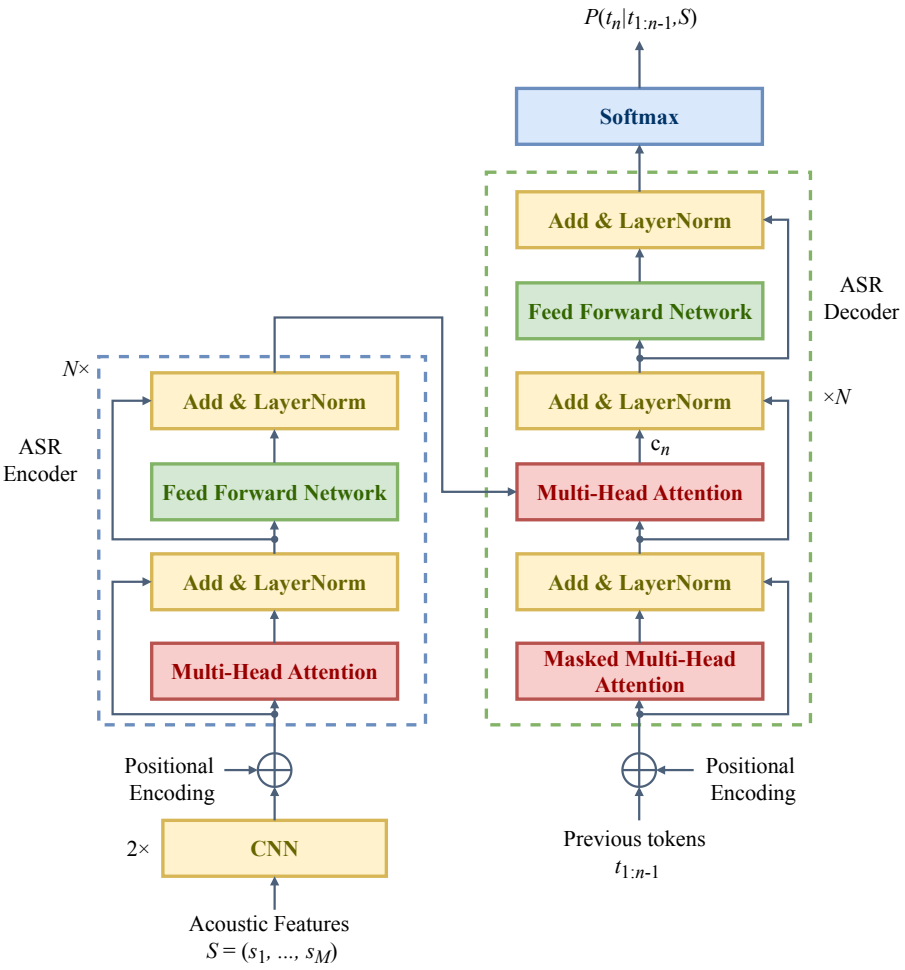


Figure 2.4: Illustration of the Transformer AED framework.

On the decoder side, instead of relying on RNNs, the Transformer decoder employs masked self-attention over the history of predicted tokens $(t_1, \ldots, t_{n-1})$, combined with cross-attention to the encoder outputs $\mathbf{H}$. At step n, the resulting decoder state d_n^{dec} interacts with $\mathbf{H}$ through attention to produce a context vector $\mathbf{c}_n$. The conditional probability of the next token is given by:

$$P(t_n \mid t_{1:n-1}, \mathbf{S}) = \text{Softmax}(W_o[d_n^{dec}; \mathbf{c}_n] + b_o), \tag{2.17}$$

where W_o and b_o are trainable parameters.

Compared with RNN-AED, the Transformer AED offers several key improvements:

- **Global dependency modeling:** Self-attention enables direct interactions among all positions, effectively capturing long-range correlations.

- **Parallelism:** Unlike RNNs that require step-by-step computation, the Transformer allows parallel processing of the entire sequence, improving efficiency.

- **Stable alignment:** Multi-head attention provides multiple alignment perspectives, reducing the risk of alignment drift in long utterances, where the attention focus gradually deviates from the correct input–output correspondence during decoding.

With these advantages, Transformer-based AED models, particularly the Conformer that integrates convolutional layers for local feature modeling, have become the SOTA architecture in E2E ASR.

Conformer Model

While Transformer-based AED models significantly improved long-range dependency modeling in ASR, they still exhibit limitations in capturing local acoustic patterns that are critical for speech signals. To address this issue, the Conformer architecture [41] was proposed, which integrates convolutional modules into the Transformer framework, thereby combining the strength of self-attention in global context modeling with the locality modeling capability of convolutional layers.

The overall Conformer encoder architecture is illustrated in Fig. 2.5. During training, the input acoustic features, such as Mel-filterbank (FBank) or Mel-frequency cepstral coefficient (MFCC) representations, denoted by $\mathbf{S} = (s_1, \ldots, s_M)$, are first passed through data augmentation (SpecAugment) to enhance robustness. A convolutional subsampling layer then reduces the sequence length (e.g., from 10 ms to 40 ms frame rate), producing a compressed representation $\tilde{\mathbf{S}}$. This is followed by a linear projection and dropout layer for normalization and regularization.

The core of the encoder is a stack of N Conformer blocks, each of which is carefully designed to capture both global and local dependencies. As shown on the right side of Fig. 2.5, each block consists of the following components:

- **Feed-Forward Modules (FFN)**: Two half-step feed-forward layers appear at the beginning and end of the block. Each module applies a position-wise transformation:

$$\mathrm{FFN}(s) = \max(0, sW_1 + b_1)W_2 + b_2, \tag{2.18}$$

  where W_1, W_2 and b_1, b_2 are learnable parameters. The scaling factor $\frac{1}{2}$ is applied to stabilize training by balancing contributions from different submodules.

- **MHSA**: This component models long-range dependencies by computing self-attention over the encoder hidden representations. Let $\mathbf{H} = (h_1, \ldots, h_{M'})$ denote the input hidden state sequence to

Table 2.1: Model characteristics comparison.

Model	Delay	Computation Complexity	Language Model Ability	Training Difficulty	Recognition Accuracy
CTC	●○○○○	●●○○○	✗	●○○○○	●●○○○
RNN-T	●●●●○	●●●●●	✓	●●●●●	●●●●○
AED	●●●●●	●●●●○	✓	●●●○○	●●●●●

a Conformer block, obtained from the output of the previous block (or the subsampling layer for the first block), where M' is the sequence length after convolutional subsampling. The query, key, and value matrices are then computed via linear projections of $\mathbf{H}$, i.e., $Q = \mathbf{H}W_Q$, $K = \mathbf{H}W_K$, and $V = \mathbf{H}W_V$, where W_Q, W_K, and W_V are learnable parameters. The attention output is then computed as:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V. \tag{2.19}$$

- **Convolution Module**: To capture local dependencies such as formant transitions or short-term temporal patterns, a depthwise separable convolution is applied. This module effectively complements the global context learned by MHSA.

- **Residual Connections and Layer Normalization**: Each submodule is wrapped with residual connections and layer normalization, ensuring stable optimization and effective gradient flow.

By stacking N such Conformer blocks, the encoder produces a hierarchical representation $\mathbf{H} = (h_1, \ldots, h_{M'})$, which encodes both long-range semantic information and local acoustic cues. This dual modeling capability makes Conformer one of the SOTA architectures for E2E ASR.

2.3.5 Model Comparison

Table 2.1 summarizes the characteristics of the three major E2E ASR architectures. As shown, CTC models exhibit low delay and relatively simple computation, but they lack intrinsic language modeling capability and thus achieve lower recognition accuracy. RNN-T models integrate an implicit language model, leading to higher accuracy but at the cost of increased delay, computational complexity, and training difficulty. AED models provide strong recognition accuracy and explicit language modeling ability, while balancing computational cost and training complexity, although their attention mechanism introduces higher latency [42].

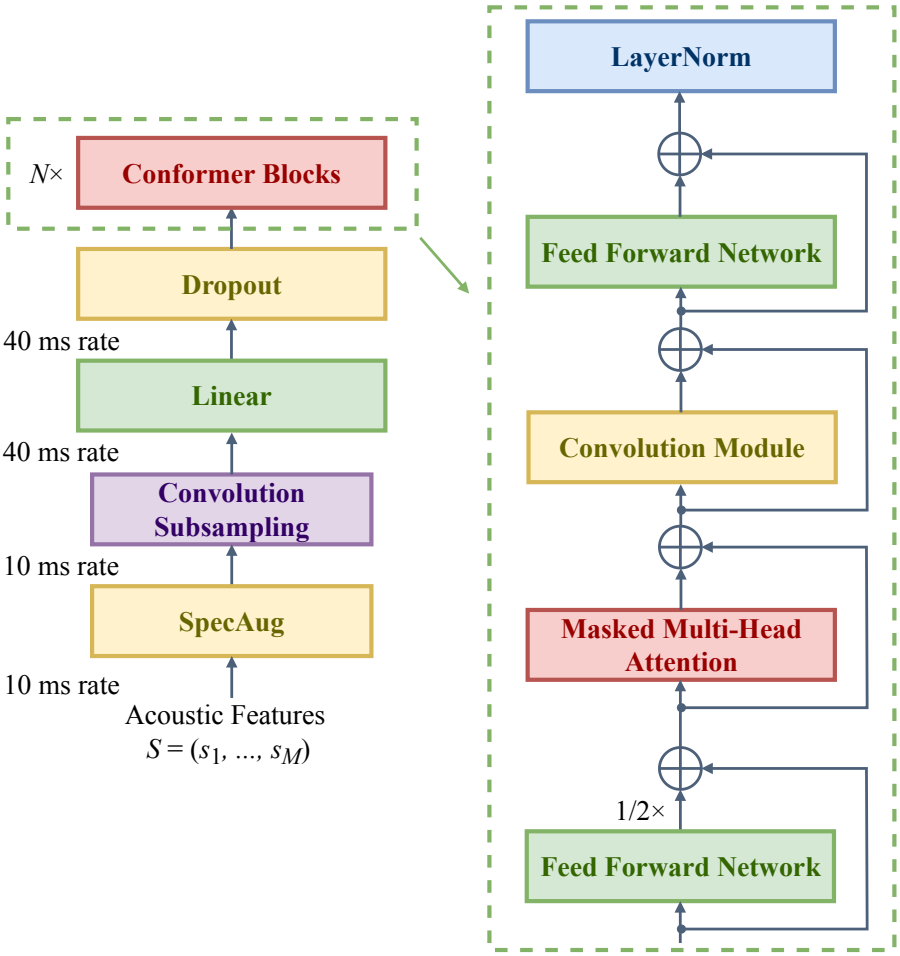


Figure 2.5: Overall structure of the Conformer encoder (left) and the detailed composition of a single Conformer block (right).

2.3.6 ASR Evaluation

The performance of ASR systems is commonly evaluated using the Word Error Rate (WER), which is defined as the minimum edit distance between the reference and the hypothesis at the word level. WER considers three types of recognition errors: insertions (I), deletions (D), and substitutions (S), and is calculated as

$$WER = \frac{I+D+S}{N} \times 100\%, \tag{2.20}$$

where N denotes the number of words in the reference transcription. A lower WER indicates better recognition performance.

Fig. 2.6 shows examples of error computation in three languages. In the English case (Fig. 2.6(a)),

Ref: Toda sensei is giving a talk on signal processing
Hyp: Total sense uh is giving talk on signal processing
Errors: S S I D

(a) English ASR with 1I/1D/2S

Ref: 户 田 老 师 正 在 发 表 关 于 信 号 处 理 的 演 讲
Hyp: 胡 天 老 师 恩 正 发 表 关 于 信 号 处 理 的 演 讲
Errors: S S I D

(b) Chinese ASR with 1I/1D/2S

Ref: 戸田先生は信号処理に関する講演を行っています
Hyp: 富田先生は信号処理に関する講演を行 ています
Errors: S D

(c) Japanese ASR with 1D/1S

Figure 2.6: Examples of WER computation.

the reference contains nine words, among which two are substituted, one is deleted, and one is inserted in the hypothesis, giving a WER of

$$WER = \frac{2+1+1}{9} \times 100\% \approx 44.4\%. \tag{2.21}$$

For languages such as Chinese and Japanese, where written text does not contain explicit word boundaries, evaluation is typically performed at the character level. This metric, referred to as the Character Error Rate (CER), uses the same formula as WER but counts characters instead of words. In Fig. 2.6(b), the Chinese sentence has 17 characters with four errors in total (two substitutions, one insertion, and one deletion), resulting in

$$CER = \frac{2+1+1}{17} \times 100\% \approx 23.5\%. \tag{2.22}$$

In Fig. 2.6(c), the Japanese sentence consists of 22 characters, where one substitution and one deletion occur, leading to

$$CER = \frac{1+1}{22} \times 100\% \approx 9.1\%. \tag{2.23}$$

It should be noted that both WER and CER can exceed 100% when many insertion errors occur. In addition, the Sentence Error Rate (SER), which represents the proportion of utterances containing at least one error, is sometimes reported.

2.4 Multi-talker ASR Model

E2E ASR models provide a strong foundation for speech recognition and have demonstrated remarkable progress under single-speaker conditions, where utterances are typically clean and non-overlapping. Nevertheless, these models are inherently limited in their ability to handle more complex conversational environments. In real-world scenarios such as meetings, interviews, or spontaneous dialogues, it is common for multiple speakers to talk in rapid succession or even simultaneously. Such conditions introduce challenges that go far beyond those addressed by single-speaker ASR: an effective system must not only transcribe speech accurately but also disentangle and attribute linguistic content to the correct speaker. Moreover, practical factors such as frequent speech overlap, background noise, and room reverberation further complicate this task.

To address these challenges, a number of multi-talker ASR approaches have been proposed in the literature, which can be broadly grouped into four representative categories:

(a) **Separation-based approaches**, where the input mixture is first decomposed into single-speaker streams and then transcribed individually.

(b) **Speaker modeling-based approaches**, which explicitly incorporate speaker representations into the recognition process.

(c) **Diarization-based approaches**, which first segment and cluster the audio by speaker before applying ASR.

(d) **E2E approaches with Serialized Output Training (SOT)**, which directly predict multiple speakers' transcriptions in a serialized manner without intermediate separation or diarization.

These four paradigms are illustrated in Fig. 2.7, and each will be introduced in detail in the following subsections.

2.4.1 Separation-based approaches

As shown in Fig. 2.7(a), a straightforward strategy for multi-talker ASR is to first separate the mixed audio signal into several non-overlapping single-speaker streams, followed by applying a conventional single-speaker ASR model to each separated stream. This paradigm is often referred to as the "separate-then-recognize" approach [43]. By explicitly disentangling the overlapped signals, the downstream ASR system can operate under conditions similar to the single-speaker scenario, which makes it easy to integrate with existing ASR architectures and models.

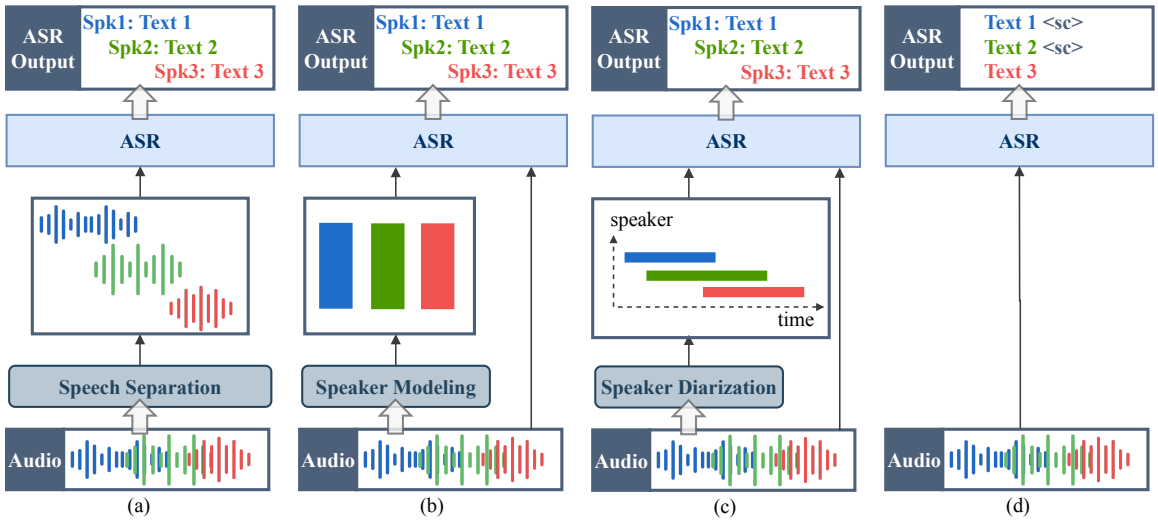


Figure 2.7: Illustration of different multi-talker ASR methods: (a) separation-based approach, (b) speaker modeling-based approach, (c) diarization-based approach, and (d) E2E approach with Serialized Output Training (SOT).

The main advantage of this approach is that it provides relatively clean audio for each speaker, enabling independent recognition and yielding explicit transcriptions associated with each speaker stream. Furthermore, separation models can, in principle, be combined with any off-the-shelf ASR back-end, which enhances flexibility and modularity. A variety of deep learning techniques, such as deep clustering [43], deep attractor networks [44], and permutation invariant training (PIT) [45] for speech separation, have been investigated and have significantly improved separation quality compared with traditional signal processing methods.

However, separation-based approaches also face several critical limitations. First, the performance of the ASR system is tightly coupled with the quality of the separation model, and errors directly propagate into the recognition stage [46]. Second, most separation models require large amounts of parallel synthetic training data, where clean source signals are artificially mixed, leading to a mismatch with real-world conversational audio [6]. Third, the optimization objectives for separation are typically defined at the signal level (e.g., minimizing signal-to-distortion ratio loss), which does not directly correlate with recognition accuracy [47]. As a result, the system may achieve good perceptual separation quality but still produce suboptimal ASR performance in practice.

In summary, separation-based approaches provide an intuitive and modular solution to multi-talker ASR and have inspired a large body of research on deep speech separation. Nevertheless, their reliance on synthetic data, sensitivity to separation errors, and mismatch between separation and recognition objectives limit their effectiveness in real-world conversational scenarios. These

challenges have motivated the development of alternative paradigms, such as speaker modeling- and diarization-based approaches, which attempt to address multi-talker ASR more directly.

2.4.2 Speaker modeling-based approaches

Another line of research avoids explicit signal-level separation and instead focuses on modeling speaker characteristics within the representation space, as illustrated in Fig. 2.7(b). In these approaches, the ASR system is conditioned on speaker-discriminative features so that the model can implicitly disentangle multiple speakers during recognition. A common strategy is to incorporate pre-trained speaker embeddings, such as i-vectors, x-vectors, or d-vectors, into the encoder or attention modules of an E2E ASR model [48–50]. By injecting speaker-specific representations, the system is able to improve robustness to speaker variability and to facilitate multi-speaker decoding.

In addition to using external embeddings, some studies have explored methods to learn speaker representations jointly with ASR, thereby enabling implicit separation of speakers within the hidden feature space [6, 51]. These approaches do not attempt to recover clean audio streams for each speaker, but instead directly guide the recognition network to assign linguistic content to the correct speaker identity. Compared with separation-based systems, this paradigm avoids signal reconstruction artifacts and can be more computationally efficient.

It should be noted that permutation invariant training (PIT) [45] was originally proposed for training separation networks [cf. Fig. 2.7(a)], but its variants have also been extended to multi-speaker ASR where PIT is applied directly on recognition outputs (e.g., PIT-CTC, PIT-attention). In such cases, PIT can be interpreted as an implicit speaker modeling mechanism, as it aligns output hypotheses with the correct speakers without requiring explicit signal separation [52].

Despite their advantages, speaker modeling-based approaches also face several limitations. Their performance strongly depends on the availability and generalization of speaker embeddings, which may degrade in unseen acoustic conditions. Furthermore, the lack of large-scale annotated multi-speaker corpora remains a bottleneck, and most studies are constrained to small-scale or synthetic datasets. Consequently, while speaker modeling provides a promising alternative to explicit separation, its practical deployment in real-world multi-talker ASR systems is still limited.

2.4.3 Diarization-based approaches

As shown in Fig. 2.7(c), another widely adopted strategy is to first determine "who speaks when" using speaker diarization techniques, and then apply ASR to each segmented speech region. This pipeline approach is conceptually straightforward and can be easily implemented by combining existing diarization and ASR modules. In addition, it produces explicit speaker-attributed transcriptions,

which is desirable in many practical applications such as meeting transcription [53, 54].

However, diarization-based methods face several limitations. Their performance deteriorates significantly in highly overlapped speech conditions, since conventional ASR systems are typically designed for single-speaker input and cannot properly handle overlapping segments. Moreover, annotation cost for diarization training data is extremely high, as it requires frame-level or segment-level speaker labels; for example, prior studies have reported that annotating just 10 minutes of audio may take several hours of manual effort [55, 56]. Finally, diarization errors (e.g., missed or incorrect speaker boundaries) can propagate to the recognition stage, resulting in error accumulation in the overall pipeline. These challenges limit the scalability and robustness of diarization-based approaches in real-world multi-talker scenarios.

2.4.4 E2E approaches with SOT

More recently, the serialized output training (SOT) approach has been proposed to directly address multi-talker ASR in an E2E manner, as illustrated in Fig. 2.7(d). In this framework, a single sequence-to-sequence ASR model generates a unified transcription sequence in which special tokens (e.g., <sc>) are inserted to indicate speaker changes [57]. In this way, multiple speakers' utterances are serialized into a single output stream, thereby enabling multi-talker transcription without the need for explicit signal separation or speaker diarization.

This paradigm offers several advantages. First, it avoids error propagation across pipeline stages, which is a common drawback of separation- and diarization-based systems. Second, it inherently supports joint optimization of recognition and speaker change detection within the same model. As a result, SOT can reduce distortions introduced by explicit separation and provide a simpler and more integrated solution. Moreover, since the output remains a single sequence, it is compatible with conventional E2E ASR architectures such as attention-based encoder–decoder or RNN-T models, requiring only minor modifications to the output vocabulary [58, 59].

Nevertheless, SOT-based methods face multiple challenges. A major limitation is that they do not provide explicit speaker identity information, as the inserted tokens only mark speaker boundaries rather than attributing the text to a specific speaker. Recent extensions have attempted to incorporate speaker-attributed labels within the serialized output [58], but this requires speaker information to be available during training and remains challenging in open-domain scenarios. Another limitation is that robust training relies on large-scale simulated multi-talker corpora constructed from single-speaker recordings. For example, [60] leveraged hundreds of thousands of hours of single-talker speech to create millions of hours of artificial mixtures for pretraining. While such methods improve accuracy, they are highly dependent on massive data resources and are difficult to reproduce outside

industrial research settings. Furthermore, the serialized nature of SOT makes it difficult to model long-range context and cross-speaker dependencies, which are crucial in conversational scenarios involving topic shifts, interruptions, and overlapping speech. These issues constrain the performance of SOT in practical applications such as meetings and spontaneous dialogues.

In summary, SOT provides a promising E2E framework for multi-talker ASR by simplifying the system architecture and eliminating the need for explicit intermediate processing. However, its lack of explicit speaker identity, heavy reliance on simulated data, and limited ability to capture complex conversational dynamics highlight important areas for future research, such as integrating speaker modeling, context-aware mechanisms, and large-scale real conversational datasets.

2.5 Contextual Biasing for ASR

As discussed in the previous section, recent multi-talker ASR frameworks such as SOT have achieved notable progress in modeling overlapping speech and conversational dynamics. However, beyond speaker disentanglement and acoustic challenges, real-world ASR systems must also deal with linguistic and contextual variability. Even when the speech signal is correctly separated and transcribed, the model may still fail to recognize rare words, named entities, or domain-specific terminology that lie outside its training distribution.

To address this limitation, contextual biasing has been introduced as an effective mechanism to incorporate external information into the decoding process. By leveraging user-provided word lists, dialogue history, or retrieval-augmented knowledge sources, contextual biasing enables ASR systems to dynamically adapt their predictions to the given environment or task domain.

Although conventional ASR architectures such as CTC, RNN-T, and AED have demonstrated remarkable progress, they inherently rely on fixed language models trained on large but generic text corpora. As a result, they lack the flexibility to integrate context-sensitive or personalized information at inference time. This section reviews representative approaches for contextual biasing, discusses their underlying mechanisms, and highlights their advantages and limitations in practical applications.

2.5.1 Graph Fusion Methods

Graph fusion methods primarily involve approaches based on finite state transducers (FSTs) and Trie-based methods, as well as their combination. On one hand, contextual biasing based on FST models has demonstrated effectiveness in both traditional [61, 62] and E2E [63, 64] ASR systems. These methods utilize LM interpolation or shallow fusion (SF), which are applicable even with minimal training data [65, 66]. FST-based contextual biasing allows precise control over token transitions

through weighted schemes, enabling ASR systems to incorporate domain-specific knowledge about the distribution of rare words. However, traditional FST biasing significantly complicates the inference process, especially in E2E ASR systems. Moreover, since traditional FST models cannot be co-optimized with ASR models via gradient descent, careful readjustment of interpolation parameters is typically necessary following ASR model updates, posing challenges when managing multiple FST models. Additionally, these methods are often limited to specific contexts, such as "call [contact name]" and "play my [playlist] on Spotify", which restricts their capability to handle diverse grammatical structures in natural language [67, 68].

On the other hand, Trie-based contextual biasing methods use Trie structures to enhance model constraints and contextual biases during the decoding process, a technique called DB. This approach leverages the efficient organization and retrieval capabilities of Tries to help ASR models more accurately predict and select vocabulary. Le et al. [1] explored combining Trie-based methods with SF using WFST, extending this integration to RNN LMs to improve the handling of biasing words. They increased efficiency by extracting biasing vectors from Trie constraints representing biasing lists. Additionally, Sun et al. [68, 69] proposed the tree-constrained pointer generator component (TCPGen), which uses a structured Trie representation of biasing words to create a neural shortcut between the biasing lists and the final model output distribution. This method effectively addresses the challenge of managing large biasing lists [70].

Approaches Recently, owing to the improved recognition of rare words and the ease of integration with E2E neural inference engines, context biasing methods based entirely on neural attention mechanisms have gained increasing popularity. Neural context biasing methods in LAS have been discussed in [71], where biasing phrases and contextual entities are encoded via BiLSTM encoders and biased using a position-aware attention mechanism [72, 73]. For RNN-T models, in [74, 75], fully neural attention-based context biasing methods were introduced. In [74, 75], context biasing methods for Transformer Transducers (T-T) and Conformer Transducers (C-T) in auditory encoders and text-based prediction networks were introduced. In [75], the use of context adapters to adapt pretrained RNN Transducer (RNN-T) and C-T models was discussed, which have the advantages of being faster and more data-efficient. Sudo et al. [76] combined biasing phrase index loss and special token training to enhance context performance during inference using the biasing phrase boosted beam search algorithm. Yu et al. [77] proposed LCB-net, employing dual encoders to model audio and long-context biasing, and enhancing model generalization and robustness through dynamic context phrase simulation. Although attention-based methods eliminate the dependence on syntactic prefixes seen in SF methods, they require more memory during training and inference, alter the structure of the original ASR model, which potentially leads to decreased performance of the original ASR model, and are less effective in handling large biasing lists [69].

2.5.2 Attention-based Deep Context Approaches

Recently, owing to the improved recognition of rare words and the ease of integration with E2E neural inference engines, context biasing methods based entirely on neural attention mechanisms have gained increasing popularity. Neural context biasing methods in LAS have been discussed in [71], where biasing phrases and contextual entities are encoded via BiLSTM encoders and biased using a position-aware attention mechanism [72, 73]. For RNN-T models, in [74, 75], fully neural attention-based context biasing methods were introduced. In [74, 75], context biasing methods for Transformer Transducers (T-T) and Conformer Transducers (C-T) in auditory encoders and text-based prediction networks were introduced. In [75], the use of context adapters to adapt pretrained RNN Transducer (RNN-T) and C-T models was discussed, which have the advantages of being faster and more data-efficient. Sudo et al. [76] combined biasing phrase index loss and special token training to enhance context performance during inference using the biasing phrase boosted beam search algorithm. Yu et al. [77] proposed LCB-net, employing dual encoders to model audio and long-context biasing, and enhancing model generalization and robustness through dynamic context phrase simulation. Although attention-based methods eliminate the dependence on syntactic prefixes seen in SF methods, they require more memory during training and inference, alter the structure of the original ASR model, which potentially leads to decreased performance of the original ASR model, and are less effective in handling large biasing lists [69].

2.5.3 ASR Error Correction (AEC)

AEC is proven effective in refining errors generated by ASR models, thereby significantly reducing the word error rate (WER) at the ASR postprocessing stage. Wang et al. [78] introduced a lightweight contextual spelling correction model (CSC) to rectify context-related recognition errors in transcription-based ASR systems, utilizing a shared context encoder and filtering algorithms for large biasing lists. For document-level AEC, Jiang et al. [79] proposed a kNN-based context-aware model with enhanced performance through retrieval from context data stores. Methods for AEC based on autoregressive sequence-to-sequence architectures may suffer from overcorrection issues, introducing new errors or alterations to correct portions and causing significant inference latency. To address these challenges, Wang et al. [80] integrated context information into a non-autoregressive spelling correction model with a shared context encoder, while introducing filtering algorithms and performance balancing mechanisms to control the biasing extent for large biasing lists. Dong et al. [81] proposed the pronunciation guided copy and correction (PGCC) model for AEC, leveraging the encoder-decoder structure of BART pretraining to optimize decisions on whether to copy source input tokens or generate modified ones, effectively identifying and correcting homophone errors.

2.5.4 LLM-based ASR and AEC methods

In recent years, with the development of LLMs, LLM-based approaches for ASR contextual biasing and AEC have attracted increasing attention. These methods leverage the strong contextual modeling and multimodal reasoning capabilities of LLMs by integrating frozen speech encoders with prompt-based frameworks. For example, Seed-ASR [82] and MaLa-ASR [83] incorporate domain knowledge or presentation keywords as prompts, significantly improving recognition of rare and domain-specific words while reducing WER.

Meanwhile, generative LLMs have been widely applied to AEC tasks. HyPoradise [84] introduces an LLM-based benchmark for N-best list correction, demonstrating the ability to recover missing hypotheses beyond traditional reranking. Yang et al. further propose task-activating prompting techniques [85] and extend the scope to speaker attribution and emotion recognition in the GenSEC challenge [86]. In addition, Hu et al. [87] explore robustness under noisy conditions by introducing noise-aware embeddings in the language space.

Despite promising results, LLM-based methods still face several practical challenges. They are computationally intensive, suffer from high inference latency, and are sensitive to prompt design. Their performance also degrades significantly when handling large-scale biasing vocabularies. Moreover, incorporating acoustic features into LLM training leads to increased training costs, and current lightweight adaptations still rely heavily on large-scale data with limited generalization. These limitations suggest that relying solely on LLMs is possibly insufficient for robust and scalable contextual biasing and correction in real-world ASR applications.

2.6 Speech-driven Affective and Multimodal Understanding

Although ASR has made remarkable progress in transcribing spoken utterances into text, transcription alone is insufficient for many real-world applications. Beyond recognizing "what is said," intelligent systems must also understand "how it is said" and "how spoken content interacts with other modalities." This motivates the exploration of higher-level spoken language understanding tasks built upon ASR outputs.

On the one hand, speech-based affective understanding aims to capture paralinguistic cues such as emotion, sentiment, and speaker state, which are not explicitly conveyed in textual transcriptions. On the other hand, speech-driven multimodal understanding focuses on aligning spoken queries with visual information, enabling tasks such as multimodal video moment retrieval. Together, these directions extend the role of ASR from mere transcription to a broader gateway for affective and cross-

modal semantic understanding. This section offers a concise overview of multimodal SER and VMR tasks.

2.6.1 Multimodal Speech Emotion Recognition

SER aims to infer a speaker's emotional state from spoken utterances and plays a central role in applications such as human–computer interaction, mental health monitoring, and social robotics. Emotional information in speech is conveyed not only through what is said (textual content) but also through how it is said (acoustic and prosodic cues). Accordingly, research on SER has evolved from single-modality acoustic modeling to multimodal approaches that integrate both speech and text.

Early SER systems primarily relied on audio signals, extracting prosodic and acoustic descriptors such as MFCC, FBank, or other handcrafted features [88]. With the advent of deep learning, models based on RNNs, CNNs, and Transformer architectures [89–92] achieved substantial improvements by capturing complex temporal and hierarchical patterns in speech. More recently, SSL has driven the development of large-scale pretrained speech models such as wav2vec [17], HuBERT [19], and WavLM [20], which provide rich contextual representations and deliver state-of-the-art performance on multiple SER benchmarks [93].

In parallel, text-based SER has emerged as a complementary paradigm, where transcripts—either manually curated or generated by ASR—serve as inputs for emotion prediction. Textual information provides high-level semantic understanding and often leverages contextual modeling techniques such as RNNs or graph neural networks (GNNs) [94]. However, text alone lacks prosodic and paralinguistic cues, limiting its ability to capture subtle emotional nuances.

To overcome these limitations, multimodal SER integrates both speech and text modalities, jointly modeling acoustic and semantic cues. A common setting involves combining speech signals with textual transcripts (typically from manual annotation or ASR outputs) [95–98]. Fan et al. [95] proposed a multi-granularity attention-based Transformer (MGAT) to address emotional asynchrony and modality misalignment. Sun et al. [96] introduced a method that incorporates shared and private encoders, projecting each modality into separate subspaces to capture both modality consistency and diversity while ensuring label consistency at the output. These methods, however, usually assume access to high-quality manual transcripts, which is often unrealistic in practical applications.

In real-world scenarios, obtaining gold-standard transcripts is costly and impractical, making ASR outputs the primary source of textual information. Yet, even SOTA ASR systems exhibit relatively high WERs in emotional speech, posing significant challenges for SER. Recent research has therefore focused on directly leveraging ASR hypotheses as inputs [8, 99–102]. For instance, Santoso et al. [99] introduced a confidence-aware self-attention mechanism that downweights unreliable ASR

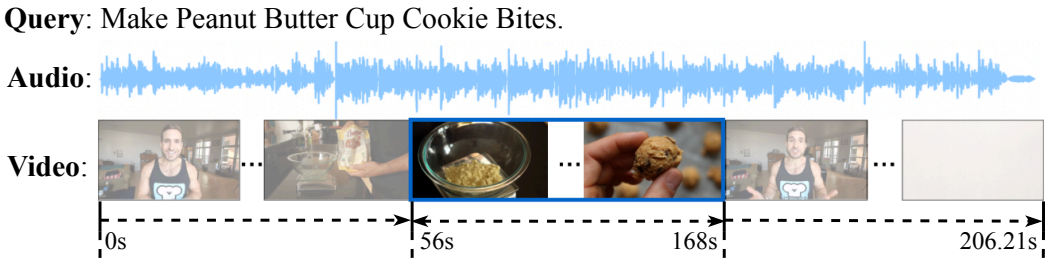


Figure 2.8: An illustrative example of multimodal video moment retrieval, where a natural language query is aligned with both visual frames and the audio track of an untrimmed video to locate the most relevant temporal segment.

tokens to mitigate error propagation. Lin and Wang [8] proposed a robust multimodal SER framework that adaptively fuses attention-weighted acoustic representations with ASR-derived text embeddings to compensate for recognition noise. More recent work [98] further incorporates ASR error detection and correction modules before multimodal fusion, improving the semantic coherence of transcripts and enhancing overall robustness.

In summary, the trajectory of multimodal SER has progressed from early acoustic-only modeling, to combining manual transcripts with speech, and finally to employing ASR-generated transcripts with error-robust mechanisms. While prior work has demonstrated the promise of integrating ASR transcripts into multimodal SER, there remains limited systematic understanding of how WER and fusion techniques interact. Moreover, designing architectures that mitigate error propagation while maintaining inference efficiency is still an open challenge. This dissertation addresses these issues by exploring error-robust multimodal SER frameworks that explicitly account for the imperfections of ASR-generated text.

2.6.2 Video Moment Retrieval

With the explosive growth of online video content, retrieving relevant segments from long untrimmed videos has become an increasingly important yet highly challenging research task. VMR is defined as locating the start and end timestamps of the video segment that semantically corresponds to a given natural language query [103–105]. Unlike traditional video classification or retrieval, which assign a single label or rank to an entire video, VMR requires precise identification of a temporal interval within a continuous video stream.

As illustrated in Fig. 2.8, multimodal VMR typically involves jointly modeling and aligning natural language queries with both visual frame sequences and the audio track of untrimmed videos, in order to identify the most relevant moment. Compared to image-text matching or full-video retrieval,

this task poses several unique challenges. First, untrimmed videos often contain complex temporal structures and long stretches of irrelevant background content, while the target moment may only occupy a small proportion of the video. Second, cross-modal alignment must simultaneously capture semantic details from natural language, visual cues from frames, and acoustic or speech information from the audio track, placing high demands on representation learning and multimodal fusion. Third, queries in practical applications are highly diverse: some describe actions or events (e.g., "a person opens the refrigerator"), while others depend on auditory evidence (e.g., "a phone is ringing" or "a blender starts running"). This requires models to perform reasoning across multiple semantic levels.

As a result, VMR not only demands fine-grained semantic modeling of video content but also precise cross-modal alignment and reasoning across extended temporal ranges, making it a non-trivial and highly complex research problem [10, 106].

Existing methods for VMR can be broadly divided into two categories: moment-based and clip-based approaches. Moment-based methods typically follow a "propose-then-rank" paradigm: they first generate a set of candidate temporal segments and then rank them against the query [103, 104, 107]. The advantage of these methods lies in their ability to achieve relatively high localization accuracy within a large search space and to flexibly model semantic relevance between candidates and queries. However, they suffer from high computational complexity, especially for long untrimmed videos, where generating and scoring a large number of proposals incurs substantial inference overhead, limiting their applicability in real-time or large-scale scenarios.

In contrast, clip-based methods segment videos into fixed-length clips and directly align them with the query [108, 109]. By avoiding candidate generation and ranking, these methods significantly improve computational efficiency, making them more suitable for long videos or large datasets. Nevertheless, clip-based matching typically overlooks broader moment-level and video-level context. Since each clip is treated independently, cross-clip temporal dependencies and global semantic context are underutilized, leading to imprecise localization. This limitation is particularly problematic when queries involve long-duration events or require reasoning with global video context.

Consequently, moment-based methods and clip-based methods represent a typical trade-off between accuracy and efficiency: the former achieves higher precision at the cost of greater computational overhead, while the latter offers higher efficiency but often sacrifices semantic completeness. Designing approaches that balance these two aspects remains an important open question in the VMR community.

More recently, researchers have recognized that relying solely on the visual modality is insufficient to capture the full semantic content of videos. Audio often provides complementary cues that are critical for disambiguation. For instance, actions such as "laughing" and "talking" may appear visually similar but can be easily distinguished through sound. Similarly, certain events like "a phone

ringing" or "a blender starting" are primarily characterized by audio signals, while visual cues alone may be ambiguous or absent. These observations have motivated the emergence of multimodal VMR methods that integrate both visual and audio features to enhance retrieval performance. The central idea is to leverage the complementarity between modalities: vision captures appearance, scenes, and motion, while audio encodes prosody, sound events, and speech content. However, most existing works still treat audio at a shallow acoustic level, extracting low-level features such as spectrograms or filterbanks and directly concatenating or aligning them with visual features, without fully exploiting the rich linguistic information embedded in speech.

The development of ASR opens a promising new avenue. By transcribing the audio stream of videos, ASR systems generate time-aligned textual sequences that provide high-level lexical and semantic information. These transcripts can serve as additional semantic anchors, effectively bridging the gap between video content and textual queries through the speech modality. Unlike low-level acoustic features, ASR outputs directly capture words and phrases, thereby substantially enriching cross-modal alignment. In essence, incorporating ASR enables VMR systems to evolve from a vision–text paradigm to a vision–speech–text framework, enhancing semantic understanding. However, ASR outputs are inherently imperfect, especially in noisy conditions, multi-speaker overlap, or with domain-specific terminology. Recognition errors such as substitutions, insertions, or deletions may distort the semantics of transcripts, leading to mismatches between queries and candidate moments. For example, misrecognizing a key named entity could directly result in erroneous localization. Thus, while ASR transcripts provide significant semantic benefits, their imperfections introduce new challenges that must be addressed.

In summary, research on VMR is gradually shifting from purely vision–text alignment toward richer multimodal frameworks that integrate vision, audio, and text. Incorporating ASR transcripts not only enhances the expressive power of the audio modality but also provides new opportunities for semantic reasoning in VMR. Effectively leveraging the semantic gains of ASR while mitigating the risks of recognition errors represents a critical direction for advancing this field.

2.7 Summary

This chapter reviewed the background and related work of ASR and ASR-related spoken language understanding, providing the technical foundation for the subsequent chapters.

We first traced the historical development of ASR, from conventional HMM-based frameworks to modern E2E models. Particular emphasis was placed on three representative E2E paradigms—CTC-based models, RNN-Transducers, and AED architectures. Their modeling assumptions, training and inference mechanisms, as well as their respective advantages and limitations in terms of latency, mod-

eling flexibility, and recognition accuracy were systematically analyzed.

We then reviewed multi-talker ASR methods, including separation-based, speaker modeling-based, diarization-based, and E2E approaches such as SOT. This line of work highlights the intrinsic difficulty of handling overlapping speech and long-context dependencies, especially in realistic conversational scenarios.

Subsequently, we surveyed contextual biasing techniques for ASR, covering graph-based fusion methods, attention-based deep contextual modeling, AEC, and recent LLM-based ASR and AEC approaches. These studies demonstrate that incorporating external context and semantic priors is crucial for improving rare-word recognition and mitigating recognition errors under domain shift and low-resource conditions.

Finally, this chapter reviewed ASR-driven downstream understanding tasks, with a focus on multimodal SER and VMR. Existing works indicate that ASR errors can significantly degrade downstream performance, motivating the need for ASR-aware and error-resilient multimodal frameworks.

In summary, this chapter highlights two key challenges: (i) improving ASR robustness to rare words, homophones, and overlapping speech through contextual modeling; and (ii) reducing the negative impact of ASR errors on downstream multimodal understanding. These observations directly motivate the methods proposed in the following chapters. Chapters 3–5 focus on context-aware ASR from post-correction, end-to-end modeling, and LLM-driven perspectives, while Chapters 6 and 7 extend these ideas to ASR-aware multimodal SER and VMR tasks, respectively.

Chapter 3

PMF-CEC: Phoneme-augmented Multimodal Fusion for Context-aware ASR Error Correction with Error-specific Selective Decoding

In this chapter, we present PMF-CEC, a phoneme-augmented multimodal fusion framework for context-aware error correction of ASR outputs. The goal is to improve the recognition of rare and phonetically similar words that are often mistranscribed by E2E ASR systems and subsequently harm downstream applications. PMF-CEC combines an error detection module with a selective correction mechanism, enhanced by phoneme-aware multimodal representations and a retention probability strategy to prevent overcorrection. This design allows the system to correct rare-word errors more accurately while preserving correctly recognized words. Experiments across multiple benchmarks show that PMF-CEC (1) significantly reduces WER and biased word error rate (B-WER) compared with text-only approaches, and (2) maintains real-time efficiency, offering a practical and scalable solution for rare-word robust ASR.

3.1 Introduction

E2E ASR systems have achieved remarkable advances in recent years [2]. Despite this progress, they continue to struggle with rare and domain-specific words—such as named entities, technical terms, or personal names—that appear infrequently or are absent from training corpora [3]. These errors are particularly detrimental because ASR transcripts often serve as inputs to downstream spoken

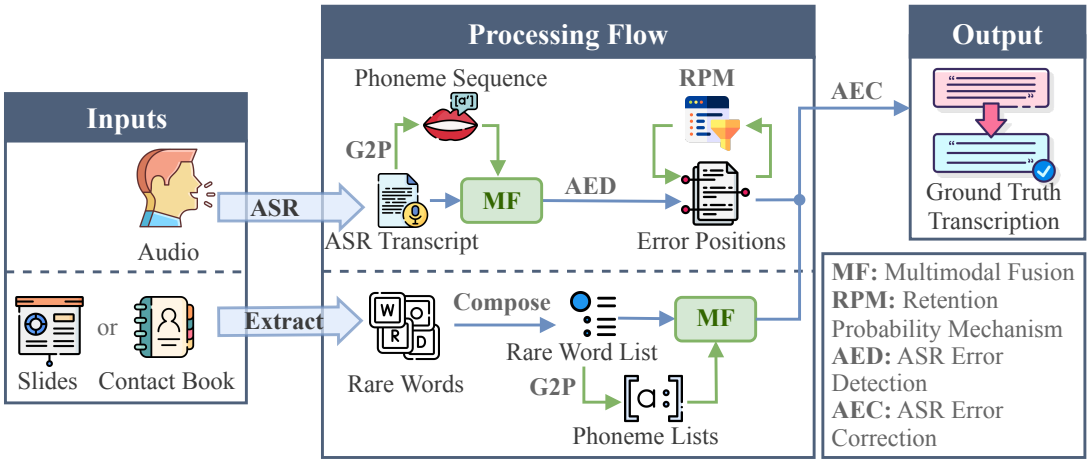


Figure 3.1: Comparison between the previous ED-CEC framework and the proposed PMF-CEC method. Blue arrows indicate components inherited from ED-CEC, while green highlights denote the new extensions introduced in PMF-CEC.

language processing tasks, where misrecognitions of rare words propagate and degrade performance in applications such as summarization, named entity recognition, or emotion recognition [4, 5, 9].

To address this challenge, context-aware AEC has emerged as a practical solution. Unlike approaches that modify the decoding process or retrain ASR models, AEC operates as a lightweight post-processing module, making it easily deployable across systems [80, 101, 110]. However, text-only AEC models struggle with phonetically confusable rare words and tend to produce overdetection errors when deciding whether and how to edit.

In this chapter, we present the error detection and context-aware error correction framework (ED-CEC) [111]. ED-CEC introduces two key ideas. First, it employs an AED module that explicitly identifies error spans before applying corrections, thereby avoiding unnecessary full-sentence rewriting and reducing latency. Second, it adopts a selective, context-aware correction strategy that leverages contextual information to enhance rare-word recognition, achieving efficient and precise post-editing.

Building upon this foundation, we further propose phoneme-augmented multimodal fusion for context-aware error correction (PMF-CEC). As illustrated in Fig. 3.1, PMF-CEC extends ED-CEC in two major directions: (i) it incorporates phoneme-augmented multimodal fusion into the correction encoder to better disambiguate homophones and phonetically similar rare words, and (ii) it introduces a retention probability mechanism (RPM) that assigns calibrated confidence scores to editing operations (Keep, Delete, Change), filtering out low-confidence edits during inference to mitigate overdetection.

The contributions of this chapter can be summarized as follows:

- From our earlier ED-CEC work:
  - An error detection–correction architecture that explicitly localizes erroneous spans through an AED module before performing targeted correction.
  - A selective, context-aware correction strategy that leverages surrounding context, improving rare-word transcription while maintaining low inference latency.
- From the proposed PMF-CEC:
  - A phoneme-augmented multimodal fusion module that enriches textual representations with phonetic cues, improving the disambiguation of homophones and phonetically confusable rare words.
  - A retention probability mechanism (RPM) that calibrates edit confidence and filters out low-confidence edits at inference, reducing overdetection and enhancing correction precision.
- Extensive experiments on five benchmark datasets demonstrate that PMF-CEC achieves 4.25%–14.46% WER reduction (WERR) and 8.39%–12.56% B-WER reduction compared with the text-only ED-CEC baseline, while maintaining inference speeds of 27.21–38.12 ms. Compared with representative LLM-based correction approaches, PMF-CEC achieves comparable or better correction accuracy with 2.4–5.3$\times$ faster inference and greater robustness under large biasing lists.

3.2 Proposed Methodology

3.2.1 Problem Formulation

The task of contextual error correction can be defined as learning a mapping function $f(S,C) = T$. Here, the input source sequence $S = (s_1, s_2, \cdots, s_m) \in \mathbb{R}^m$ corresponds to the raw ASR transcript, while the context $C = (C_1, C_2, \cdots, C_l) \in \mathbb{R}^l$ denotes a list of l contextual terms, typically consisting of rare words or domain-specific items. The desired output is the corrected target sequence $T = (t_1, t_2, \cdots, t_n) \in \mathbb{R}^n$, which aligns with the ground-truth transcript. All sequences are tokenized using a predefined WordPiece vocabulary [112].

To extend this formulation with multimodal information, we additionally incorporate phoneme-level representations. Specifically, $P^s = (p_1^s, p_2^s, \cdots, p_q^s) \in \mathbb{R}^q$ denotes the phoneme sequence corresponding to the source S, where q is the length of the phoneme sequence, and $P^C = (P_1^{(C)}, P_2^{(C)}, \cdots, P_l^{(C)}) \in \mathbb{R}^l$ represents the phoneme sequences aligned with the contextual word list C.

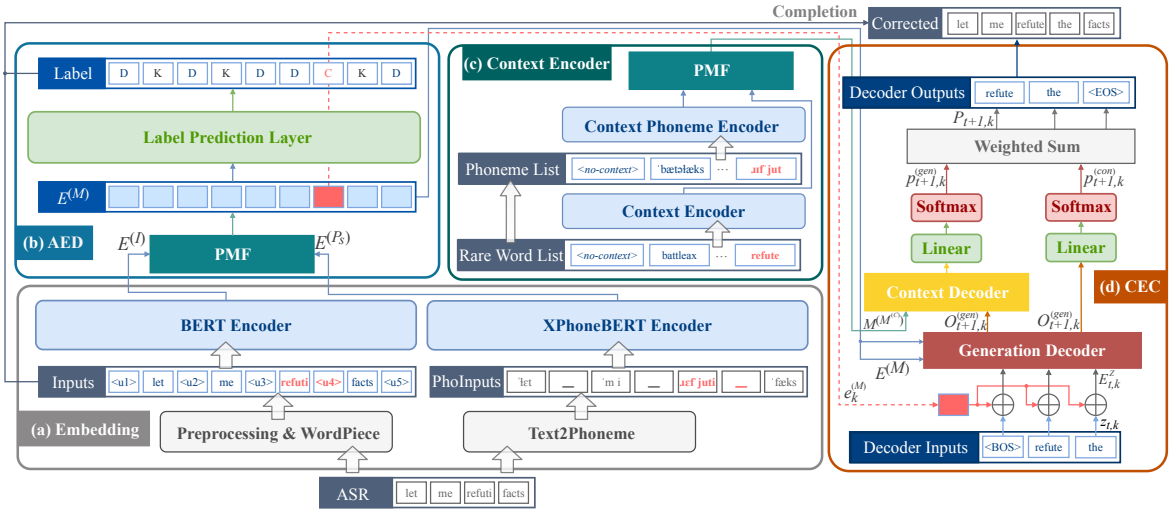


Figure 3.2: Overall architecture of the proposed PMF-CEC model.

3.2.2 Preprocessing

In line with prior work [111, 113], the preprocessing step begins by inserting special placeholder tokens (e.g., <u1>, <u2>, etc.) between every two consecutive words in the ASR transcript S, as illustrated in Fig. 3.2(a). These placeholders act as intermediate positions into which new tokens can be introduced during correction. This design reduces the ambiguity of editing operations: original tokens are restricted to either being preserved or removed, whereas inserted placeholders are constrained to either deletion or substitution with new tokens.

In addition to token-level preprocessing, phoneme sequences are also generated for each word using the `Text2PhonemeSequence` tool[1]. This provides complementary phonetic information, which is later integrated into the multimodal error correction framework.

To supervise the training of the error detection module, we define three elementary editing operations: *KEEP* (K), *DELETE* (D), and *CHANGE* (C). The semantics are straightforward: K indicates that the token remains unchanged, D marks the token for removal, and C specifies that the token should be substituted with another candidate.

Label construction proceeds by first computing the longest common subsequence (LCS) between the ASR transcript S and the ground truth transcript T. With dummy tokens inserted, we obtain an extended sequence $I = (i_1, i_2, \cdots, i_{2m+1}) \in \mathbb{R}^{2m+1}$, where $m+1$ corresponds to the number of dummy tokens. Each element in I is then labeled according to the LCS alignment: aligned tokens are assigned K, misaligned tokens that require substitution are assigned C, and remaining tokens are assigned D. An example of this alignment process is depicted in Fig. 3.2(b).

[1] `https://github.com/thelinhbkhn2014/Text2PhonemeSequence`

Table 3.1: Comparison of the average length between the added tokens and the ground truth transcripts on Librispeech test sets.

	Added Tokens	**Ground Truth Transcripts**
Avg. length	3.00	18.94

This preprocessing design offers a major advantage over fully autoregressive error correction models. Rather than regenerating the entire sequence, our method only modifies tokens at positions requiring correction, while other tokens are either kept intact or removed directly. This selective editing strategy substantially reduces decoding time. As shown in Table 3.1, the average length of added tokens is significantly smaller than that of the ground-truth transcriptions, leading to an estimated reduction of at least 80% in decoding cost compared with autoregressive AEC approaches.

3.2.3 Embedding Module

The embedding module is responsible for producing contextualized representations of both tokens and phonemes, which serve as the foundation for subsequent multimodal fusion. It consists of two components: a text encoder for token-level embeddings and a phoneme encoder for phoneme-level embeddings. A detailed description of each component is provided below.

Contextual Token Representations. For the textual input sequence I, we adopt the pretrained language model BERT [114], which is a multilayer bidirectional Transformer encoder [115]. The model produces contextualized token representations $E^{(I)} = (e_1^{(I)}, e_2^{(I)}, \cdots, e_{2m+1}^{(I)}) \in \mathbb{R}^{(2m+1)\times d_h}$, where d_h denotes the hidden dimension of the encoder:

$$E^{(I)} = \text{BERT}(\text{TE}(I) + \text{PE}(I)). \tag{3.1}$$

Here, $\text{TE}(\cdot)$ represents the token embedding layer and $\text{PE}(\cdot)$ denotes a learnable positional embedding layer, which together provide both semantic and sequential information to the encoder.

Contextual Phoneme Representations. To complement textual features with phonetic information, we employ a pretrained SSL model, XPhoneBERT [116], as the phoneme encoder. Given the phoneme input sequence P_s, the encoder outputs contextualized phoneme representations $E^{(P_s)} = (e_1^{(P_s)}, e_2^{(P_s)}, \cdots, e_q^{(P_s)}) \in \mathbb{R}^{q\times d_h}$, where q denotes the sequence length.

XPhoneBERT adopts the same architecture as BERT, comprising 12 Transformer blocks, a hidden size of 768, and 12 self-attention heads. This design allows it to effectively capture long-range dependencies among phonemes and to encode fine-grained phonetic information that is complementary to textual embeddings.

3.2.4 Phoneme-augmented Multimodal Fusion (PMF) Module

The PMF module is designed to integrate textual and phonetic information into a unified representation. Since the token and phoneme sequences may differ in length, we employ a cross-attention mechanism to enable effective alignment between the two modalities, as illustrated in Fig. 3.3.

Specifically, the output of the encoded text $E^{(I)}$ is used as the query, while the phoneme encoder output $E^{(P_s)}$ serves as both the key and the value. The cross-attention operation is formulated as follows:

$$E^{(M)}_{\text{attn}} = \text{Softmax}\left(\frac{E^{(I)}(E^{(P_s)})^\top}{\sqrt{d_h}}\right)E^{(P_s)}, \tag{3.2}$$

where $E^{(M)}_{\text{attn}} \in \mathbb{R}^{(2m+1)\times d_h}$ denotes the intermediate representation obtained via the cross-attention operation. For simplicity, the residual connection with the original text representation $E^{(I)}$ and the subsequent normalization are omitted from the formulation.

To obtain the final multimodal representation, we combine the text-aware phoneme features with the original contextual token representations using an element-wise addition operation:

$$E^{(M)} = E^{(M)}_{\text{attn}} + E^{(I)}, \tag{3.3}$$

where $E^{(M)} = (e^{(M)}_1, e^{(M)}_2, \cdots, e^{(M)}_{2m+1}) \in \mathbb{R}^{(2m+1)\times d_h}$ represents the final fused multimodal embedding sequence. This integration enables the model to leverage both textual semantics and fine-grained phonetic cues, which is particularly beneficial for distinguishing phonetically similar but orthographically distinct rare words.

3.2.5 ASR Error Detection (AED) Module

The AED module is responsible for identifying erroneous tokens within the ASR hypothesis, thereby guiding the subsequent correction process. As illustrated in Fig. 3.2(b), the AED module operates on the multimodal representation sequence $E^{(M)}$ and predicts an editing action for each token position in the input.

The prediction layer is implemented as a lightweight fully connected network followed by a softmax function, which assigns probabilities over three predefined operations: **K**, **D**, and **C**. Despite its architectural simplicity, the AED module plays a crucial role: it introduces only negligible additional parameters while enabling efficient and accurate localization of potential error spans, substantially improving inference speed compared to fully autoregressive correction strategies.

Formally, for the o-th token representation $e^{(M)}_o \in E^{(M)}$, the probability distribution over the

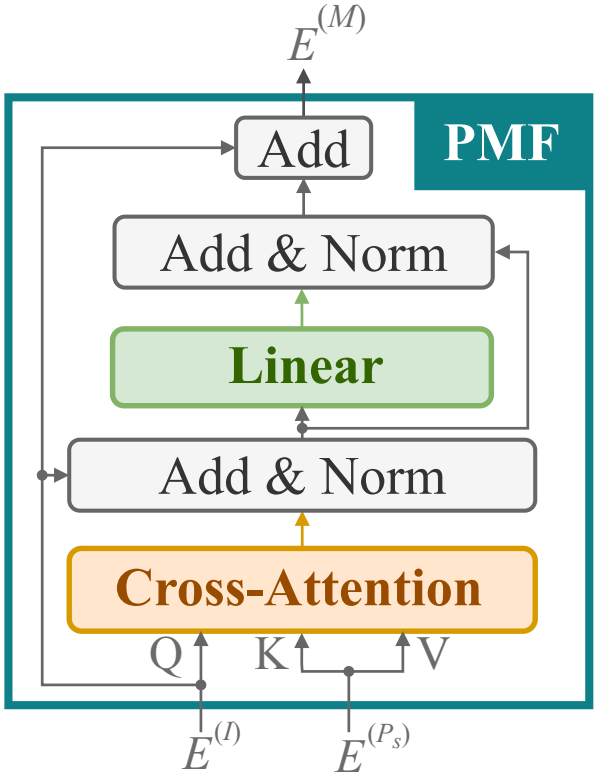


Figure 3.3: Architecture of the proposed PMF module. The text encoder output $E^{(I)}$ attends to the phoneme encoder output $E^{(P_s)}$ through a cross-attention mechanism, after which the fused multimodal representation $E^{(M)}$ is obtained.

editing operations is computed as:

$$P(y_o \mid e_o) = \text{Softmax}\big(\text{FC}(e_o)\big) \in \mathbb{R}^3, \tag{3.4}$$

where y_o denotes the predicted editing operation for the o-th token, and FC represents a fully connected transformation. This design ensures that the AED module not only guides targeted corrections but also maintains computational efficiency.

3.2.6 Context-aware Error Correction (CEC) Module

In contrast to conventional autoregressive decoders that initiate error correction from the ground up, our CEC module concurrently processes all **C** tokens identified by the AED module. This module either generates new tokens via the transformer decoder or selects pertinent tokens from the rare word list to rectify ASR errors, thereby optimizing the correction process, as denoted in Figs. 3.2(c) and

3.2(d).

Decoder Inputs. For the k^{th} **C** position, the generated decoding sequence of length T by the transformer decoder can be represented as $Z_k = (z_{1,k}, z_{2,k}, \cdots, z_{T,k}) \in \mathbb{R}^T$, where $z_{1,k}$ is initialized by a special start token $<BOS>$. We computed the decoder inputs at step t as follows:

$$E_{t,k}^{(Z)} = \text{FC}(\text{TE}(z_{t,k}) + \text{PE}(z_{t,k})) \oplus e_k^{(M)} \in \mathbb{R}^{d_h}, \tag{3.5}$$

where $e_k^{(M)}$ is the multimodal representation of the k^{th} change position. "$\oplus$" denotes a concatenate function. FC is the fully connected layer that maps the decoder inputs back to the same dimension as the embedding of $z_{t,k}$.

Generation Decoder. To generate new tokens, we used the output of the decoder input $\hat{E}_{t,k}^{(Z)}$ as the query and the multimodal representation $E^{(M)}$ as the key and value to the Transformer decoder to obtain the output representation of the decoder layer:

$$O_{t+1,k}^{(gen)} = \text{Softmax}\left(\frac{(\hat{E}_{t,k}^{(Z)})(E^{(M)})^\top}{\sqrt{d_h}}\right)(E^{(M)}), \tag{3.6}$$

where $O_{t+1,k}^{(gen)} \in \mathbb{R}^{d_h}$ is the decoder layer output. For simplicity, the residual connection and the normalization are omitted from the formulation. Finally, the generation output was calculated as

$$p_{t+1,k}^{(gen)} = \text{Softmax}(\text{FC}(O_{t+1,k}^{(gen)})) \in \mathbb{R}^{d_{vb}}, \tag{3.7}$$

where d_{vb} is the vocabulary size of the BERT. Therefore, the next generated token is $z_{t+1,k} = \text{argmax}(p_{t+1,k}^{gen})$.

Additionally, we introduced a context mechanism that dynamically selects between generating new tokens through the generation decoder or choosing relevant tokens from a preprepared rare word list. This context mechanism consists of a context encoder and a context decoder, with the context decoder comprising context attention and context-item attention. The detailed description is as follows:

Context Encoder. We stored l contextual items, consisting of rare words or phrases, in the rare word list, with the construction process detailed in Section 3.3.3. The j^{th} contextual item and the corresponding phoneme sequence are represented as $C_j = (c_j^1, \cdots, c_j^u) \in \mathbb{R}^u$ and $P_j^{(C)} = ((p_j^{(c)})^1, \cdots, (p_j^{(c)})^v) \in \mathbb{R}^v$, $j \in \{1, 2, \cdots, l\}$, where u and v denote the numbers of tokens and phonemes in the j^{th} contextual item, respectively. To optimize model size and improve inference speed, we implemented parameter sharing between the BERT encoder and the contextual encoder. Thus, we used the same BERT encoder to obtain the token representations of each contextual item $E^{(C_j)} =$

$(e_1^{(c_j)}, e_2^{(c_j)}, \cdots, e_u^{(c_j)}) \in \mathbb{R}^{u \times d_h}$:

$$E^{(C_j)} = \text{BERT}(\text{TE}(C_j) + \text{PE}(C_j)), \tag{3.8}$$

where C_j is the j^{th} contextual item. Similarly, we also obtained the corresponding phoneme sequences $E^{(P_j^{(C)})} = (e_1^{(p_j^{(c)})}, e_2^{(p_j^{(c)})}, \cdots, e_v^{(p_j^{(c)})}) \in \mathbb{R}^{v \times d_h}$ for all contextual items, as well as the multimodal representations $E^{(M_j^{(C)})} = (e_1^{(M_j^{(C)})}, e_2^{(M_j^{(C)})}, \cdots, e_u^{(M_j^{(C)})}) \in \mathbb{R}^{u \times d_h}$. Therefore, for all contextual items, we defined them as $E^{(C)} = (E^{(C_1)}, E^{(C_2)}, \cdots, E^{(C_l)}) \in \mathbb{R}^{l \times u \times d_h}$, $E^{(P^{(C)})} = (E^{(P_1^{(C)})}, E^{(P_2^{(C)})}, \cdots, E^{(P_l^{(C)})}) \in \mathbb{R}^{l \times v \times d_h}$, and $E^{(M^{(C)})} = (E^{(M_1^{(C)})}, E^{(M_2^{(C)})}, \cdots, E^{(M_l^{(C)})}) \in \mathbb{R}^{l \times u \times d_h}$, respectively.

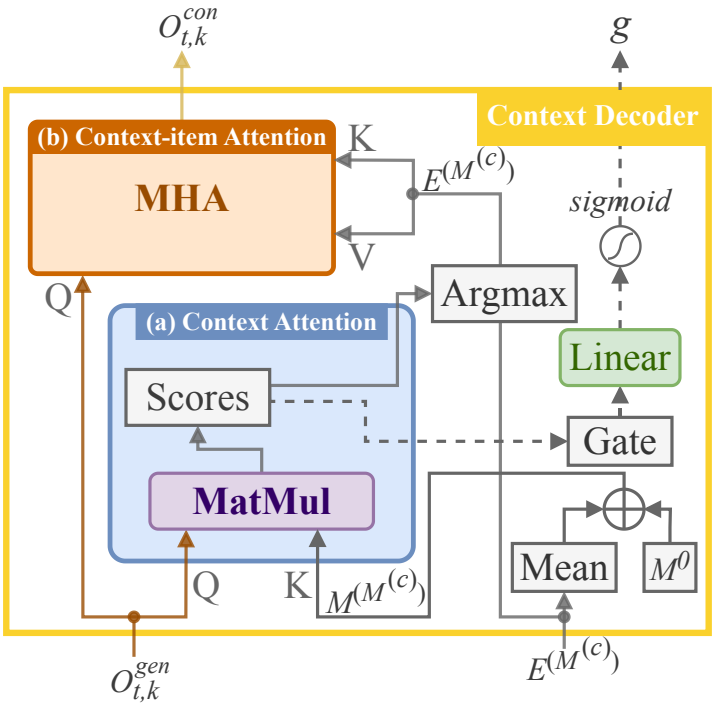


Figure 3.4: Illustration of the context decoder.

Context Decoder. The illustration of the context decoder is shown in Fig. 3.4. We first computed the average of the encoded contextual items and then introduced a learnable dummy token <*no-context*> at the beginning of these contextual items to determine whether relevant information is stored in the rare word list:

$$\bar{E}^{(M^{(C)})} = \text{mean}(E^{(M^{(C)})}) \in \mathbb{R}^{l \times d_h}, \tag{3.9}$$

$$M^{(M^{(C)})} = M^0 \oplus \bar{E}^{(M^{(C)})} \in \mathbb{R}^{(l+1) \times d_h}, \tag{3.10}$$

where $M^0 \in \mathbb{R}^{d_h}$ is the learned hidden representation of the dummy token <*no-context*> and $M^{(M^{(C)})}$ can be interpreted as the summarized tokens for each contextual item.

During step t, the contextual decoder first utilizes a context attention layer to determine the availability and specific positions of relevant contextual items from the rare word list. In this process, similarity scores are calculated by treating the output $O_{t,k}^{(gen)}$ of the generation decoder as the query and summary contextual tokens $M^{(M_c)}$ as the key:

$$\text{scores}_t = O_{t,k}^{(\text{gen})} M^{(M^{(C)})^\top} \in \mathbb{R}^{(l+1)}, \tag{3.11}$$

$$gate_t = \text{scores}_t^{(1)} \in \mathbb{R}^1, \tag{3.12}$$

where $gate_t$ are the similarity scores corresponding to the <*no-context*> token M^0. We defined the index of the highest similarity score for the query Q_t at step t as $m = \text{argmax}(\text{scores}_t) \in \mathbb{R}^1$. If m is nonzero, indicating the presence of relevant contextual knowledge in the rare word list, we computed the contextual output using the context-item attention layer. This layer extracts the relevant information from a specific contextual item using a multihead attention (MHA) mechanism. In the MHA layer, the query input Q_t is the output of the decoder layer $O_{t,k}^{(gen)} \in \mathbb{R}^{d_h}$, whereas the key input K_t and the value input V_t are both the multimodal representation $E^{(M_m^{(C)})} \in \mathbb{R}^{u \times d_h}$ of the m^{th} contextual item:

$$Q_t = O_{t,k}^{(gen)}, K_t = E^{(M_m^{(C)})}, V_t = E^{(M_m^{(C)})}, \tag{3.13}$$

$$O_{t+1,k}^{(con)} = \text{Softmax}(\frac{Q_t K_t^T}{\sqrt{d_h}}) V_t, \tag{3.14}$$

$$p_{t,k}^{(con)} = \text{Softmax}(\text{FC}(O_{t,k}^{(con)})) \in \mathbb{R}^{d_{vb}}, \tag{3.15}$$

where $O_{t,k}^{(con)} \in \mathbb{R}^{d_h}$ is the output of the context-item attention layer. Note that no attention is calculated when the max score corresponds to the dummy token, namely, $m = 0$. Then, the predicted word was acquired by a weighted sum between the generation output $p_{t,k}^{(gen)}$ and the contextual output $p_{t,k}^{(con)}$:

$$g = \sigma(\text{FC}(gate_t)), \tag{3.16}$$

$$P_{t,k} = g \cdot p_{t,k}^{(gen)} + (1-g) \cdot p_{t,k}^{(con)}, \tag{3.17}$$

where σ is the sigmoid function and g is the gate to make a trade-off between the chosen token from the rare word list and the generated token by the generation decoder.

3.2.7 Joint Training

The training of the proposed model is formulated as a multi-objective optimization problem that jointly considers both error detection and context-aware error correction. Specifically, two complementary loss functions are defined to guide the learning process.

Error Detection Loss. The error detection module is optimized using a cross-entropy loss over the predicted editing operations. For each token position o in the ASR hypothesis, the loss is defined as:

$$\text{Loss}_d = -\sum_o \log P(y_o|i_o), \tag{3.18}$$

where i_o denotes the input representation of the o^{th} token and y_o is the corresponding ground-truth label (**K**, **D**, or **C**).

Error Correction Loss. The error correction module is supervised by a combination of two objectives: (1) the likelihood of generating the correct output tokens at each decoding step, and (2) the accuracy of selecting the appropriate contextual item when contextual knowledge is required. The combined correction loss is given as:

$$\begin{aligned}\text{Loss}_e = -\Bigg(&\sum_k \sum_t \log P_{t,k} \\ &+ \sum_k \sum_t \log P(\text{label}_{t,k}|\text{scores}_{t,k})\Bigg),\end{aligned} \tag{3.19}$$

where $P_{t,k}$ denotes the probability of generating the correct token at step t for the k^{th} correction position, $\text{scores}_{t,k}$ are the attention scores over the contextual list, and $\text{label}_{t,k}$ is the ground-truth contextual index.

Overall Objective. Finally, the total training loss is expressed as a weighted combination of the two components:

$$\text{Loss} = \gamma \cdot \text{Loss}_d + \text{Loss}_e, \tag{3.20}$$

where γ is a tunable hyperparameter that balances the relative contributions of error detection and correction during optimization.

This joint objective enables the model to simultaneously improve its ability to accurately identify erroneous tokens and to effectively correct them using both generative and context-based strategies.

3.2.8 Inference

During the inference stage, the model reconstructs the corrected utterance by applying the predicted editing operations together with the generated tokens. As illustrated in Fig. 3.2, the procedure is as follows. Tokens predicted with the label *KEEP* (**K**) are preserved in the output sequence, tokens with the label *DELETE* (**D**) are removed, and tokens marked as *CHANGE* (**C**) are replaced by newly generated words from the correction module. In this manner, the final transcription is composed by integrating both the preserved tokens and the corrected outputs.

A practical challenge arises when the ASR Error Detection (AED) module tends to overdetect, i.e., positions that are already correct are erroneously labeled as errors. Such misclassification may cause the word error rate (WER) to increase rather than decrease. To mitigate this issue, we incorporated a Retention Probability Mechanism (RPM) during inference. Concretely, the AED module produces confidence scores for each predicted editing operation. If the confidence associated with an edit falls below a predefined threshold, the original operation at that position is retained, thereby reducing the risk of unnecessary modifications.

For example, consider the sentence in Fig. 3.2, where dummy tokens have been inserted into the ASR output: *"<u1> let <u2> me <u3> refuti <u4> facts <u5>"*. The ground-truth editing sequence should be *"D K D K D K D K D"*, requiring no modifications. However, the AED predictions may produce *"D K D K D D C K D"*, with corresponding confidence scores of *"0.8 0.6 0.7 0.9 0.8 0.4 0.3 0.6 0.7"*. Given a threshold of 0.5, operations with confidence scores below this value are discarded in favor of the original labels. Consequently, the final operations are corrected back to *"D K D K D K D K D"*.

This mechanism allows the model to maintain high precision in its corrections while avoiding degradation in recognition accuracy due to overdetection.

3.3 Experimental Setup

3.3.1 Implementation Details

The proposed model was implemented in Python 3.7 using the PyTorch 1.11.0 framework. All experiments were conducted on a workstation equipped with an Intel(R) Xeon(R) Gold 6248 CPU @ 2.50 GHz, 32 GB of RAM, and a single NVIDIA Tesla V100 GPU. The overall model configurations and hyperparameters are summarized in Table 6.1.

Both the text encoder and the context encoder were initialized with the shared `bert-base-uncased`[2]

[2] `https://huggingface.co/google-bert/bert-base-uncased`

Table 3.2: Model configurations and hyperparameters used in our experiments.

Configuration	Value
Epochs	20
Optimizer	Adam
Learning rate	0.00005
Dropout	0.1
Hidden size (d_h)	768
Batch size	32
γ	3
Pretrained models	
Word-based encoder	`bert-base-uncased`
Vocabulary size (d_{vb})	30,522
Phoneme-based encoder	`xphonebert-base`
Vocabulary size (d_{vp})	1,960

model, producing token-level embeddings of dimension 768. The corresponding vocabulary size for word tokenization was $d_{vb} = 30{,}522$. Likewise, the phoneme encoder and the context phoneme encoder were initialized using the shared `xphonebert-base`[3] model, yielding phoneme embeddings of dimension 768, with a vocabulary size of $d_{vp} = 1{,}960$.

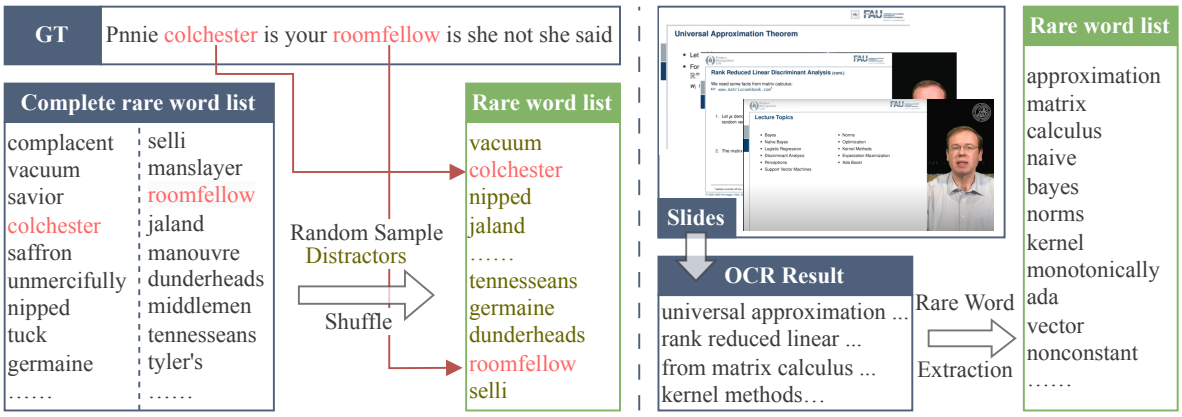
The hidden size d_h was fixed at 768, with 12 attention layers and 12 self-attention heads. The correction decoder was implemented as a single-layer Transformer decoder with a hidden dimension of 768. Model optimization was carried out using the Adam optimizer [117] with a batch size of 32. The loss balancing parameter γ was set to 3, and the initial learning rate was fixed at 5×10^{-5}.

All hyperparameters were tuned on validation sets following standard practice. For inference, the threshold of the Retention Probability Mechanism (RPM) was empirically set to 0.5, which provided the best balance between correction precision and prevention of overdetection.

3.3.2 Datasets

To comprehensively evaluate the performance and robustness of our proposed PMF-CEC method, we conducted experiments across five datasets generated using different ASR engines. These datasets cover a diverse range of speech domains and transcription settings, allowing us to assess both generalization capability and task-specific effectiveness. The construction process for rare word lists is

[3] `https://huggingface.co/vinai/xphonebert-base`



(a) Librispeech (b) PRLVS

Figure 3.5: Pipeline for constructing the rare word list. (a) Simulation of real-world conditions using the LibriSpeech dataset, following the methodology in [1], which is also applied to ATIS and SNIPS. (b) Extraction of novel words from lecture slides for experiments under real-world scenarios.

illustrated in Fig. 3.5, while the overall dataset statistics are summarized in Table 3.3.

- **ATIS** [118]: This corpus contains approximately 8 hours of user queries related to flight reservations, together with manual transcriptions. The ASR hypotheses were generated using a `LAS` system [40].
- **SNIPS** [119]: Originally designed for natural language understanding tasks in voice assistants, this dataset consists of spoken queries collected from real-world user interactions. The ASR transcripts were produced using the `Kaldi` toolkit [120][4].
- **LibriSpeech** [121]: A large-scale corpus of 960 hours of English audiobooks. For ASR transcription, we used the `Wenet` toolkit [122][5]. The *dev-clean* and *dev-other* sets were adopted for validation, whereas *test-clean* and *test-other* were reserved for evaluation.
- **DATA2** [123]: A dataset tailored for E2E spoken named entity recognition (NER), containing 70,763 speech–text pairs with entity-level annotations. We randomly split the corpus into 2,000 validation pairs, 2,000 test pairs, and the remainder for training. The transcripts were generated using the `ESPNet` toolkit [124][6].
- **PRLVS** [125]: A multimodal dataset comprising a full-semester pattern recognition course, with 43 lecture videos and their corresponding slides, totaling 11.4 hours of content. The `SpeechBrain` toolkit [126][7] was employed to generate the ASR transcripts.

[4] `https://github.com/kaldi-asr/kaldi`
[5] `https://github.com/wenet-e2e/wenet/tree/main`
[6] `https://github.com/espnet/espnet/tree/master`
[7] `https://github.com/speechbrain/speechbrain`

Table 3.3: Utterance statistics for the datasets used in our experiments.

Dataset	ATIS	SNIPS	LibriSpeech		DATA2	PRLVS
			Clean	Other		
Train	3,867	13,084	132,553	148,688	66,763	3,680
Valid	967	700	2,703	2,864	2,000	460
Test	800	700	2,620	2,939	2,000	460

3.3.3 Rare Word List Construction

For several datasets, including ATIS, SNIPS, and LibriSpeech, no pre-defined rare word lists are publicly available. To address this limitation, we adopted the validated simulation strategy introduced in [1] to construct rare word lists, as illustrated in Fig. 3.5(a). Specifically, we first compiled a comprehensive rare word inventory for LibriSpeech, consisting of approximately 209.2K distinct words, by removing the 5,000 most frequent words from the language model training corpus of LibriSpeech. Words included in this inventory were then defined as rare words.

For each utterance, we constructed a corresponding rare word list by identifying words in the reference transcription that overlapped with the complete rare word inventory. To increase the difficulty and realism of the task, we additionally augmented each rare word list with a predefined number of distractors (e.g., 1,000), selected according to the experimental requirements. This procedure ensured that each utterance-specific rare word list contained a mixture of actual rare words and distractors[8]. The same methodology was applied consistently to generate rare word lists for the ATIS and SNIPS datasets.

To demonstrate the practicality of rare word list construction in real-world settings, we further considered the PRLVS dataset. In this case, the construction process relied on auxiliary materials associated with lecture videos, namely presentation slides. As shown in Fig. 3.5(b), we extracted text from lecture slides using the Tesseract 4 OCR engine[9]. The extracted tokens were filtered such that only those belonging to the complete rare word list, or occurring fewer than 15 times in the PRLVS training set, were retained. The resulting lecture-specific rare word lists were subsequently applied to all utterances from the corresponding lecture sessions [69].

In addition, to verify the applicability of our PMF-CEC method to rare phrase entities rather than only individual tokens, we employed the DATA2 dataset, originally designed for spoken NER tasks.

[8] `https://github.com/facebookresearch/fbai-speech/tree/master/is21_deep_bias`
[9] `https://github.com/tesseract-ocr/tesseract`

Table 3.4: Coverage of rare word lists on evaluation sets (%). Coverage is computed as the number of rare words divided by the total number of words in each evaluation set.

Dataset	**ATIS**	**SNIPS**	**LibriSpeech**		**DATA2**	**PRLVS**
			Clean	**Other**		
Coverage	27.08	18.11	10.73	9.87	5.13	8.42

Because this dataset provides entity-level annotations (e.g., persons, organizations, and locations), we directly treated the annotated entities as rare words.

The overall proportion of rare words across different datasets is summarized in Table 3.4. Notably, the coverage varies considerably across corpora, ranging from over 27% in ATIS to less than 6% in DATA2, reflecting the different linguistic characteristics and domain-specific vocabularies of each dataset.

3.3.4 Evaluation Metrics

To comprehensively assess the effectiveness of our proposed PMF-CEC approach, we adopted five evaluation metrics that capture both overall transcription accuracy and performance on rare words:

- **WER (Word Error Rate)** measures the overall error rate across all words in the test set, serving as the most widely used metric for evaluating ASR performance.
- **WERR (Word Error Rate Reduction)** quantifies the relative reduction in WER compared with the baseline system, thereby reflecting the extent to which error correction improves recognition quality.
- **U-WER (Unbiased Word Error Rate)** computes the WER exclusively on words that do not appear in the rare word list. This metric ensures that the proposed method does not inadvertently degrade recognition accuracy for common vocabulary items.
- **B-WER (Biased Word Error Rate)** evaluates the WER restricted to words included in the rare word list. It directly measures the system's ability to correctly transcribe rare and domain-specific vocabulary, which is the primary focus of contextual error correction.
- **RW-Recall (Rare Word Recall)** captures the recall of rare words, defined as the proportion of rare words correctly recognized among those present in the utterances. This metric complements B-WER by providing a recall-oriented perspective on rare word transcription.

For insertion errors, if the inserted token belongs to the rare word list, it is counted toward B-WER; otherwise, it contributes to U-WER. The overarching goal of contextualized error correction is

to reduce B-WER and enhance RW-Recall, while ensuring that U-WER remains stable without significant degradation [1]. This balance highlights the model's ability to improve rare word transcription without compromising performance on general vocabulary.

Table 3.5: Measurements of error correction performance on five datasets (%).

Method	ATIS	SNIPS	Librispeech		DATA2	PRLVS
			Test-clean	Test-other		
	WER/WERR (U-WER/B-WER)	WER/WERR (U-WER/B-WER)	WER/WERR (U-WER/B-WER)	WER/WERR (U-WER/B-WER)	WER/WERR (U-WER/B-WER)	WER/WERR (U-WER/B-WER)
Original	30.65/- (20.58/87.78)	45.73/- (34.20/99.64)	3.51/- (1.07/13.31)	9.57/- (3.22/30.63)	7.94/- (6.90/26.52)	18.66/- (10.66/47.24)
SC_BART [127]	21.47/29.95 (14.63/49.25)	30.35/33.63 (21.86/70.25)	3.12/11.11 (1.20/12.15)	9.05/5.43 (3.76/26.83)	7.23/8.94 (6.18/25.18)	15.14/18.86 (10.43/27.67)
distillBART [128]	26.51/13.51 (18.54/74.67)	33.28/27.23 (24.08/76.43)	3.35/4.56 (1.26/12.75)	9.37/2.40 (4.42/27.11)	7.62/4.03 (6.54/26.36)	17.98/3.64 (10.64/43.57)
$\text{ConstDecoder}_{\text{trans}}$ [113]	21.74/29.07 (14.78/50.57)	30.98/32.25 (22.09/71.43)	3.27/6.84 (1.22/12.47)	9.35/2.30 (4.38/27.67)	7.41/6.68 (6.11/27.20)	15.31/17.95 (10.95/28.33)
ED-CEC [111]	18.95/38.17 (14.88/38.38)	28.57/37.52 (21.47/62.79)	2.71/22.80 (1.25/9.72)	8.23/14.00 (3.85/21.02)	5.39/32.16 (4.72/15.11)	13.17/29.42 (10.01/20.72)
PMF-CEC (Proposed)	**16.21/47.11 (13.25/35.16)**	**25.34/44.59 (19.89/56.37)**	**2.52/28.21 (1.12/8.54)**	**7.88/17.66 (3.35/18.38)**	**5.11/35.64 (4.69/13.69)**	**11.68/37.41 (9.31/18.56)**

3.4 Experimental Results

3.4.1 Comparisons with Baseline AEC Methods

To validate the effectiveness of our proposed PMF-CEC model, we compared it against five representative AEC baselines on five public datasets. The compared systems are summarized as follows:

- **Original**: the raw ASR transcripts without any postprocessing. This serves as the baseline reference point.
- **SC_BART** [127]: a sequence-to-sequence model based on BART, which has demonstrated strong performance in AEC tasks and represents a state-of-the-art autoregressive baseline.
- **distillBART** [128]: a lightweight, distilled version of BART designed to reduce computational cost while maintaining reasonable performance.
- **$\text{ConstDecoder}_{\text{trans}}$** [113]: a constrained decoding approach that improves inference speed in AEC by reducing decoding complexity.

Table 3.6: Average inference time in ms.

Method	**ATIS**	**SNIPS**	**Librispeech**	**DATA2**	**PRLVS**
SC_BART [127]	90.30	75.30	144.32	156.12	103.62
distillBART [128]	45.55	41.55	69.80	75.74	57.60
$\text{ConstDecoder}_{\text{trans}}$[113]	25.61	26.66	23.69	21.39	18.29
ED-CEC [111]	32.59	31.87	30.16	24.93	22.86
PMF-CEC (Proposed)	38.12	36.94	34.48	29.19	27.21
vs SC_BART	2.8×	2.4×	4.2×	5.3×	4.5×
vs distillBART	1.4×	1.3×	2.0×	2.6×	2.5×
vs $\text{ConstDecoder}_{\text{trans}}$	0.8×	0.8×	0.7×	0.7×	0.8×
vs ED-CEC	0.9×	0.9×	0.9×	0.9×	0.8×

- **ED-CEC** [111]: our previous work, which introduced an error detection (AED) mechanism to locate error-prone tokens and a context-aware correction module to selectively perform token replacement, thus reducing decoding overhead and enhancing rare word transcription.

The comparative results are presented in Table 3.5. When the size of the rare word list is set to 100, PMF-CEC consistently outperforms all baselines across the five datasets. In particular, PMF-CEC achieves substantial reductions in both overall WER and B-WER, demonstrating its ability to improve rare word transcription accuracy. Compared with SC_BART, our model achieves relative B-WER reductions ranging from 19.76% to 45.63%, highlighting the effectiveness of phoneme-augmented multimodal fusion. Furthermore, relative to ED-CEC, PMF-CEC introduces two key improvements—phoneme augmentation and the retention probability mechanism (RPM)—which together yield an average reduction of 8.22% in U-WER and 10.52% in B-WER. These results confirm that PMF-CEC not only strengthens error correction capability but also mitigates overcorrection.

In addition to correction accuracy, inference efficiency is an essential factor for practical deployment. As shown in Table 3.6, PMF-CEC delivers a 2.4 to 5.3× speedup compared with the autoregressive baseline SC_BART, while maintaining significantly better correction accuracy. Although its inference time is slightly higher than those of partially autoregressive methods such as

$\text{ConstDecoder}_{\text{trans}}$ and ED-CEC, the performance gains in error correction make PMF-CEC a more balanced solution in terms of WERR and latency.

Finally, Table 3.7 further demonstrates that PMF-CEC consistently improves performance across multiple ASR backends, including Whisper, TCPGen, and GPT-2 rescoring. This indicates that our method is not tied to a specific ASR architecture and can be robustly applied to enhance diverse ASR systems.

Table 3.7: WER and B-WER on LibriSpeech test sets, PRLVS test set and DATA2 test set using Whisper and TCPGen, rescored with GPT-2 (%), and corrected by PMF-CEC. LM weights are finetuned on each validation set separately.

Method	**Librispeech Test-clean**		**Librispeech Test-other**		**DATA2**		**PRLVS**	
	WER ↓	B-WER ↓	WER ↓	B-WER ↓	WER ↓	B-WER ↓	WER ↓	B-WER ↓
Whisper meduim.en	4.00	15.08	6.83	21.89	4.96	15.22	6.74	19.83
+ TCPGen [68]	3.46	12.05	6.41	18.15	4.02	11.64	6.27	15.77
+ GPT-2 [129]	3.88	14.91	6.74	21.34	4.51	13.98	6.50	18.26
+ PMF-CEC	2.99	9.61	6.02	16.45	3.31	8.82	5.99	12.51
+ TCPGen + GPT-2 + PMF-CEC	**2.57**	**8.17**	**5.85**	**15.50**	**3.04**	**7.27**	**5.56**	**10.67**

3.4.2 Comparisons with Other Contextual Biasing Methods

To further examine the effectiveness of our proposed PMF-CEC when applied to large-scale ASR systems, we conducted experiments using the `Whisper-medium.en` model [2, 67]. We selected one general-domain corpus (LibriSpeech) and two domain-specific corpora (PRLVS and DATA2), and applied PMF-CEC to the corresponding ASR transcripts. For comparison, we incorporated two representative contextual biasing approaches:

- **TCPGen** [68]: This method integrates rare word lists into E2E ASR models by organizing words into an efficient prefix tree and adding a neural shortcut to improve rare word recognition during decoding. Following the original setup, we froze the parameters of the `Whisper` model and trained only the TCPGen module.
- **GPT-2** [129]: A large pretrained Transformer-based LM trained on over ten billion words from web text[10]. In our experiments, we used `GPT-2`[11] without fine-tuning to rescore the 50-best hypotheses generated by the ASR system, with LM weights tuned on the validation sets.

[10] `https://openai.com/research/gpt-2-1-5b-release`
[11] `https://huggingface.co/openai-community/GPT-2`

To ensure comparability with prior contextual biasing work [1, 68–70], text normalization was not applied during evaluation unless explicitly specified. A detailed discussion of this decision is provided in Section 3.4.9.

As reported in Table 3.7, PMF-CEC achieves consistent improvements when applied to ASR transcripts from large-scale models with a rare word list size of 100. Specifically, on the LibriSpeech dataset, PMF-CEC achieves relative B-WER reductions of 36.27% and 24.85% on the *test-clean* and *test-other* sets, respectively. On the DATA2 corpus, a relative rare word recall improvement of 42.05% is obtained, while on PRLVS, a relative B-WER reduction of 36.91% is observed. These results suggest that the benefits of PMF-CEC are more pronounced on domain-specific corpora than on LibriSpeech, since rare word lists in the latter are largely composed of generic vocabulary, whereas the former include specialized terms such as named entities, technical jargon, and domain-specific expressions.

In addition, TCPGen also leads to notable reductions in B-WER, while GPT-2, serving primarily as a general-purpose LM, provides only limited improvements in rare word recognition. On average, PMF-CEC achieves relative B-WER reductions of 18.63% compared with TCPGen and 31.72% compared with GPT-2, clearly demonstrating the advantages of our approach. Finally, when all three methods are combined, further improvements are obtained, yielding the best overall WER performance.

3.4.3 Comparisons with LLM-based ASR and AEC Methods

To further assess the effectiveness of PMF-CEC, we compared it with three representative LLM-based ASR and AEC approaches on the DATA2 dataset:

- **Whispering LLaMA ($\mathscr{WL}$)** [130]: A cross-modal generative AEC framework that integrates audio and text features for instruction-guided correction. It is initialized from `Alpaca`[12], a model finetuned from `LLaMA-7B` [131]. Following the original setup, we generated 5-best hypotheses using `whisper-tiny`[13], and extracted audio features with `whisper-large-v2`[14]. For $\mathscr{WL}$, we adopted the medium variant $\mathscr{WL}_M$ with low-rank adaptation (LoRA) rank $r = 16$.
- **SLAM-ASR** [21]: An LLM-based E2E ASR framework that connects a speech encoder and an LLM via a linear projection layer. In our experiments, we used `WavLM-Large`[15] as the speech encoder and `Vicuna-7B`[16] as the LLM backbone, finetuned with LoRA (rank $r = 16$). The encoder

[12] `https://github.com/tatsu-lab/stanford_alpaca`
[13] `https://huggingface.co/openai/whisper-tiny`
[14] `https://huggingface.co/openai/whisper-large-v2`
[15] `https://huggingface.co/microsoft/wavlm-large`
[16] `https://huggingface.co/lmsys/vicuna-7b-v1.5`

outputs were downsampled and projected to match the LLM input dimension.

- **MaLa-ASR** [83]: An extension of SLAM-ASR that incorporates domain-specific keyword prompting to improve rare-word recognition. The same SLAM-ASR architecture is employed, with rare words prepended to the prompt during decoding.

The results in Table 3.8 show that $\mathscr{W}\mathscr{L}_M$ achieves a WER of 15.61% by performing generative AEC over 5-best hypotheses, outperforming PMF-CEC applied directly to the Whisper 1-best output (20.90%). However, PMF-CEC achieves significantly lower B-WER, demonstrating stronger biasing capability. Importantly, combining the two approaches ("$\mathscr{W}\mathscr{L}_M$ + PMF-CEC") yields further improvements, indicating complementarity between generative AEC and context-aware correction.

Table 3.9 further compares PMF-CEC with SLAM-ASR and MaLa-ASR. When integrated into SLAM-ASR with 100 biasing words, PMF-CEC reduces WER from 5.75% to 4.66% and B-WER from 23.44% to 16.13%, while increasing inference time by only 0.04 seconds. These results confirm that PMF-CEC scales well with larger biasing lists and can be seamlessly combined with LLM-based ASR systems.

By contrast, MaLa-ASR reduces B-WER to 5.75% with 100 biasing terms, outperforming SLAM-ASR in rare word recognition. However, it incurs a higher overall WER of 5.87%, suggesting interference between keyword prompting and semantic modeling. More critically, when the number of biasing terms increases to 1000, MaLa-ASR suffers from severe instability, with WER rising to 116.99% and B-WER reaching 100.00%, indicating poor robustness under large-scale prompting.

In summary, PMF-CEC demonstrates stable and efficient performance compared with LLM-based approaches. Its lightweight design ensures superior efficiency and reliability, making it a strong candidate for real-time ASR applications where both correction accuracy and latency are critical.

3.4.4 Impact of Individual Modules in PMF-CEC

To investigate the individual contributions of different components within PMF-CEC, we conducted an ablation study on the LibriSpeech *test-clean* dataset. The results are summarized in Table 3.10.

When the phoneme encoder was removed, the WER increased from 2.52% to 2.62%, and the B-WER rose from 8.54% to 9.55%. This finding indicates that phoneme-level representations provide complementary cues that improve model robustness against acoustic ambiguity and homophonic interference, particularly in cases where biased substitutions involve phonetically similar words.

Excluding the context decoder led to the most substantial degradation in performance, with WER increasing to 3.18% and B-WER to 12.13%. This highlights the pivotal role of the context decoder in integrating linguistic knowledge and rare-word biasing information. By explicitly modeling con-

Table 3.8: WER and B-WER comparison of the LLM-based AEC model $\mathscr{WL}_M$ and our PMF-CEC model on the DATA2 test set (%). We use Whisper Tiny to generate 5-best hypotheses. "Oracle" refers to the candidate with the lowest WER compared with the ground truth within the 5-best hypotheses. The unit of inference time is seconds per sentence (s/sentence).

Model	**Biasing Size**	**WER↓ / B-WER↓**	**Inference Time**
Whisper Oracle	-	13.87 / 45.48	-
$\mathscr{WL}_M$ [130]	-	15.61 / 44.73	3.47
Whisper 1-best	-	25.69 / 56.51	-
+ PMF-CEC	100	20.90 / 30.82	**0.10**
$\mathscr{WL}_M$ [130] + PMF-CEC	100	**13.14 / 28.33**	3.58

textual relevance, the decoder enables the system to correctly prioritize rare-word candidates, thereby achieving significant improvements in bias correction accuracy.

Finally, removing the Reliability-based Postprocessing Module (RPM) resulted in a higher WER of 2.61%, although the raw accuracy of editing operations within the AED module increased from 95.73% to 97.40%. These results suggest that the RPM is essential for mitigating overcorrection by filtering out low-confidence edits, thus stabilizing model outputs and improving overall correction quality.

Overall, the ablation results demonstrate that each module contributes uniquely to the effectiveness of PMF-CEC: the phoneme encoder enhances discrimination at the acoustic level, the context decoder provides strong biasing capabilities, and the RPM safeguards against overcorrection, together yielding a balanced and robust correction framework.

3.4.5 Impact of Rare Word List Size

To further assess the robustness of PMF-CEC under different contextual conditions, we examined the effect of rare word list size on model performance. Experiments were conducted on the LibriSpeech *test-clean* and *test-other* sets, with the results summarized in Table 3.7 and visualized in Fig. 3.6. Rare word lists were constructed with varying sizes, ranging from 100 to 3000 entries, and extended with distractors.

As illustrated in Fig. 3.6, PMF-CEC achieves the lowest WER and B-WER on both test sets when the rare word list contains 100 entries. As the list size increases to 3000, both WER and B-WER

Table 3.9: WER and B-WER comparison of different LLM-based ASR models and our PMF-CEC model on the DATA2 test set (%). The unit of inference time is seconds per sentence (s/sentence).

Model	**Biasing Size**	**WER↓ / B-WER↓**	**Inference Time**
SLAM-ASR [21]	-	5.75 / 23.44	2.34
+ PMF-CEC	100	**4.66** / 16.13	**+ 0.04**
	300	4.69 / 16.36	+ 0.13
	500	4.72 / 16.45	+ 0.34
	1000	4.78 / 16.71	+ 0.53
MaLa-ASR [83]	100	5.87 / **5.75**	3.52
	300	10.03 / 12.59	5.92
	500	31.61 / 44.02	7.93
	1000	116.99 / 100.00	10.57

show a slight upward trend. Nevertheless, even under this challenging setting, PMF-CEC maintains relative reductions of 34.81% and 20.92% in B-WER compared with the original ASR transcripts, thereby confirming the resilience of our model to large-scale rare word lists.

We also investigated the scenario in which no relevant rare word list is provided. On the *test-clean* set, the absence of a rare word list caused WER to increase to 3.69%. However, PMF-CEC still achieved a relative WERR of 7.75% compared with the original ASR transcripts. This demonstrates that our model can partially correct errors by generating plausible words through the decoder, even without explicit contextual guidance. Nonetheless, accurate correction of rare or domain-specific terms ultimately requires the availability of a relevant rare word list.

Finally, to further evaluate the model's robustness, we conducted an "anti-context" experiment in which the rare word list consisted of 100 irrelevant distractors. On the *test-clean* set, this setting yielded a WER of 3.72%, corresponding to a 7.00% relative WERR compared with the original ASR outputs. These findings highlight that PMF-CEC does not naively rely on the provided rare word list; rather, it effectively suppresses irrelevant contextual terms and instead generates more appropriate corrections using its decoder.

Table 3.10: Impact of each module in PMF-CEC evaluated via ablation experiments on Librispeech test-clean set (%). Accuracy refers to the AED module's detection accuracy.

Method	WER ↓	B-WER ↓	AED Accuracy ↑
PMF-CEC (full)	**2.52**	**8.54**	**97.40**
w/o Phoneme Encoder	2.62	9.55	97.40
w/o Context Decoder	3.18	12.13	97.40
w/o RPM	2.61	8.78	95.73

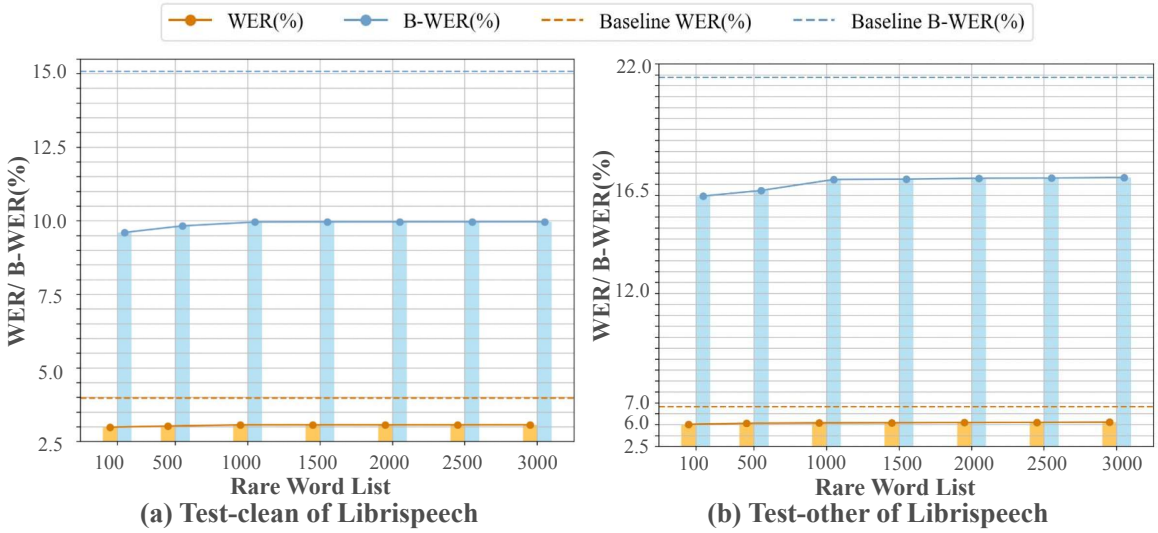
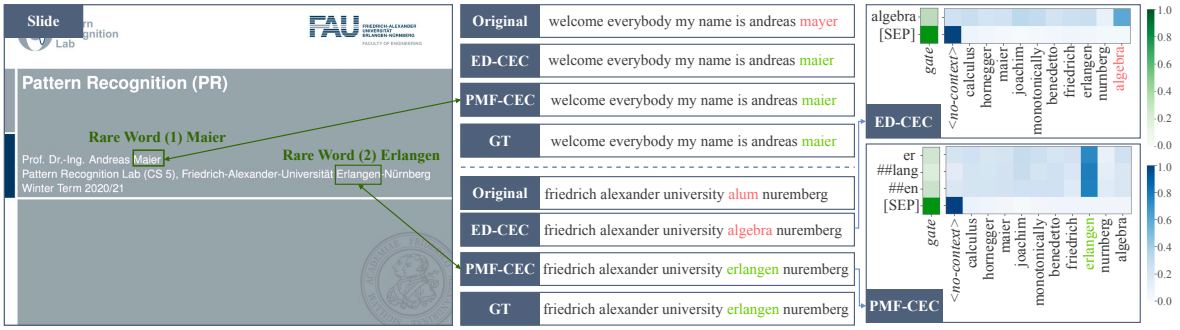


Figure 3.6: WER results of Librispeech test sets with varying rare word list sizes. The baseline corresponds to the original ASR texts without applying AEC.

3.4.6 Few-shot Generalization

To further evaluate the adaptability of the proposed approach, we examined the few-shot generalization capability of PMF-CEC, with results summarized in Table 3.11. In this evaluation, "0-shot" denotes rare words that were completely unseen during training, whereas "5-shot" and "100-shot" correspond to rare words that appeared five and one hundred times, respectively, in the training corpus. Here, the term "n-shot" is used to indicate the number of occurrences in the training data, rather than the conventional few-shot inference setting. Rare words were grouped according to their frequency of occurrence, and performance was measured in terms of B-WER and RW-Recall. Specifically, a rare word was considered successfully corrected if it appeared in the corrected transcript.

The baseline "Original" system refers to ASR outputs without any AEC model applied; there-



(a) Slide with Highlighted Rare Words (b) Comparison of AEC Results (c) Attention Visualization of Correcting "erlangen"

Figure 3.7: Examples of correcting the rare words "maier" and "erlangen" in the PRLVS dataset. We present slides with the rare words, the original ASR transcription, the corrected results using the ED-CEC method, the corrected results using our PMF-CEC method, and the ground truth (GT) transcription. Errors are marked in red, and correctly corrected parts are highlighted in green. Additionally, we provide heatmaps illustrating the correction process for the rare word "erlangen" using both methods.

Table 3.11: Analysis of few-shot generalization ability on the Librispeech test-clean and test-other and PRLVS test set. The results show B-WER ↓ / RW-Recall ↑ (%).

Model	Librispeech Test-clean			Librispeech Test-other			PRLVS Test Set		
	≤ 0-shot	≤ 5-shot	≤ 100-shot	≤ 0-shot	≤ 5-shot	≤ 100-shot	≤ 0-shot	≤ 5-shot	≤ 100-shot
Original	50.91/46.05	32.94/65.66	13.43/87.21	67.30/32.58	51.94/45.66	25.22/73.46	64.83/40.58	41.51/60.16	17.32/80.34
ED-CEC	38.43/62.09	25.82/73.14	7.87/92.78	57.63/46.30	44.92/57.12	20.03/79.57	56.10/49.33	30.12/69.98	13.01/86.73
PMF-CEC	**35.72/69.11**	**23.37/78.26**	**7.52/95.75**	**56.13/48.31**	**43.84/58.45**	**19.57/80.84**	**55.79/50.62**	**27.83/72.55**	**12.42/88.25**

fore, its B-WER and RW-Recall values vary according to the frequency of rare words. On both the *LibriSpeech* and *PRLVS* test sets, ED-CEC and PMF-CEC exhibit the ability to handle unseen or infrequently observed rare words, thereby demonstrating strong zero-shot and few-shot learning capabilities. Furthermore, across all shot conditions, PMF-CEC consistently outperforms ED-CEC, highlighting the benefit of incorporating phoneme-augmented multimodal fusion for improving generalization. These findings confirm that the integration of phonetic information enhances the robustness of rare-word correction, even under data-scarce scenarios.

3.4.7 Domain Adaptation with Limited Data

To further assess the applicability of PMF-CEC in resource-constrained domains, we performed domain adaptation experiments by extracting 10% of the training data from the DATA2 and PRLVS corpora to finetune models initially trained on the 960-hour *LibriSpeech* dataset. The experimental results are summarized in Table 3.12. We considered three training strategies:

Table 3.12: Results of domain adaptation with limited data on PRLVS and DATA 2 (%).

Method	PRLVS		DATA2	
	WER ↓	B-WER ↓	WER ↓	B-WER ↓
Original	6.74	19.83	4.96	15.22
PT	6.89	18.33	5.12	13.78
FT (Full Data) ◇	5.99	12.51	3.31	8.82
PT + FT (10% Data)	6.29	14.79	3.17	7.63
PT + FT (Full Data)	**5.40**	**10.27**	**2.72**	**6.35**

◇ Technically, since there is no PT on Librispeech, it is not appropriate to use the term "FT" as the model is directly trained on PRLVS or DATA2 full training data. However, we keep "FT" here for consistency.

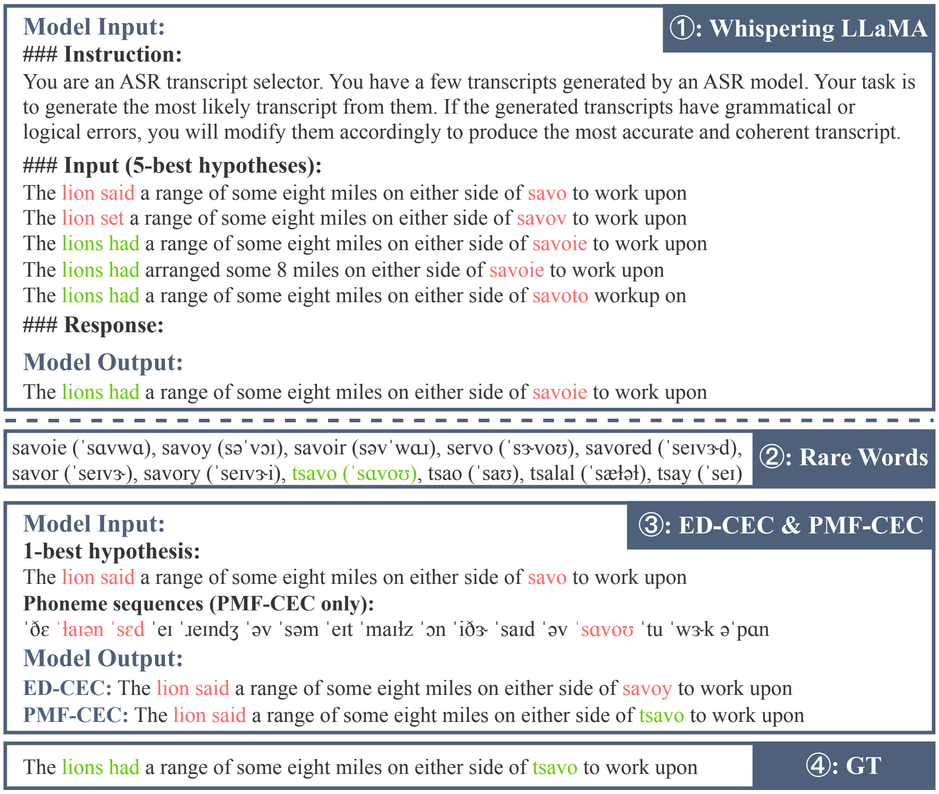
- **FT (Full Data):** training directly from scratch using the complete dataset.
- **PT + FT (10% Data):** pretraining on *LibriSpeech* followed by finetuning with only 10% of the domain-specific data.
- **PT + FT (Full Data):** pretraining on *LibriSpeech* followed by finetuning with the entire dataset.

The results reveal several important observations. First, the "PT + FT (10% Data)" strategy yielded lower WER and B-WER than "FT (Full Data)" on DATA2, while slightly higher values were observed on PRLVS. Nevertheless, both strategies substantially outperformed the original ASR transcripts, confirming that even limited finetuning data can provide significant benefits when adapting pretrained models to new domains.

Second, the "PT + FT (Full Data)" strategy achieved the best overall performance across both datasets, demonstrating that pretraining on large-scale data combined with sufficient domain-specific finetuning maximizes model adaptation capability.

Finally, the "PT" setting, where the pretrained *LibriSpeech* model was directly evaluated on DATA2 and PRLVS test sets without finetuning, resulted in a slight increase in WER compared with the original ASR outputs. This suggests that while large-scale pretraining provides a strong initialization, domain-specific finetuning—even with limited data—is essential for capturing local error patterns and improving correction effectiveness.

In summary, these findings validate the effectiveness of PMF-CEC in low-resource scenarios. The combination of large-scale pretraining and targeted finetuning offers a practical and efficient



①: Whispering LLaMA

Model Input:
Instruction:
You are an ASR transcript selector. You have a few transcripts generated by an ASR model. Your task is to generate the most likely transcript from them. If the generated transcripts have grammatical or logical errors, you will modify them accordingly to produce the most accurate and coherent transcript.

Input (5-best hypotheses):
The lion said a range of some eight miles on either side of savo to work upon
The lion set a range of some eight miles on either side of savov to work upon
The lions had a range of some eight miles on either side of savoie to work upon
The lions had arranged some 8 miles on either side of savoie to work upon
The lions had a range of some eight miles on either side of savoto workup on

Response:

Model Output:
The lions had a range of some eight miles on either side of savoie to work upon

②: Rare Words

savoie (ˈsɑvwɑ), savoy (səˈvɔɪ), savoir (səvˈwɑɹ), servo (ˈsɝvoʊ), savored (ˈseɪvɝd), savor (ˈseɪvɝ), savory (ˈseɪvɝi), tsavo (ˈsɑvoʊ), tsao (ˈsaʊ), tsalal (ˈsæłəł), tsay (ˈseɪ)

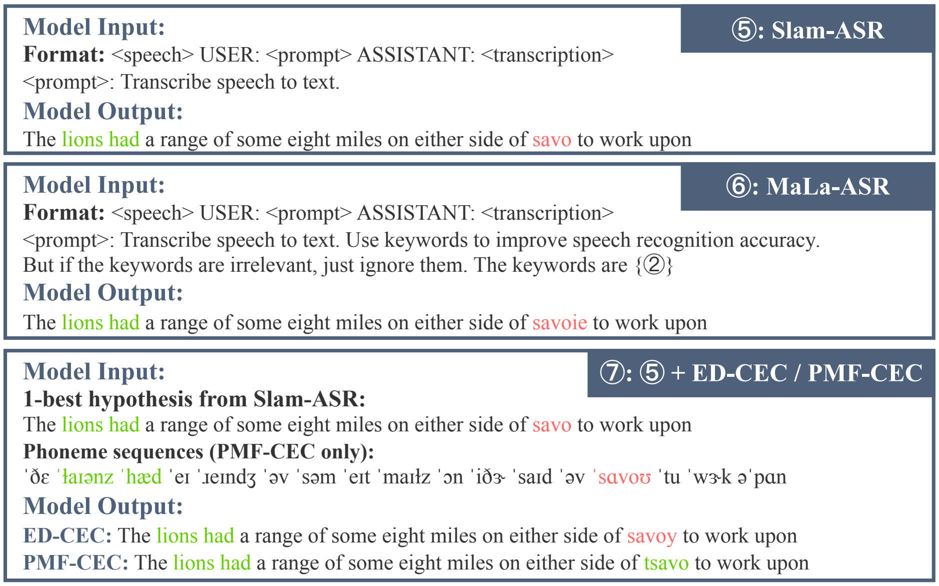
③: ED-CEC & PMF-CEC

Model Input:
1-best hypothesis:
The lion said a range of some eight miles on either side of savo to work upon
Phoneme sequences (PMF-CEC only):
ˈðɛ ˈłaɪən ˈsɛd ˈeɪ ˈɹeɪndʒ ˈəv ˈsəm ˈeɪt ˈmaɪłz ˈɔn ˈiðɝ ˈsaɪd ˈəv ˈsɑvoʊ ˈtu ˈwɝk əˈpɑn
Model Output:
ED-CEC: The lion said a range of some eight miles on either side of savoy to work upon
PMF-CEC: The lion said a range of some eight miles on either side of tsavo to work upon

④: GT

The lions had a range of some eight miles on either side of tsavo to work upon

(a) Comparison of Whispering LLaMA and PMF-CEC on Rare Word Correction ("tsavo")



⑤: Slam-ASR

Model Input:
Format: <speech> USER: <prompt> ASSISTANT: <transcription>
<prompt>: Transcribe speech to text.
Model Output:
The lions had a range of some eight miles on either side of savo to work upon

⑥: MaLa-ASR

Model Input:
Format: <speech> USER: <prompt> ASSISTANT: <transcription>
<prompt>: Transcribe speech to text. Use keywords to improve speech recognition accuracy. But if the keywords are irrelevant, just ignore them. The keywords are {②}
Model Output:
The lions had a range of some eight miles on either side of savoie to work upon

⑦: ⑤ + ED-CEC / PMF-CEC

Model Input:
1-best hypothesis from Slam-ASR:
The lions had a range of some eight miles on either side of savo to work upon
Phoneme sequences (PMF-CEC only):
ˈðɛ ˈłaɪənz ˈhæd ˈeɪ ˈɹeɪndʒ ˈəv ˈsəm ˈeɪt ˈmaɪłz ˈɔn ˈiðɝ ˈsaɪd ˈəv ˈsɑvoʊ ˈtu ˈwɝk əˈpɑn
Model Output:
ED-CEC: The lions had a range of some eight miles on either side of savoy to work upon
PMF-CEC: The lions had a range of some eight miles on either side of tsavo to work upon

(b) Comparison of Mala-ASR and PMF-CEC on Rare Word Correction ("tsavo")

Figure 3.8: An example from the DATA2 test set illustrating the challenge of rare word recognition under homophone confusion. Red indicates errors; green highlights correct recovery.

pathway for domain adaptation, ensuring robust performance even when training resources are scarce.

3.4.8 Examples of Correcting Rare Words

To further illustrate the practical benefits of PMF-CEC, we present qualitative examples of correcting rare words in Figs. 3.7 and 3.8.

Fig. 3.7 shows two cases involving the rare words "maier" and "erlangen." In the first case, the misrecognized word "mayer" shares a high degree of orthographic similarity with the ground-truth word "maier." Both ED-CEC and PMF-CEC successfully identify and select the correct item from the rare word list for correction. In the second case, however, the ground-truth word "erlangen" is misrecognized as "alum." Despite their substantial difference in spelling, the two words are acoustically similar. Here, ED-CEC incorrectly selects "algebra" from the rare word list due to its orthographic similarity to "alum," leading to a failed correction. In contrast, PMF-CEC successfully identifies "erlangen" by leveraging phoneme-level information in addition to contextual cues, thereby demonstrating the importance of phonetic representations for resolving homophone-induced errors.

While the examples in Fig. 3.7 are derived from lecture slides where phonetic ambiguity is relatively limited, Fig. 3.8 presents a more challenging case explicitly designed to evaluate the handling of homophonic confusion. This example involves the rare named entity "tsavo," which is frequently misrecognized as the more common word "savo," despite their identical pronunciation. In Fig. 3.8(a), both Whispering LLaMA and ED-CEC fail to resolve this error. By contrast, PMF-CEC correctly identifies "tsavo" from the rare word list by exploiting phoneme-level representations and contextual bias. Fig. 3.8(b) further compares the MaLa-ASR method, which fails to correct the same error even though "tsavo" is explicitly provided in the prompt, underscoring the limitations of text-only prompting in the presence of phonetic ambiguity. In contrast, PMF-CEC, as a lightweight postprocessing model, accurately corrects the error directly on the 1-best output from SLAM-ASR, striking an effective balance between correction accuracy and computational efficiency.

3.4.9 Discussion

Impact of Text Normalization.

An important factor in evaluating contextual biasing methods is the role of text normalization applied to both reference and hypothesis transcripts. Table 3.13 summarizes the word error rate (WER) and biased word error rate (B-WER) results on the LibriSpeech *test-clean* and *test-other* datasets under different normalization settings.

As shown in Table 3.13, applying text normalization consistently reduces both WER and B-WER across all configurations. For example, on the *test-clean* dataset, the WER of the Whisper baseline decreases from 4.00% to 3.02%, while the corresponding B-WER drops from 15.08% to

Table 3.13: WER and B-WER on LibriSpeech test-clean and test-other sets using Whisper medium.en and PMF-CEC models, combined with **text normalization** (%).

Method	Test-clean		Test-other	
	WER ↓	B-WER ↓	WER ↓	B-WER ↓
Whisper	4.00	15.08	6.83	21.89
+ Text norm.	3.02	9.48	5.84	19.88
Whisper + PMF-CEC	2.99	9.61	6.02	16.45
+ Text norm.	**2.85**	**8.58**	**5.78**	**15.96**

9.48%. When combined with PMF-CEC, the WER is further reduced to 2.85% and the B-WER to 8.58%. These results demonstrate that normalization contributes to improved apparent performance.

However, such improvements may not faithfully reflect the true effectiveness of contextual biasing. This is because normalization operations—such as handling possessive contractions, converting numerical expressions, or standardizing spelling variations (e.g., "fiber" vs. "fibre")—can inadvertently correct rare or domain-specific terms without the intervention of the biasing model. Consequently, normalization may obscure the actual contribution of contextual methods by artificially lowering error rates.

For a fair and comprehensive evaluation of contextual biasing systems, it is therefore essential to report results both with and without text normalization. While normalization improves readability and reduces superficial discrepancies, unbiased comparisons of contextual biasing approaches should also account for raw outputs to fully capture improvements in rare word recognition.

Phoneme Integration Beyond Postprocessing: Opportunities in LLM Decoding.

The PMF-CEC framework integrates phoneme information into the context-aware error correction process by jointly encoding textual and phonetic representations through the Context Encoder and the Context Phoneme Encoder. The resulting multimodal representations are fused to enhance the model's capability of detecting and correcting homophone-related errors in ASR transcripts. Despite these advantages, the current design is still situated within a postprocessing paradigm: phoneme-level information is incorporated only after the ASR output has been produced, and its role is limited to editing or refining the generated text.

A key limitation of this paradigm is that phoneme embeddings are not directly involved in the decoding dynamics of the language model. Consequently, their influence on sequence generation is

delayed, and they cannot interact with semantic modeling in real time. While such modularity is beneficial from an engineering perspective—ensuring compatibility with existing ASR pipelines—it constrains the deeper integration of phonetic cues into linguistic reasoning during decoding.

Looking forward, a promising research direction is to integrate phoneme embeddings directly into the decoding process of LLMs. This would allow the model to dynamically access and exploit phonetic information at each generation step, thereby facilitating more accurate contextual reasoning and improved disambiguation of phonetically similar terms. Potential strategies to achieve this include embedding-level fusion, cross-modal attention mechanisms, and phoneme-guided prompting. These approaches will be systematically explored in future work to further advance phoneme-aware ASR correction.

The Continued Relevance of Seq2Seq (S2S) AEC in the Era of LLMs.

In comparison with LLM-based AEC and contextual biasing approaches, sequence-to-sequence (S2S) AEC models, such as PMF-CEC, provide a lightweight and computationally efficient alternative for contextual error correction tasks. These models are readily compatible with conventional 1-best ASR outputs, thereby eliminating the need for prompt engineering and avoiding the substantial computational overhead associated with generative inference.

Furthermore, S2S models (e.g., ED-CEC and PMF-CEC) exhibit strong robustness when operating with large-scale biasing lists, in contrast to prompt-driven LLM approaches (e.g., MaLa-ASR), which often experience severe degradation in performance as the biasing list expands. This robustness underscores the practical scalability of S2S-based correction methods.

Finally, it is noteworthy that integrating the S2S-based PMF-CEC with generative AEC frameworks yields additional performance gains. Such hybrid strategies combine the fine-grained phoneme-aware modeling of PMF-CEC with the expressive semantic rewriting capabilities of LLMs, thereby achieving complementary improvements and opening avenues for future research on hybrid correction architectures.

3.5 Summary

In this chapter, we extended our previous work on context-aware ASR error correction and introduced a phoneme-augmented multimodal fusion framework, termed PMF-CEC. By integrating phonetic information into the correction pipeline through multimodal fusion, PMF-CEC is able to more effectively resolve homophone-induced errors in ASR transcripts, where acoustically similar words differ in their orthographic forms. In addition, we proposed a retention probability mechanism to enhance

the reliability of the error detection module by filtering out low-confidence editing operations.

Extensive experiments conducted on multiple benchmark datasets demonstrated that PMF-CEC consistently outperforms both our earlier ED-CEC model and other representative contextual biasing approaches, yielding substantial reductions in WER and B-WER while maintaining competitive inference speed. Furthermore, despite the increasing prevalence of LLM-based ASR frameworks, the lightweight and efficient nature of PMF-CEC highlights its continued relevance. The model can be seamlessly deployed as a postprocessing component without requiring retraining or architectural modifications to existing ASR systems, including LLM-based architectures. This design makes PMF-CEC particularly well-suited for latency-sensitive, real-time applications where both accuracy and efficiency are of paramount importance.

Chapter 4

PARCO: End-to-End Phoneme-Augmented Robust Contextual ASR via Contrastive Entity Disambiguation

In the previous chapter, we introduced PMF-CEC, a phoneme-augmented multimodal fusion framework for context-aware error correction of ASR outputs. Although PMF-CEC effectively improves rare-word recognition and mitigates homophone errors in a postprocessing paradigm, it still operates on the outputs of ASR systems rather than directly influencing the decoding process itself. This motivates the development of a more integrated approach that addresses the limitations of context-aware ASR at the source.

In this chapter, we present PARCO, an E2E phoneme-augmented robust contextual ASR framework designed to enhance recognition of domain-specific named entities, particularly under challenging homophone and distractor-heavy conditions. The goal is to achieve robust and accurate entity recognition within the ASR decoding process, thereby reducing errors that conventional contextual biasing or postprocessing methods fail to resolve. Specifically, PARCO incorporates (1) phoneme-aware encoding to better capture fine-grained pronunciation variations, (2) contrastive entity disambiguation to improve discrimination among similar-sounding entities, (3) entity-level supervision to ensure complete retrieval of multi-token entities, and (4) hierarchical entity filtering to suppress false positives under uncertainty.

Experiments across both Chinese and English benchmarks demonstrate that PARCO (1) substantially reduces recognition errors of rare and domain-specific entities, achieving a CER of 4.22%

on AISHELL-1 and a WER of 11.14% on DATA2 with 1,000 distractors, and (2) generalizes effectively to out-of-domain datasets such as THCHS-30 and LibriSpeech, highlighting its robustness and scalability.

4.1 Introduction

Although the previous chapter introduced PMF-CEC, a phoneme-augmented postprocessing method for context-aware error correction, its design still follows a pipeline paradigm: contextual information is applied only after the ASR output is generated. Such post-hoc correction effectively improves rare-word recognition but does not influence the E2E decoding process itself, limiting the degree of contextual integration.

In this chapter, we shift our focus to E2E contextual biasing, where contextual knowledge is directly incorporated into the ASR model during decoding. E2E approaches enable tighter integration of external context with acoustic and linguistic modeling, but they face two key challenges: (i) entities are often treated as sequences of independent tokens, leading to fragmented or incomplete recognition of multi-token entities; and (ii) distinguishing phonetically similar entities remains difficult, especially under open-domain or low-resource conditions.

To overcome these challenges, we propose **PARCO** (**P**honeme-**A**ugmented robust contextual ASR via **CO**ntrastive entity disambiguation), a novel E2E contextual biasing framework designed to improve both the accuracy and robustness of named entity recognition in ASR. The design of PARCO is driven by three major considerations: (1) enhancing phonetic discrimination for homophone entities, (2) ensuring complete recognition of multi-token entities as unified semantic units, and (3) reducing false positives in open-domain or noisy conditions. To achieve these goals, PARCO incorporates four key components:

- **Phoneme-aware encoding.** A dedicated phoneme encoder augments textual entity representations with fine-grained phonetic cues, helping the model resolve ambiguities among entities with similar pronunciations.
- **Contrastive Entity Disambiguation (CED).** A novel contrastive learning objective explicitly encourages the decoder to differentiate between phonetically similar but semantically distinct entities, thereby improving robustness against homophone confusion.
- **Entity-level supervision.** Instead of treating entities as isolated tokens, PARCO introduces entity-level supervision that guides the decoder to generate entire entity spans, avoiding fragmented or incomplete entity recognition.

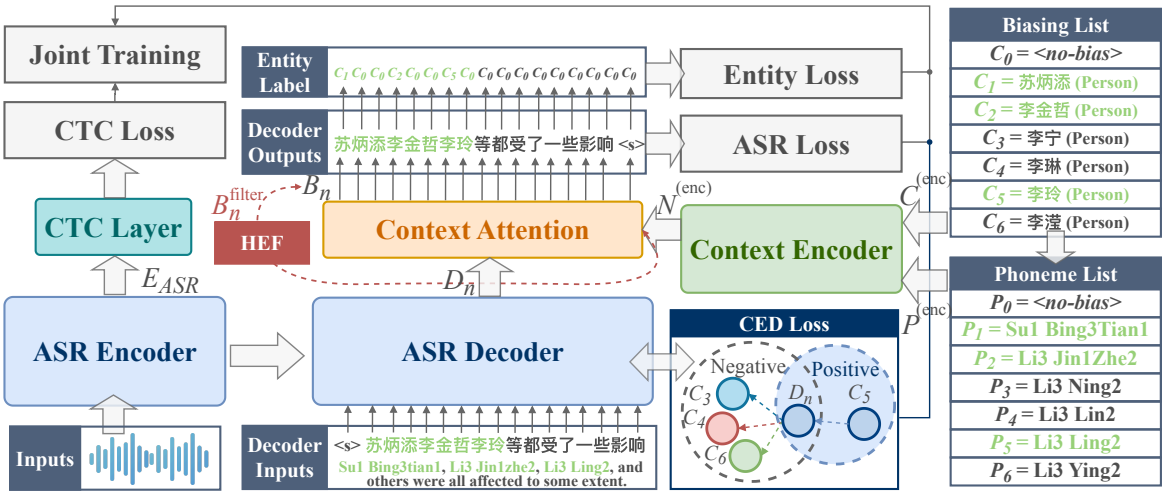


Figure 4.1: Overall architecture of the proposed PARCO model. The ASR backbone is based on a Conformer encoder-decoder architecture. A biasing list containing multi-token entities is encoded by a phoneme-enriched text encoder to support robust contextual biasing. The contrastive entity disambiguation (CED) loss enhances discriminability among phonetically similar entities and is detailed in Section 4.3.3. The hierarchical entity filtering (HEF) strategy, used only during inference, dynamically refines the biasing list for improved precision under ambiguity, as described in Section 4.3.5.

- **Hierarchical Entity Filtering (HEF).** During inference, a two-stage filtering strategy dynamically prunes the biasing list based on phoneme similarity and confidence gating, reducing spurious matches while retaining true entity candidates.

Together, these components enable PARCO to achieve robust entity recognition under diverse conditions, improving both error rates (CER/WER) and rare-word recall while maintaining scalability with large biasing lists.

4.2 AED-based ASR Model in PARCO

In the proposed PARCO framework, the underlying automatic speech recognition backbone is built upon an AED architecture, a widely adopted paradigm in E2E ASR. As illustrated in Fig. 4.1, the model is composed of two primary components: an audio encoder and an attention-based decoder.

The encoder transforms the input acoustic sequence S into a series of latent representations $E_{\text{ASR}} = (e_1, e_2, \ldots, e_M) \in \mathbb{R}^{M \times d}$, where M denotes the length of the encoded sequence and d represents the hidden feature dimension.

At decoding step n, the decoder generates a hidden state $D_n \in \mathbb{R}^d$, conditioned on both the previously generated tokens $T_{1:n-1} = (t_1, \ldots, t_{n-1})$ and the encoder output E_{ASR}. The probability

distribution over the vocabulary is then obtained via:

$$\hat{t}_n = \text{Softmax}(D_n W_n + b_n), \tag{4.1}$$

where $W_n \in \mathbb{R}^{V \times d}$ and $b_n \in \mathbb{R}^V$ are learnable parameters, and V denotes the vocabulary size. Here, $\hat{t}_n$ corresponds to the posterior probability $p(t_n | T_{1:n-1}, S)$.

The model parameters are optimized by minimizing the sequence-level negative log-likelihood:

$$\mathscr{L}_{\text{ASR}} = -\sum_{n=1}^{N} \log p(t_n | T_{1:n-1}, S), \tag{4.2}$$

where N is the length of the target transcription.

Furthermore, following the hybrid CTC/attention framework [132, 133], an auxiliary CTC loss term $\mathscr{L}_{\text{CTC}}$ is incorporated to encourage monotonic alignments and improve both convergence speed and recognition accuracy.

4.3 Proposed PARCO Method

As shown in Fig. 4.1, PARCO consists of an AED ASR model, a context encoder, and a context attention module, enhancing biasing for phonemically similar entities.

4.3.1 Context Encoder

The context encoder is designed to represent entities in the biasing list by jointly modeling their textual and phonetic information. This dual encoding enables the system to distinguish phonetically similar entities while preserving semantic integrity. The module consists of three components: a text encoder, a phoneme encoder, and a fusion mechanism.

Text Encoder. Let the biasing list be denoted as $C = (C_1, C_2, \ldots, C_L)$, where each entity C_l may consist of multiple tokens. Following the approach in [134], an additional "<no-bias>" token is included as C_0, allowing the model to determine when no entity should be selected. Each entity is encoded using stacked LSTM layers:

$$C_l^{(\text{enc})} = \text{LSTM}(C_l) \in \mathbb{R}^d. \tag{4.3}$$

The overall text-based entity representation is then obtained as

$$C^{(\text{enc})} = (C_0^{(\text{enc})}, C_1^{(\text{enc})}, \ldots, C_L^{(\text{enc})}) \in \mathbb{R}^{(L+1) \times d}. \tag{4.4}$$

Phoneme Encoder. In parallel, each entity is also represented at the phonetic level. Let $P = (P_1, P_2, \ldots, P_L)$ denote the phoneme sequences corresponding to the entities in C. These sequences are embedded using stacked LSTM layers to capture contextualized phonetic representations:

$$P^{(\text{enc})} = (P_0^{(\text{enc})}, P_1^{(\text{enc})}, \ldots, P_L^{(\text{enc})}) \in \mathbb{R}^{(L+1)\times d}. \tag{4.5}$$

Fusion of Text and Phoneme Representations. To obtain entity representations that are both semantically and phonetically enriched, the outputs of the text encoder and phoneme encoder are concatenated and projected into a shared space:

$$N^{(\text{enc})} = (C^{(\text{enc})} \oplus P^{(\text{enc})})W_N + b_N, \tag{4.6}$$

where $W_N \in \mathbb{R}^{2d\times d}$ and $b_N \in \mathbb{R}^d$ are learnable parameters, and "$\oplus$" denotes concatenation. The resulting representations $N^{(\text{enc})} = (N_0^{(\text{enc})}, N_1^{(\text{enc})}, \ldots, N_L^{(\text{enc})}) \in \mathbb{R}^{(L+1)\times d}$ are subsequently used in the decoding stage to perform phoneme-augmented contextual biasing.

4.3.2 Context Attention

The context attention mechanism enables the decoder to selectively attend to the most relevant entity representations from the biasing list at each decoding step. Specifically, the hidden state of the decoder at step n, denoted as D_n, is used as the query, while the phoneme-augmented entity embeddings $N^{(\text{enc})}$ serve as the keys.

The attention score for entity C_l at step n is calculated as:

$$s_{n,l} = \text{Softmax}\left(\frac{D_n W_Q \cdot (N_l^{(\text{enc})} W_K)^\top}{\sqrt{d}}\right) \in \mathbb{R}^{L+1}, \tag{4.7}$$

where $W_Q, W_K \in \mathbb{R}^{d\times d_h}$ are learnable projection matrices, and d_h denotes the attention dimension. The resulting distribution $s_{n,l}$ can also be interpreted as the probability of selecting entity C_l conditioned on the audio sequence and previously generated tokens, i.e., $p_c(C_l | T_{1:n-1}, S)$.

Next, the attention-weighted sum of the entity representations is computed to obtain the biasing vector B_n:

$$B_n = \sum_{l=0}^{L} s_{n,l} N_l^{(\text{enc})} W_V, \tag{4.8}$$

where $W_V \in \mathbb{R}^{d\times d_h}$ is a learnable parameter. Intuitively, B_n integrates contextual information derived from the biasing list and acts as an auxiliary signal to guide the decoder.

Finally, following [71, 74], we concatenate the decoder hidden state D_n with the biasing vector

B_n, and jointly use them to predict the next token:

$$\hat{t}_n = \text{Softmax}\left((D_n \oplus B_n)W_n + b_n\right), \tag{4.9}$$

where $W_n \in \mathbb{R}^{2d_h \times V}$ and $b_n \in \mathbb{R}^V$ are trainable parameters, and V is the vocabulary size. This formulation allows the decoder to dynamically balance acoustic evidence with contextual entity biasing, thereby improving the accuracy of named entity recognition.

4.3.3 Contrastive Entity Disambiguation (CED)

While the context attention mechanism (Section 4.3.2) enables the decoder to leverage biasing information, it often exhibits difficulties in distinguishing between entities that share highly similar pronunciations. This limitation becomes particularly pronounced in noisy or ambiguous acoustic conditions, where the attention weights may be dispersed across multiple confusable entities. Such phonetic ambiguity leads to uncertainty in entity selection and ultimately degrades recognition accuracy.

To mitigate this issue, we introduce a CED objective, designed to explicitly improve the discriminative power of the decoder's hidden representations. The central idea is to guide the decoder state at each step toward the correct entity representation while simultaneously pushing it away from phonetically similar but incorrect candidates.

Concretely, at decoding step n, let D_n denote the decoder hidden state, and let $N_p^{(\text{enc})}$ be the representation of the ground-truth biasing entity. We further define a set of I hard negatives, $\{N_{n_i}^{(\text{enc})}\}_{i=1}^{I}$, drawn from phonetically similar but semantically incorrect entities such that $n_i \neq p$. To enforce discrimination, we adopt a contrastive learning objective based on the InfoNCE loss:

$$\mathscr{L}_{\text{CED}} = -\log \frac{e^{\text{sim}(D_n, N_p^{(\text{enc})})/\tau}}{e^{\text{sim}(D_n, N_p^{(\text{enc})})/\tau} + \sum_{i=1}^{I} e^{\text{sim}(D_n, N_{n_i}^{(\text{enc})})/\tau}}, \tag{4.10}$$

where $\text{sim}(\cdot,\cdot)$ denotes cosine similarity and τ is a temperature hyperparameter controlling the sharpness of the distribution. By minimizing this loss, the decoder is encouraged to align more closely with the target entity while maintaining sufficient separation from acoustically confusable alternatives. The procedure for hard negative sampling is detailed in Section 4.4.3.

4.3.4 Joint Training

In prior studies [65, 76], entity supervision was typically implemented by assigning the same bias index to all constituent tokens of a given entity. This token-level alignment strategy, although straight-

forward, often fails to guarantee that the entire entity is consistently recognized during decoding, particularly for multi-token entities.

To illustrate this limitation, consider the example in Fig. 7.2 with the sentence: "苏炳添李金哲李玲等都受了一些影响" (*Su1 Bing3tian1, Li3 Jin1zhe2, Li3 Ling2, and others were all affected to some extent.*) Suppose the biasing list contains the entities "苏炳添" (C_1), "李金哲" (C_2), and "李玲" (C_5). The conventional approach would yield the following entity index sequence: $[C_1, C_1, C_1, C_2, C_2, C_2, C_5, C_5, C_0, C_0, \ldots]$, where every token within an entity is repeatedly labeled with the same index.

In contrast, our proposed entity-level supervision scheme adopts a more targeted labeling strategy. Specifically, the entity index is assigned only to the first token of each entity, while the subsequent tokens are marked with a <no-bias> index C_0: $[C_1, C_0, C_0, C_2, C_0, C_0, C_5, C_0, C_0, C_0, \ldots]$. This design explicitly enforces entity retrieval at the correct decoding step—namely, when the first token of the entity is generated—while suppressing redundant decisions for the remaining tokens. As a result, the model learns to decode complete entity spans rather than fragmentary or partial outputs, thereby improving entity-level consistency.

Formally, the entity supervision objective is defined as:

$$\mathscr{L}_{\text{Entity}} = -\sum_{n=1}^{N} \log p_c(\beta_n \mid T_{1:n-1}), \tag{4.11}$$

where $\beta_n \in (\beta_1, \beta_2, \ldots, \beta_L)$ denotes the constructed sequence of entity indices.

The overall training loss integrates four components:

$$\mathscr{L} = \lambda \mathscr{L}_{\text{ASR}} + (1-\lambda)\mathscr{L}_{\text{CTC}} + \mathscr{L}_{\text{Entity}} + \mathscr{L}_{\text{CED}}, \tag{4.12}$$

where $\mathscr{L}_{\text{ASR}}$ and $\mathscr{L}_{\text{CTC}}$ denote the standard objectives of the ASR model, and $\mathscr{L}_{\text{CED}}$ corresponds to the contrastive entity disambiguation loss described in Section 4.3.3. This joint optimization framework ensures that the model simultaneously learns accurate transcription, robust alignment, entity-level supervision, and fine-grained disambiguation.

4.3.5 Hierarchical Entity Filtering (HEF)

To further refine entity biasing during inference, we propose a HEF mechanism that combines phoneme-level similarity with confidence-based gating. The objective is to constrain the decoder's search space to a smaller, acoustically and semantically plausible candidate set, thereby reducing erroneous biasing without sacrificing recall.

Phoneme-Aware Pre-selection. At each decoding step n, the attention distribution $s_n \in \mathbb{R}^{L+1}$ over the biasing list is first computed as defined in (4.7). The entity index with the highest attention score is identified as:

$$\hat{l} = \arg\max_{l} s_{n,l}. \tag{4.13}$$

If the selected entity $C_{\hat{l}}$ corresponds to a valid candidate (i.e., not the special "<no-bias>" token), we regard it as the anchor entity at step n. Using this anchor, we retrieve its top-K phonemically similar candidates from the biasing list via an edit-distance-based phoneme retrieval algorithm. This procedure ensures that acoustically confusable alternatives (e.g., "陈冠希 (Chen2 Guan4Xi1)" versus "陈观鑫 (Chen2 Guan1Xin1)") are retained as priority candidates.

We then form a refined biasing list by combining these retrieved entities with the special "<no-bias>" token:

$$B_n^{\text{filtered}} = \{C_0\} \cup \{C_l \mid l \in \text{Top-}K(C_{\hat{l}})\}, \tag{4.14}$$

where Top-$K(C_{\hat{l}})$ denotes the indices of the K most similar entities in phoneme space. This filtered candidate set is subsequently used to replace the full biasing list when computing the biasing representation at step n, thereby enforcing a more focused and phonetically informed context modeling process. Consequently, (4.9) is reformulated as:

$$\hat{t}_n = \text{Softmax}\big((D_n \oplus B_n^{\text{filtered}})W_n + b_n\big). \tag{4.15}$$

Confidence-Based Gating. To avoid erroneous biasing toward unreliable candidates, we further incorporate a confidence-gating mechanism. Specifically, if the maximum selection probability among all entities in the filtered set (excluding the "<no-bias>" token) falls below a predefined threshold σ, the model defaults to standard vocabulary generation. Formally:

$$\delta = \mathbb{I}\left[\max_{\beta_n \in B_n^{\text{filtered}},\ \beta_n \neq C_0} p_c(\beta_n \mid T_{1:n-1}) < \sigma\right], \tag{4.16}$$

where $\mathbb{I}[\cdot]$ is the indicator function. The entity selection probability is then defined as:

$$p_c(\beta_n \mid T_{1:n-1}) = \begin{cases} 1, & \text{if } \delta = 1 \text{ and } \beta_n = C_0 \\ 0, & \text{if } \delta = 1 \text{ and } \beta_n \neq C_0 \\ p_c(\beta_n \mid T_{1:n-1}), & \text{if } \delta = 0. \end{cases} \tag{4.17}$$

In summary, the HEF module integrates a two-stage refinement: (1) phoneme-aware candidate pre-selection, which restricts attention to acoustically plausible alternatives, and (2) confidence-based

Table 4.1: Statistics of the datasets used in this study. "Utt." refers to the number of utterances and "NE" to the number of named entities. For LibriSpeech, only the test-clean subset was employed.

Dataset	Train		Dev		Test	
	Utt.	NE	Utt.	NE	Utt.	NE
AISHELL-1	119919	14241	14326	2194	7176	1186
DATA2	64570	11858	3100	2568	3100	2508
THCHS-30	–	–	–	–	2495	268
LibriSpeech	–	–	–	–	2620	2130

gating, which suppresses unreliable predictions. Together, these mechanisms substantially mitigate false positives while maintaining robust recall performance.

4.4 Experimental Evaluations

4.4.1 Implementation Details

All experiments were conducted on a computing cluster equipped with 4 NVIDIA A100 GPUs (80 GB memory per GPU). The models were trained with a batch size of 64. The input speech signals were converted into 80-dimensional log-Mel filterbank features using a 25 ms window and a 10 ms frame shift. To reduce the sequence length, the acoustic features were passed through a two-dimensional convolutional subsampling module, which downsampled the temporal resolution by a factor of four. The resulting features were then linearly projected into a 256-dimensional representation.

The ASR backbone followed a joint CTC/attention-based Conformer architecture [41], consisting of 12 encoder layers with 4 attention heads and an output dimension of 256. The decoder comprised 4 layers with hidden states of 512 dimensions. For contextual biasing, both text and phoneme sequences were encoded using 3-layer LSTM networks, each with 512 hidden units. The weight of the CTC objective λ was fixed at 0.7. For the contrastive entity disambiguation loss introduced in Section 4.3.3, the temperature hyperparameter τ in Eq. (4.10) was set to 0.1.

Regarding the HEF strategy described in Section 4.3.5, the pool size for phonemically similar entities (K) and the confidence threshold parameter (σ) were determined empirically, and set to 20 and 0.9, respectively.

4.4.2 Experimental Conditions

Datasets. The proposed PARCO framework was primarily evaluated on two benchmark corpora: the Chinese AISHELL-1 dataset [135] and the English DATA2 dataset [123]. To further examine the generalization capability of the model across languages and domains, we additionally incorporated two evaluation-only corpora: the Chinese THCHS-30 dataset [136] and the English LibriSpeech test-clean set [121]. Comprehensive dataset statistics, including utterance counts and the number of named entities, are summarized in Table 4.1.

Evaluation Metrics. For the Chinese benchmarks, recognition accuracy was measured using the CER, while named entity recognition accuracy was assessed via NE-CER, following prior studies [137, 138]. For the English datasets, we adopted WER as the primary metric, complemented by NE-WER to specifically quantify performance on named entity transcription.

Baselines. To provide a fair and comprehensive comparison, three representative baseline systems were selected:

- **CBA** [139]: introduces a contextual bias attention mechanism within an E2E ASR model, leveraging decoder-side attention over bias phrase embeddings to better capture rare contextual entities.
- **CopyNE** [138]: incorporates a copying mechanism that retrieves entities directly from a pre-defined NE dictionary, thereby ensuring complete and accurate transcription of multi-token entities.
- **ED-CEC** [111]: implements a postprocessing-based contextual ASR correction framework, which detects transcription errors and performs context-aware correction using rare word lists, thereby improving recognition accuracy without modifying the base ASR model.

4.4.3 Biasing List Construction

Entity Selection. The construction of the biasing list was tailored to each dataset to ensure consistency across languages and domains. For AISHELL-1, we directly adopted the named entities released by Chen et al. [140]. For THCHS-30, entities corresponding to persons, locations, and organizations were automatically extracted using the `HanLP` toolkit[1]. For DATA2, we employed the annotated entities of persons, locations, and organizations available in the original corpus [123]. For LibriSpeech, the same three categories of entities were extracted using the `NuNER_Zero` model[2].

Following the practice of prior studies [134, 138], we augmented the biasing vocabulary during training by randomly sampling two- or three-character substrings (Chinese) or two- or three-word spans (English) from utterances without annotated entities. These substrings were retained as

[1]https://github.com/hankcs/HanLP

[2]https://huggingface.co/numind/NuNER_Zero

Table 4.2: Performance on AISHELL-1 and DATA2 under different numbers of distractors N. Each cell shows CER/NE-CER or WER/NE-WER. Values in parentheses denote relative NE-CER or NE-WER reduction (%).

Model	N=0	N=100	N=1000	N=5000
AISHELL-1 (CER ↓ (CERR ↑) / NE-CER ↓ (NE-CERR ↑))				
Conformer [41]	4.94 (–) / 9.60 (–)	4.94 (–) / 9.60 (–)	4.94 (–) / 9.60 (–)	4.94 (–) / 9.60 (–)
+ CBA [139]	4.57 (+7.49) / 5.36 (+44.17)	4.65 (+5.87) / 5.91 (+38.44)	4.86 (+1.62) / 6.82 (+28.96)	5.18 (-4.86) / 8.12 (+15.42)
+ CopyNE [138]	4.37 (+11.54) / 2.24 (+76.67)	4.48 (+9.31) / 2.97 (+69.06)	4.72 (+4.45) / 3.76 (+60.83)	5.05 (-2.23) / 4.80 (+50.00)
+ ED-CEC [111]	4.49 (+9.11) / 4.96 (+48.33)	4.54 (+8.10) / 5.23 (+45.52)	4.66 (+5.67) / 5.68 (+40.83)	4.81 (+2.63) / 6.31 (+34.27)
+ PARCO	**4.03 (+18.42) / 1.57 (+83.65)**	**4.04 (+18.22) / 1.69 (+82.40)**	**4.22 (+14.57) / 2.84 (+70.42)**	**4.45 (+9.92) / 3.56 (+62.92)**
DATA2 (WER ↓ (WERR ↑) / NE-WER ↓ (NE-WERR ↑))				
Conformer [41]	12.05 (–) / 26.64 (–)	12.05 (–) / 26.64 (–)	12.05 (–) / 26.64 (–)	12.05 (–) / 26.64 (–)
+ CBA [139]	11.12 (+7.72) / 18.78 (+29.50)	11.52 (+4.40) / 20.71 (+22.26)	12.28 (-1.91) / 25.46 (+4.43)	13.23 (-9.79) / 28.04 (-5.26)
+ CopyNE [138]	10.26 (+14.85) / 10.34 (+61.19)	10.54 (+12.53) / 12.57 (+52.82)	11.44 (+5.06) / 19.14 (+28.15)	12.31 (-2.16) / 26.60 (+0.15)
+ ED-CEC [111]	10.59 (+12.12) / 16.54 (+37.91)	10.66 (+11.54) / 16.87 (+36.67)	11.38 (+5.56) / 17.94 (+32.66)	11.67 (+3.15) / 20.76 (+22.07)
+ PARCO	**10.00 (+17.01) / 8.34 (+68.69)**	**10.17 (+15.60) / 9.47 (+64.45)**	**11.14 (+7.55) / 14.16 (+46.85)**	**11.49 (+4.65) / 17.15 (+35.62)**

pseudo-entities if they contained valid phoneme sequences. Phoneme sequences were generated using `pypinyin`[3] for Chinese and `g2pE`[4] for English, thereby ensuring a phoneme-aligned representation across all corpora.

Hard Negative Sampling. To improve discriminability during training, we incorporated phonetically similar but semantically incorrect entities as hard negatives. For each GT entity, 1–3 hard negatives were selected from the training vocabulary by computing phoneme-level edit distances and retrieving the most similar candidates with different meanings. This approach ensured that the negative samples were acoustically confusable but contextually irrelevant, thereby posing a challenging learning signal.

An illustrative example is provided in Fig. 7.2: for the GT entity "李玲 (Li3 Ling2, Person)", potential hard negatives include "李宁 (Li3 Ning2, Person)", "李琳 (Li3 Lin2, Person)", and "李滢 (Li3 Ying2, Person)". These distractors were incorporated into the training objective alongside GT entities, compelling the model to explicitly distinguish correct entities from similar-sounding alternatives.

During inference, biasing lists were constructed by combining the GT entities present in each utterance with a set of phonetically retrieved distractors, ensuring realistic evaluation under large and confusable entity sets.

[3]https://pypi.org/project/pypinyin

[4]https://github.com/Kyubyong/g2p

Table 4.3: Ablation study on AISHELL-1 and DATA2 (%). NE-CER / NE-WER are reported with relative error rate reduction (↑) in parentheses under 5,000 distractors. "w/o": "without", TE: Text Encoder, PE: Phoneme Encoder, CED: Contrastive Entity Disambiguation, HEF: Hierarchical Entity Filtering.

No.	Configuration	AISHELL-1	DATA2
1	**PARCO**	**3.56 (+62.92)**	**17.15 (+35.62)**
2	No.1 w/o HEF	4.07 (+57.60)	19.52 (+26.73)
3	No.2 w/o CED Loss	4.39 (+54.27)	20.10 (+24.55)
4	No.3 w/o Entity Loss	5.51 (+42.60)	21.25 (+20.23)
5	No.4 w/o PE	8.27 (+13.85)	28.15 (-5.67)
6	No.5 w/o TE	9.60 (–)	26.64 (–)

4.4.4 Results and Analysis

Comparative Performance under Varying Biasing List Sizes. Table 4.2 reports the comparative results of PARCO and baseline systems under different distractor sizes on AISHELL-1 and DATA2. Across all settings, PARCO consistently outperforms alternative approaches. On AISHELL-1, the proposed model achieves the lowest NE-CER in every distractor condition, with error rates of 1.57%, 1.69%, 2.84%, and 3.56% for $N = 0, 100, 1{,}000$, and 5,000, respectively. Similarly, on DATA2, PARCO yields the lowest NE-WER across all settings, reaching 8.34%, 9.47%, 14.16%, and 17.15% for the same values of N. Although the addition of distractors generally leads to performance degradation across all methods, PARCO demonstrates markedly higher robustness. For instance, compared with the strongest baseline ED-CEC, PARCO maintains superior accuracy even at $N = 5{,}000$, indicating its effectiveness in resisting interference from acoustically confusable entities. We attribute these gains primarily to the HEF mechanism, which prunes the candidate set by leveraging phonetic similarity and confidence gating, thereby suppressing spurious biasing.

Ablation Study. To further examine the contribution of individual components, we conducted an ablation study under the most challenging setting with 5,000 distractors (Table 4.3). Eliminating the HEF strategy results in significant performance degradation on both AISHELL-1 and DATA2, confirming its importance in mitigating false-positive entity predictions. Removing either the CED loss or the entity-level supervision also leads to consistent declines, underscoring their complementary role in strengthening entity-aware learning. The most severe performance degradation is observed when either the phoneme encoder or the text encoder is removed, highlighting the necessity of both

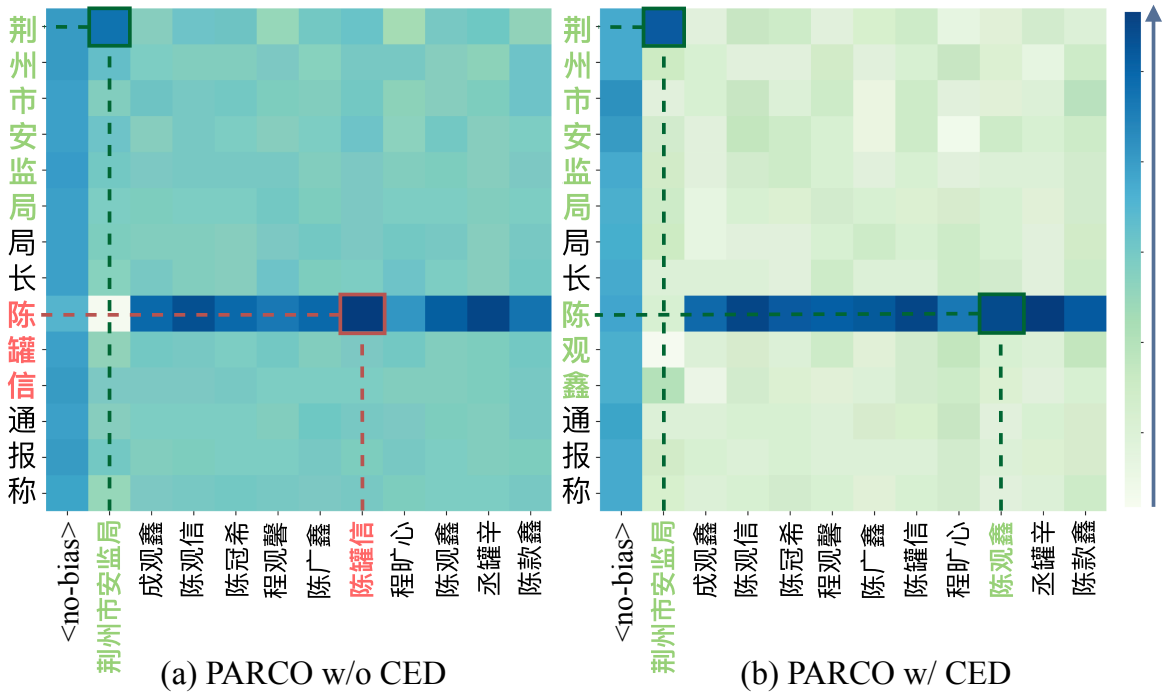
Table 4.4: Robustness analysis on OOD datasets (%). Values in parentheses denote the relative reduction compared with the Conformer baseline.

Model	THCHS-30		LibriSpeech test-clean	
	CER ↓	NE-CER ↓	WER ↓	NE-WER ↓
Conformer [41]	27.21 (–)	46.33 (–)	7.91 (–)	12.39 (–)
+ CBA [139]	28.55 (-4.92)	47.61 (-2.76)	9.68 (-22.38)	13.72 (-10.73)
+ CopyNE [138]	25.29 (+7.06)	15.65 (+66.22)	9.31 (-17.70)	10.93 (+11.78)
+ ED-CEC [111]	30.11 (-10.66)	50.52 (-9.04)	8.62 (-8.98)	14.38 (-16.06)
+ PARCO	**21.53 (+20.87)**	**8.48 (+81.70)**	**7.09 (+10.37)**	**6.17 (+50.20)**

phonetic and contextual information for accurate entity recognition. Taken together, these findings validate the overall architecture of PARCO and demonstrate that each module makes a non-trivial contribution to the final performance.

Robustness to Domain Shift. We further evaluated model robustness through cross-domain transfer experiments. Specifically, the AISHELL-1–trained model was tested on the out-of-domain THCHS-30 dataset, and the DATA2–trained model was evaluated on LibriSpeech. As presented in Table 4.4, baseline systems suffer from pronounced error increases when applied to unseen domains, reflecting their limited generalization capability. By contrast, PARCO achieves the lowest error rates in both transfer scenarios, reducing NE-CER by 81.70% on THCHS-30 and NE-WER by 50.20% on LibriSpeech compared with the base Conformer. These results highlight the strong generalization ability of the proposed framework, confirming its robustness to domain variability and its applicability to real-world ASR scenarios.

Qualitative Analysis. To further validate the effectiveness of the proposed CED loss, we designed a controlled evaluation scenario on AISHELL-1 involving ten candidate entities. The set comprises the ground-truth entity 陈观鑫 (Chen2 Guan1Xin1, Person) alongside nine carefully constructed hard negatives. These distractors were generated by modifying initials, finals, or nasal endings of the original name, thereby introducing fine-grained phonetic confusion while ensuring semantic irrelevance. The distractors include: 成观鑫 (Cheng2 Guan1Xin1, Person), 陈观信 (Chen2 Guan1Xin4, Person), 陈冠希 (Chen2 Guan4Xi1, Person), 程观馨 (Cheng2 Guan1Xin1, Person), 陈广鑫 (Chen2 Guang3Xin1, Person), 陈罐信 (Chen2 Guan4Xin4, Person), 程旷心 (Cheng2 Kuang4Xin1, Person), 丞罐辛 (Cheng2 Guan4Xin1, Person), 陈款鑫 (Chen2 Kuan3Xin1, Person).



(a) PARCO w/o CED (b) PARCO w/ CED

Figure 4.2: Comparison of attention visualization with and without CED loss. The horizontal and vertical axes are the biasing list and the transcript, respectively.

Fig. 4.2 presents the decoder's attention distribution over the biasing list, comparing results before and after applying the CED objective. As shown in subfigure (a), without CED supervision the model incorrectly allocates dominant attention to the distractor "陈罐信", which closely resembles the target entity phonetically. This indicates the inherent difficulty of disambiguating entities with nearly identical pronunciations under noisy acoustic conditions. In contrast, subfigure (b) illustrates that when trained with the CED loss, the decoder produces a sharper and more discriminative attention pattern. The attention is concentrated on the correct entity "陈观鑫", while competing distractors are effectively suppressed. These observations confirm that the contrastive objective strengthens the model's capacity to capture subtle phonetic distinctions and improves robustness in bias-aware decoding.

Table 4.5 provides additional transcription examples from AISHELL-1 and DATA2. While baseline systems frequently hallucinate entities or yield incomplete spans, PARCO consistently generates correct and complete entity sequences in both Chinese and English contexts. This qualitative evidence further highlights the model's ability to enforce entity integrity and maintain resilience against phonetically similar or contextually irrelevant distractors.

Table 4.5: Comparative examples of transcriptions generated by different models. Red text indicates errors, while green text highlights correctly transcribed entities.

AISHELL-1	
Entities:	罗宾索尔科维 (**Robin Szolkowy, Person**), 罗宾斯切姆贝拉 (**Robin Schembera, Person**)
Translation	*800 meters: Robin Schembera*
Ground Truth	八百米罗宾斯切姆贝拉
CBA [139]	八百米罗宾斯窃姆贝拉
CopyNE [138]	八百米罗滨斯切尔姆贝拉
ED-CEC [111]	八百米罗宾索尔科维姆被拉
PARCO	八百米罗宾斯切姆贝拉

DATA2	
Entities:	WM WALTERS CARPENTER'S, WM JONES
Ground Truth	WM WALTERS CARPENTER'S CREW WM JONES DOG DRIVER
CBA [139]	WILLIAM WALTIS CARPING HIS CREW WILLIAM JONES DON'T DRIVER
CopyNE [138]	WILLIAM WALTIS CARPENTIS CREW WILLIAM JONES DOG DRIVER
ED-CEC [111]	WILLIAM WALTERS CARPENTER'S CREW WILLIAM JONES DOG DRIVER
PARCO	WM WALTERS CARPENTER'S CREW WM JONES DOG DRIVER

4.5 Summary

In this chapter, we presented **PARCO**, a phoneme-augmented contextual ASR framework designed to address two key challenges in bias-aware speech recognition: (i) the disambiguation of homophones and (ii) the accurate recognition of multi-token entities. The proposed framework integrates four complementary components: (1) a phoneme-aware encoder that captures fine-grained pronunciation variations, (2) a contrastive entity disambiguation (CED) objective that strengthens discriminative decoding under phonetic ambiguity, (3) an entity-level supervision mechanism that enforces span-level consistency for multi-token entities, and (4) a hierarchical entity filtering (HEF) strategy that enhances retrieval precision and robustness during inference.

Extensive experiments demonstrate that PARCO consistently improves contextual biasing performance on both Chinese (AISHELL-1) and English (DATA2) benchmarks, achieving substantial gains over strong baselines. Furthermore, cross-domain evaluations on THCHS-30 and LibriSpeech confirm its robustness and generalization capability beyond in-domain training data.

Looking ahead, we plan to extend PARCO toward multilingual scenarios and to investigate its integration within retrieval-augmented speech understanding pipelines, thereby broadening its applicability to real-world ASR systems operating in diverse and dynamic environments.

Chapter 5

CMT-LLM: Contextual Multi-Talker ASR Utilizing Large Language Models

Although the previous chapter introduced PARCO to improve contextual ASR through phoneme-augmented entity disambiguation, its design primarily targeted single-speaker conditions. In real-world applications, however, speech often involves multiple overlapping speakers, making recognition substantially more challenging. At the same time, rare and domain-specific words continue to pose difficulties, especially when multiple candidate entities must be distinguished under noisy or ambiguous conditions. Addressing these two challenges—multi-talker recognition and contextual biasing—within a unified framework remains an open problem.

Recent advances in LLMs provide a promising solution. LLMs are equipped with powerful semantic reasoning and contextual modeling abilities, which can help disambiguate rare entities and resolve long-range dependencies that are difficult for traditional ASR architectures. However, directly applying LLMs to multi-talker ASR is computationally expensive and inefficient, particularly when handling large biasing lists. To fully leverage the strengths of LLMs while maintaining efficiency, careful integration with speech encoders and biasing mechanisms is required.

In this chapter, we present **CMT-LLM**, a contextual multi-talker ASR framework that unifies overlapping speech recognition and contextual biasing under a single model. Specifically, CMT-LLM combines pretrained speech encoders with LLMs using optimized finetuning strategies, and incorporates a two-stage filtering algorithm to efficiently select relevant rare words from large candidate sets. These selected terms are then embedded into the LLM's prompt input, enhancing rare-word recognition in multi-talker conditions.

Experiments across multiple benchmarks demonstrate that CMT-LLM outperforms traditional contextual biasing methods, achieving a WER of 7.9% on LibriMix and 32.9% on AMI SDM with

a biasing size of 1,000. These results highlight the effectiveness of integrating LLMs into contextual multi-talker ASR, offering a scalable and robust solution for complex conversational scenarios.

5.1 Introduction

ASR in multi-talker scenarios, particularly under overlapping speech conditions, remains an open and challenging problem. Existing approaches include PIT [6], heuristic error assignment methods [141], and SOT [7]. Among them, SOT has become widely adopted as it resolves the speaker assignment ambiguity by concatenating transcriptions in temporal order. Nevertheless, SOT places heavy demands on long-context modeling, where conventional AED models often struggle to capture complex inter-speaker dependencies.

A second major challenge for ASR systems is the accurate recognition of rare words such as proper nouns and technical terms, which are sparsely represented in training data [142]. To mitigate this, contextual biasing methods have been proposed. Shallow fusion techniques adjust decoding scores using external language models but suffer when dealing with large or dynamically changing biasing lists [143]. Deep biasing methods [144] incorporate contextual features into the decoder, but require retraining and structural modifications. Deep context models [145] utilize contextual text encoders at the cost of high computational overhead. Post-recognition error correction models [9, 111, 146] refine ASR outputs with contextual knowledge but inherently depend on initial hypotheses and introduce additional latency.

In practice, multi-talker ASR and contextual biasing challenges are tightly coupled. For example, in meeting transcription, the system must not only separate overlapping speech and track speaker turns but also correctly identify specialized terms and named entities. Similarly, in customer service applications, the ability to distinguish concurrent dialogues while biasing toward contextually relevant terms is crucial for downstream utility. Despite this strong interdependence, most prior research has treated multi-talker ASR and contextual biasing as distinct tasks. Motivated by this gap, we propose to integrate contextual biasing into multi-talker ASR, with the aim of improving recognition accuracy for rare and domain-specific words under overlapping speech conditions. To the best of our knowledge, this represents the first systematic effort to unify these two tasks.

LLMs provide a natural pathway toward this integration. With their powerful capacity for semantic reasoning and global context modeling, LLMs are particularly effective at capturing inter-speaker dependencies and producing coherent transcriptions in multi-talker conditions [147, 148]. In the context of contextual biasing, LLMs support prompt-based adaptation, which allows dynamic incorporation of user-provided biasing lists, thereby improving rare word recognition without requiring complex decoding modifications [149]. These capabilities highlight the unique advantages of LLMs

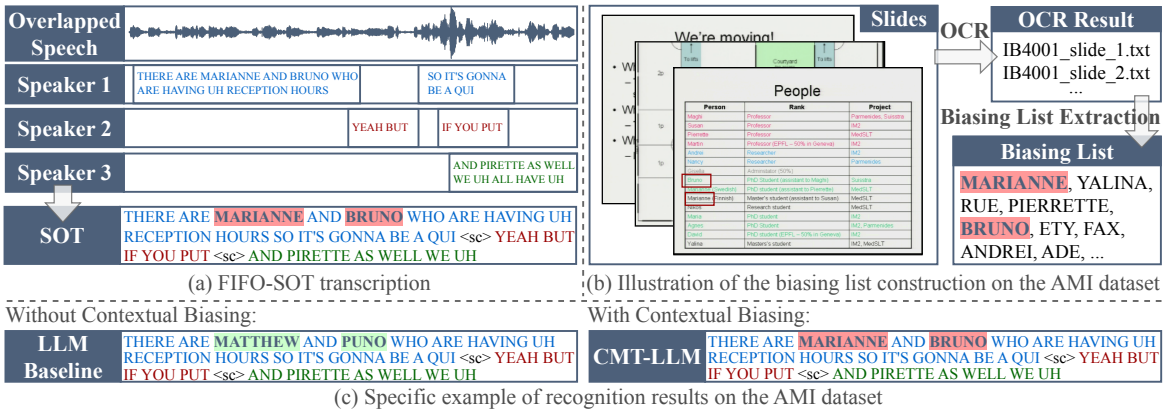


Figure 5.1: Overall pipeline of the proposed contextual multi-talker ASR framework integrating pretrained speech encoders, projectors, and LLMs.

in simultaneously addressing the challenges of multi-talker recognition and contextual biasing.

Building on these insights, we propose a unified framework, **CMT-LLM**, which integrates a pretrained speech encoder, a projection module, and a LLM, as illustrated in Fig. 5.1. To train the system efficiently, we adopt a two-stage finetuning strategy: (1) the SSL speech encoder is first adapted to overlapping speech using the SOT objective; (2) the adapted encoder and LLM are subsequently frozen, while the projector is trained and the LLM is further tuned through LoRA. In addition, to address the practical challenge of handling large-scale biasing lists (e.g., thousands of words), we introduce a two-stage filtering mechanism. A coarse decoding stage first selects a small subset of highly relevant rare words, which are then incorporated into the LLM prompt for fine-grained contextual biasing. This design enables efficient integration of contextual knowledge while preserving scalability and accuracy in multi-talker ASR.

5.2 Proposed Method

5.2.1 Problem Formulation

We define the contextual multi-talker ASR task as a mapping function $f(S,C) = T$, where the input speech $S = (S_1, S_2, \ldots, S_M)$ consists of M acoustic frames, and the contextual biasing list $C = (C_1, C_2, \ldots, C_L) \in \mathbb{R}^L$ contains L rare or domain-specific words. The desired output is the ground-truth transcript $T = (T_1^1, T_2^1, \ldots, \texttt{<sc>}, T_1^2, T_2^2, \ldots, \texttt{<sc>}, T_1^3, T_2^3, \ldots) \in \mathbb{R}^N$, represented as a token sequence of length N. Here, T_a^b denotes the a-th token uttered by the b-th speaker.

Following prior work on SOT [7, 57], overlapping multi-speaker speech is modeled by concatenating the transcriptions of individual speakers into a single output sequence. Speaker boundaries

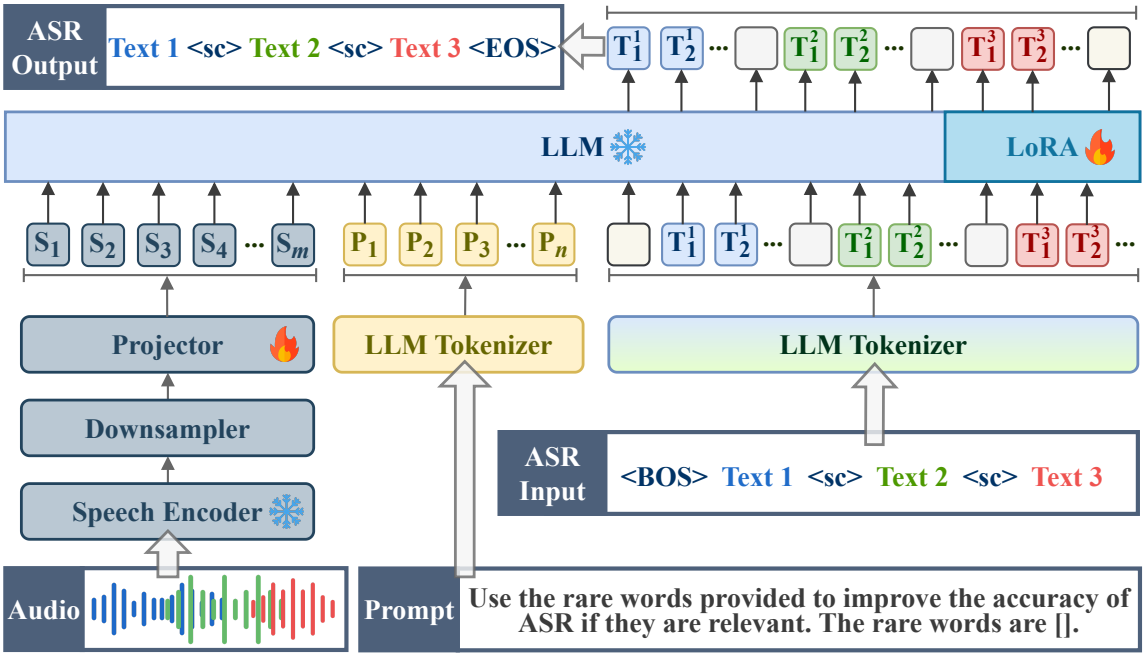


Figure 5.2: Overall architecture of the CMT-LLM model.

are explicitly marked with the special token "<sc>". For example, in the case of three concurrent speakers, the corresponding reference transcription is constructed by concatenating each speaker's utterances in first-in first-out (FIFO) order, determined by their respective speaking times. This serialization strategy is illustrated in Fig. 5.1(a), which provides the basis for handling multi-talker scenarios within the proposed contextual ASR framework.

5.2.2 Proposed CMT-LLM

We propose a contextual multi-talker ASR framework built upon LLMs, referred to as **CMT-LLM**. The overall architecture, shown in Fig. 5.2, follows the design paradigm of recent LLM-based approaches [21, 147–150], and consists of three major components: a speech encoder, a projection module, and an LLM decoder.

The speech encoder first processes the overlapping input signal and generates a sequence of acoustic representations $E_S \in \mathbb{R}^{M \times d_S}$, where d_S denotes the feature dimension. To reduce computational complexity and align with the input requirements of the LLM, we apply a one-dimensional convolutional downsampling layer, compressing the temporal resolution by a factor of n. The downsampled features are then transformed by a two-layer linear projection network into speech embeddings $\bar{E}_S \in \mathbb{R}^{\frac{M}{n} \times d_P}$, with d_P corresponding to the hidden size of the LLM. This ensures dimensional

compatibility between the speech encoder output and the LLM input space.

To address the contextual multi-talker recognition problem, we incorporate rare-word biasing lists into the LLM through prompt-based conditioning. The constructed prompt includes task-specific instructions as well as the biasing list, which is tokenized and encoded into a prompt embedding $E_P \in \mathbb{R}^{P \times d_P}$, where P represents the prompt length. The details of biasing list construction are provided in Section 5.3.2.

For training, we adopt the SOT strategy. The reference multi-talker transcript is tokenized and converted into embeddings $E_{ASR} \in \mathbb{R}^{N \times d_P}$, which are fed to the LLM alongside the speech embeddings $\bar{E}_S$ and the prompt embeddings E_P. The model is optimized to predict the transcription $\hat{T} \in \mathbb{R}^N$ by minimizing the cross-entropy loss between $\hat{T}$ and the ground-truth transcript T.

During inference, the LLM receives both the speech embeddings and the contextual prompt (instructions plus biasing list) and autoregressively generates the final transcription. To further assess the effect of contextualization, we also evaluate a variant of the model without biasing information in the prompt, denoted as the **LLM Baseline**.

The number of biasing terms included in the LLM prompt is restricted to 100 during training in order to keep computational costs tractable. The details of the biasing list construction process are provided in Section 5.3.2. In practical deployment, however, the biasing list may contain tens of thousands of candidate terms, which substantially complicates inference. Such large-scale lists can overwhelm the model's capacity to reliably identify relevant words, resulting in significant degradation of recognition performance.

To mitigate this limitation, we adopt a two-stage filtering strategy inspired by recent advances in contextual biasing for ASR [151–154]. In the first stage, a self-supervised pretrained speech encoder is finetuned on the target dataset with an additional CTC prediction head. Greedy decoding is then employed to produce preliminary transcription hypotheses with relatively low computational overhead. This step serves two purposes: (1) finetuning the pretrained encoder with a conventional objective has been shown to outperform direct use of the frozen encoder for downstream LLM-based ASR; and (2) the preliminary decoding results provide a coarse filtering mechanism, eliminating irrelevant candidates from the large biasing list.

In the second stage, the coarse CTC outputs are further processed to refine the biasing vocabulary. Common words are first removed by excluding the 5,000 most frequent terms, thereby retaining more informative rare candidates. For instance, consider the ground-truth transcription "... *MORE THAN THE SPEAKER CHARACTERISATION AS M STEVE* ..." and the CTC decoding output "... *MORE THAN THE SPEAKER CHARACE THSATION AS STEE* ...". After eliminating common words, the remaining uncertain tokens are "CHARACE THSATION" and "STEE." All possible sub-segments of these candidates are then enumerated, such as "CHARACE," "THSATION," and "CHARACE TH-

SATION." Each segment is compared against entries in the biasing list using word-level edit distance, and the closest match is selected—in this example, "CHARACTERISATION." Although phoneme-based edit distance and semantic similarity measures were also considered, they proved less efficient and, in practice, less effective than the simpler word-based approach. Moreover, restricting evaluation to only individual words (e.g., "CHARACE" and "THSATION") risks overlooking the correct entity. To balance efficiency and accuracy, each candidate segment is matched to its top-10 nearest neighbors in the biasing list. Duplicates are removed, and the resulting filtered set of rare words is incorporated into the LLM prompt.

This hierarchical filtering procedure reduces large-scale biasing lists from thousands of entries to a size that is computationally manageable by the LLM, while preserving the most relevant contextual candidates. As a result, inference efficiency and accuracy are both substantially improved, effectively addressing the performance bottleneck posed by large biasing lists in real-world contextual multi-talker ASR scenarios.

5.3 Experimental Evaluation

5.3.1 Implementation Details

All experiments were conducted on a cluster equipped with 4 NVIDIA A100 GPUs (80 GB memory each), using a batch size of 2. As the speech encoder, we adopted the finetuned `WavLM-Large`[1] model [20], which processes 16 kHz sampled audio into frame-level feature embeddings at a frame rate of 50 Hz with dimensionality 1,024. These features were subsequently downsampled by a factor of $n = 5$ and passed through two linear projection layers, yielding speech embeddings with a final frame rate of 10 Hz and dimensionality 4,096.

For the language modeling component, we employed `Vicuna-7B`[2] [155], a conversationally finetuned variant of `LLaMA` [131] trained on ShareGPT data. During training, parameters of both the speech encoder and the LLM were kept frozen, while only the projection layers were optimized. Optimization was performed with the AdamW algorithm [156], using a learning rate of 0.0001, $\beta = (0.9, 0.999)$, $\varepsilon =$ 1e-08, and a weight decay of 1e-6. A linear warm-up schedule with 1,000 warm-up steps was applied, followed by training for up to 100,000 steps, with early stopping based on the validation loss. LoRA was applied to the LLM with configuration $\alpha = 32$, rank $r = 8$, and dropout $= 0.05$.

We also defined task-specific prompt templates for different model settings. For the **LLM Base-**

[1]https://huggingface.co/microsoft/wavlm-large
[2]https://huggingface.co/lmsys/vicuna-7b-v1.5

line, the prompt consisted of the simple instruction: *"Transcribe speech to text."*. In contrast, the **CMT-LLM** employed an enriched prompt format incorporating contextual biasing information: *"Use the rare words provided to improve the accuracy of ASR if they are relevant. The rare words are [...]."*, where the bracketed segment was dynamically filled with the biasing list. During inference, beam search decoding was performed with a beam size of 4.

Table 5.1: WER performance comparison with different multi-talker ASR models on LibriMix (%) with 1,000 distractors.

Model	**Year**	**LibriMix**	
		Dev	**Test**
Conditional-Conformer-CTC [157]	2021	24.5	24.9
WavLM-CTC [20]	2022	23.0	20.3
Whisper (Small)[3]	2023	26.0	25.0
Conformer[3]	2022	24.7	23.3
+ WavLM-Large upstream[3]	2023	19.4	17.1
GEncSep [7]	2024	17.2	15.0
LLM Baseline	2025	12.7	9.2
CMT-LLM (ours)	**2025**	**8.1**	**7.3**

5.3.2 Experimental Conditions

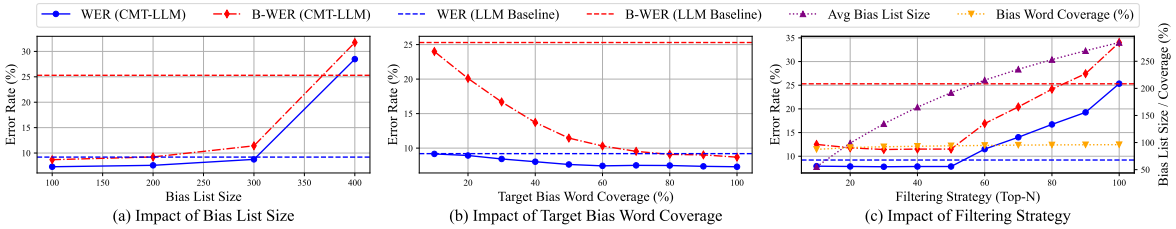
Datasets: We conducted evaluations on two widely used benchmark corpora: **LibriMix** [159] and **AMI** [160].

- **LibriMix:** This dataset is constructed by mixing clean speech from LibriSpeech [121] with environmental noise from WHAM! [161]. Following the ESPNet SOT recipe[3], we generated two-speaker mixtures by overlapping speech signals with a randomly sampled relative offset of 1.0–1.5 seconds between speakers. This procedure results in partially overlapping speech segments and is commonly used to simulate realistic two-speaker overlap conditions. The final dataset comprises approximately 830 hours of speed-perturbed training data, along with 8.2 hours for validation and 7.6 hours for testing.

[3]https://github.com/espnet/espnet/tree/master/egs2/librimix/sot_asr1

Table 5.2: WER performance comparison with different multi-talker ASR models on AMI (%) with 1,000 distractors.

Model	Year	AMI					
		IHM-Mix		SDM		MDM	
		Dev	Test	Dev	Test	Dev	Test
WavLM-CTC [20]	2022	34.4	34.3	39.8	44.0	38.1	41.5
SURT 2.0 (Large) [158]	2023	-	36.8	-	62.5	-	44.4
+ Adaptation [158]	2023	-	35.1	-	44.6	-	41.4
LLM Baseline	2025	24.1	23.5	30.9	34.2	32.9	31.2
CMT-LLM (ours)	**2025**	**21.9**	**22.8**	**30.0**	**32.9**	**29.7**	**30.4**



(a) Impact of Bias List Size (b) Impact of Target Bias Word Coverage (c) Impact of Filtering Strategy

Figure 5.3: Impact of Biasing List Size, Coverage, and Filtering on ASR Performance.

- **AMI:** The AMI corpus consists of approximately 100 hours of real meeting recordings involving 4–5 speakers. In accordance with the icefall SURT recipe[4], we utilized three microphone configurations: IHM-Mix (mixed headset microphones), SDM (single distant microphone), and MDM (beamformed microphone array) [162]. The dataset is divided into 79.4 hours of training data, 9.7 hours for validation, and 9.1 hours for testing.

Biasing List Construction: As the LibriMix dataset does not provide predefined biasing lists, we adopt the established simulation strategy introduced in [1]. Given that LibriMix is derived from LibriSpeech, we employ the same comprehensive biasing list[5], which contains approximately 209.2K entries. All words in this list are treated as rare. For each utterance, the corresponding biasing list is constructed by extracting words from the reference transcript that also appear in the full list, while additional distractors are incorporated according to the specific experimental configuration.

[4]https://github.com/k2-fsa/icefall/tree/master/egs/ami/SURT
[5]https://github.com/facebookresearch/fbai-speech/tree/master/is21_deep_bias

Table 5.3: WER performance comparison with different contextual ASR models on LibriMix and AMI test sets (%).

Model	Type	Distractors	LibriMix	AMI		
				IHM-Mix	SDM	MDM
			WER / B-WER	WER / B-WER	WER / B-WER	WER / B-WER
LLM Baseline	No Biasing	-	9.2 / 25.3	23.5 / 45.6	34.2 / 51.0	31.2 / 49.7
+ ED-CEC [111]	Biasing List	+ 100	8.4 / 15.3	23.1 / 34.2	33.6 / 35.5	30.8 / 32.1
		+ 1,000	8.7 / 15.5	23.1 / 34.8	33.7 / 36.9	30.8 / 35.8
		+ 2,000	8.8 / 16.0	23.2 / 35.1	33.7 / 37.7	30.9 / 38.7
		+ 5,000	8.9 / 16.4	23.2 / 35.5	33.8 / 39.0	31.0 / 40.3
CMT-LLM	No Biasing	-	9.3 / 28.7	23.4 / 48.7	33.6 / 50.8	31.1 / 49.5
	Anti-Context	+ 100	9.4 / 29.7	23.3 / 43.8	33.6 / 50.2	30.9 / 48.2
	Biasing List	+ 100	7.3 / 8.7	22.7 / 29.8	32.7 / 30.6	30.3 / 30.2
		+ 1,000	7.9 / 12.5	22.8 / 33.6	32.9 / 35.0	30.4 / 34.6
		+ 2,000	8.4 / 14.1	22.8 / 35.1	33.0 / 38.9	30.5 / 36.7
		+ 5,000	8.4 / 15.2	22.9 / 36.4	33.1 / 42.5	30.6 / 38.1
	GT Rare Words	-	6.6 / 2.6	22.4 / 24.6	31.7 / 28.4	29.5 / 28.6

To approximate realistic application scenarios, the AMI biasing lists are generated by extracting textual content from lecture slides via Tesseract OCR[6], as illustrated in Fig. 5.1(b). After removing duplicates, words that either appear in the full biasing list or occur fewer than 100 times in the AMI corpus are classified as rare. These lecture-specific lists are subsequently used to support contextual biasing, following the methodology of [69]. Furthermore, for large-scale experiments, all lecture-derived lists are merged to create a unified AMI biasing list, from which distractors are also sampled. It is worth noting that the total number of words included in the AMI biasing lists represents only about 1.0% of the overall vocabulary. Despite their relatively small proportion, these words consist mainly of essential content terms whose accurate recognition is critical for semantic understanding, as exemplified in Fig. 5.1.

Evaluation Metrics: System performance is assessed using both WER and B-WER [1], where B-WER specifically measures the recognition accuracy of words present in the biasing list.

[6]https://github.com/tesseract-ocr/tesseract

5.3.3 Results and Analysis

Baseline Comparisons.

The overall performance of different multi-talker ASR models on LibriMix and AMI is summarized in Table 5.1 and Table 5.2. Across both datasets, the proposed CMT-LLM achieves the lowest WER, consistently outperforming the LLM Baseline as well as state-of-the-art conventional systems. These results clearly demonstrate the effectiveness of the proposed framework in addressing contextual multi-talker ASR.

Table 5.3 further compares several contextual ASR approaches. The LLM Baseline, which does not incorporate a biasing list, exhibits high B-WER, confirming the inherent difficulty of recognizing rare words in the absence of explicit contextual support. ED-CEC [111], a state-of-the-art post-processing approach, achieves improved B-WER when the biasing list contains only a small number of distractors (e.g., +100). However, its performance deteriorates as the size of the biasing list grows, due to increased confusion introduced by distractors.

In contrast, CMT-LLM directly integrates the biasing list into the prompt, leading to substantial gains under small biasing list conditions, surpassing both the LLM Baseline and ED-CEC by a significant margin. In the Anti-Context condition—where target biasing words are replaced with distractors—the model exhibits a sharp increase in B-WER, underscoring the importance of accurate biasing. The GT Rare Words setting, in which only the ground-truth rare words are provided, serves as an upper bound, yielding a B-WER as low as 2.6% on LibriMix and confirming the potential of well-optimized biasing strategies.

Nonetheless, performance degradation is observed when the biasing list expands beyond 1,000 entries. This decline is not only attributed to the increased number of distractors but also to the reduced coverage of target words after filtering. Specifically, coverage falls to 87.40%, 85.07%, and 83.07% when 1,000, 2,000, and 5,000 distractors are added, respectively. Although the proposed filtering mechanism mitigates the negative effect of large lists, the reduction in target coverage inevitably impacts recognition accuracy. Taken together, the results indicate that, for comparable levels of target word coverage, incorporating contextual information directly into the prompt is more effective than relying on post-hoc correction strategies.

Impact of Biasing List Size and Word Coverage.

We further investigate the relationship between biasing list size, word coverage, and recognition performance, as illustrated in Fig. 5.3.

When no filtering mechanism is applied (Fig. 5.3(a)), enlarging the biasing list from 100 to 400 words results in a marked increase in both WER and B-WER. Once the list exceeds 300 entries,

the WER even surpasses that of the LLM Baseline, suggesting that excessively large lists introduce substantial noise and hinder recognition accuracy.

In a complementary analysis with list size fixed at 100 (Fig. 5.3(b)), reducing coverage from 100% to 10% causes a steep rise in B-WER. This outcome underscores the importance of sufficient coverage, as insufficient inclusion of target words substantially diminishes the effectiveness of contextual biasing.

To address these issues, we apply a filtering mechanism to an initial list containing 1,000 distractors (Fig. 5.3(c)). The results indicate that maintaining an average filtered list size below 200 achieves high word coverage while effectively suppressing interference. However, as the Top-N parameter exceeds 50, both WER and B-WER increase sharply, and at Top-60, WER once again surpasses the LLM Baseline. This confirms that recognition deteriorates when the list grows beyond a manageable threshold.

Overall, these findings highlight the critical trade-off between list size and coverage. Future research will focus on strategies for maintaining high coverage while further constraining list size, thereby enhancing robustness and reliability of contextual multi-talker ASR systems.

Illustrative Example of Rare Word Recognition.

An illustrative case is presented in Fig. 5.1(c), where rare personal names extracted from lecture slides are incorporated into the biasing list. By leveraging this contextual information, the proposed CMT-LLM successfully improves transcription accuracy, particularly for infrequent entities that are prone to recognition errors. This example highlights the practical effectiveness of CMT-LLM in enhancing rare word recognition within real-world ASR scenarios.

5.4 Summary

This study introduces, for the first time, an LLM-based SOT framework, termed **CMT-LLM**, for contextual multi-talker ASR. By capitalizing on the strong decoding capacity, long-range contextual modeling, and cross-speaker reasoning ability of LLMs, CMT-LLM demonstrates superior effectiveness in managing complex overlapping speech scenarios. Moreover, contextual information is seamlessly incorporated through prompt-based learning, yielding substantial gains in the recognition of rare words. To further address the practical challenge of large-scale biasing lists containing thousands of entries, we design a coarse decoding-driven filtering algorithm that reduces the candidate pool by preserving only the ten most relevant words for each surviving candidate, which are subsequently integrated into the prompt. This refinement ensures that contextual biasing remains effective even

under conditions with up to 5,000 distractors.

Extensive experimental evaluations confirm that the proposed CMT-LLM achieves state-of-the-art recognition accuracy on both the simulated LibriMix corpus and the real-world AMI meeting dataset, consistently outperforming existing conventional baselines.

Chapter 6

M^{4}SER: Multimodal, Multirepresentation, Multitask, and Multistrategy Learning for Speech Emotion Recognition

In the previous chapters, we primarily focused on advancing ASR, with particular emphasis on contextual biasing, phoneme-augmented error correction, and multi-talker scenarios. Athough ASR plays a central role in transcribing speech with high accuracy, many downstream tasks depend not only on the lexical correctness of the transcription but also on the ability to capture paralinguistic cues embedded in speech. Among these downstream applications, SER has attracted significant research attention, as it is crucial for enhancing human–machine interaction in real-world systems such as conversational agents, customer service platforms, and affective computing applications.

In this chapter, we present **M^{4}SER**, a multimodal, multirepresentation, multitask, and multistrategy learning framework for robust speech emotion recognition. Specifically, we leverage both speech and textual modalities, the latter derived from ASR outputs, to jointly model acoustic and linguistic cues for emotion classification. To mitigate the negative impact of ASR errors, our framework introduces two auxiliary tasks—ASR error detection and ASR error correction—which enhance the reliability of text-based representations. Furthermore, M^{4}SER employs a novel multimodal fusion mechanism to learn modality-specific as well as modality-invariant features, ensuring complementary information across modalities is effectively captured.

Building upon this foundation, we incorporate two additional training strategies: (1) an adversarial learning module that increases the diversity of modality-specific representations, thereby strength-

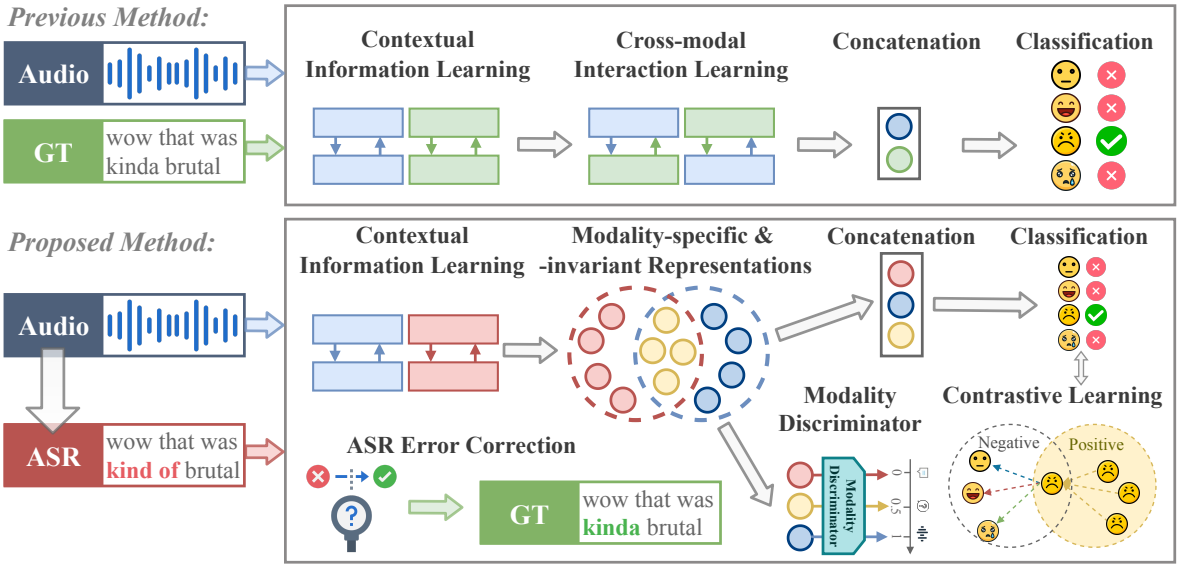


Figure 6.1: Illustration of the difference between previous multimodal SER methods and our proposed method. "ASR" and "GT" denote ASR and ground truth transcripts, respectively.

ening generalization, and (2) a label-based contrastive learning objective that explicitly encourages the extraction of discriminative emotional features. Experiments across two widely used benchmarks, IEMOCAP and MELD, demonstrate that M^{4}SER achieves substantial improvements over state-of-the-art baselines, confirming the effectiveness of our multitask and multistrategy design.

6.1 Introduction

SER has emerged as a fundamental component in advancing human–computer interaction, with applications spanning healthcare, intelligent customer service, and affective computing systems [163, 164]. The ability to automatically infer human emotions from spoken input enables machines to provide more empathetic, adaptive, and context-aware responses. With the rapid development of deep learning, SER has evolved from traditional handcrafted feature pipelines to E2E systems capable of directly mapping raw acoustic signals into categorical emotion labels.

Recent advances in SSL have led to the emergence of powerful speech encoders such as wav2vec [17], HuBERT [19], and WavLM [20], which have achieved state-of-the-art performance in a variety of speech processing tasks, including SER. Despite these successes, speech-only approaches often struggle to capture the full complexity of emotional expression, motivating the integration of multimodal information. In particular, speech and text modalities are frequently combined to provide complementary cues, where text is typically obtained via human transcription or ASR.

Although text-based pretrained models such as BERT [114] and RoBERTa [165] have significantly advanced multimodal SER, the accuracy of ASR remains a critical bottleneck. Recognition errors, especially on emotionally salient words, can propagate into downstream SER systems, degrading their performance. Prior studies have attempted to mitigate this problem by incorporating auxiliary mechanisms such as confidence estimation [166] or ASR error detection [8], yet these approaches either rely too heavily on ASR quality or focus only on identifying erroneous words without directly correcting them, limiting their effectiveness in preserving semantic consistency.

Another important challenge lies in effectively fusing heterogeneous modalities. Conventional strategies such as feature concatenation, recurrent networks, convolutional layers, or cross-modal attention [163] often fall short in aligning modality-specific signals and fail to capture fine-grained emotional dynamics across time. To address these limitations, recent research has emphasized disentangled representation learning, where modality-specific representations (MSRs) preserve unique characteristics of each modality, while modality-invariant representations (MIRs) capture their shared emotional content [167]. However, most existing methods operate at the utterance level, which oversimplifies the temporal structure of emotion and neglects subtle variations critical for robust SER.

To address these challenges, we propose **M^{4}SER**, a framework based on multimodal, multirepresentation, multitask, and multistrategy learning for speech emotion recognition. Building on our previous work [9], which introduced auxiliary tasks of AED and AEC together with a novel multimodal fusion mechanism, we extend the model with two additional strategies. First, we incorporate adversarial learning to encourage richer and more robust modality-specific representations, thereby mitigating noise introduced by ASR errors. Second, we design a label-based contrastive learning objective to enhance the discriminability of emotional features by pulling together instances of the same emotion category and pushing apart those from different categories. These innovations collectively strengthen the model's ability to integrate multimodal signals, preserve semantic coherence, and capture fine-grained emotional dynamics.

Our experimental results on two widely used benchmarks, IEMOCAP and MELD, demonstrate that M^{4}SER achieves substantial improvements over state-of-the-art baselines. The proposed framework not only reduces the negative impact of ASR errors but also leverages complementary cues across modalities to produce more accurate and robust emotion predictions.

The remainder of this chapter is structured as follows. Section 6.2 introduces the details of the proposed M^{4}SER framework. Section 6.3 outlines the experimental setup, including datasets, evaluation metrics, and implementation details. Section 6.4 presents and analyzes the experimental results. Finally, Section 6.5 concludes this chapter and discusses potential directions for future research.

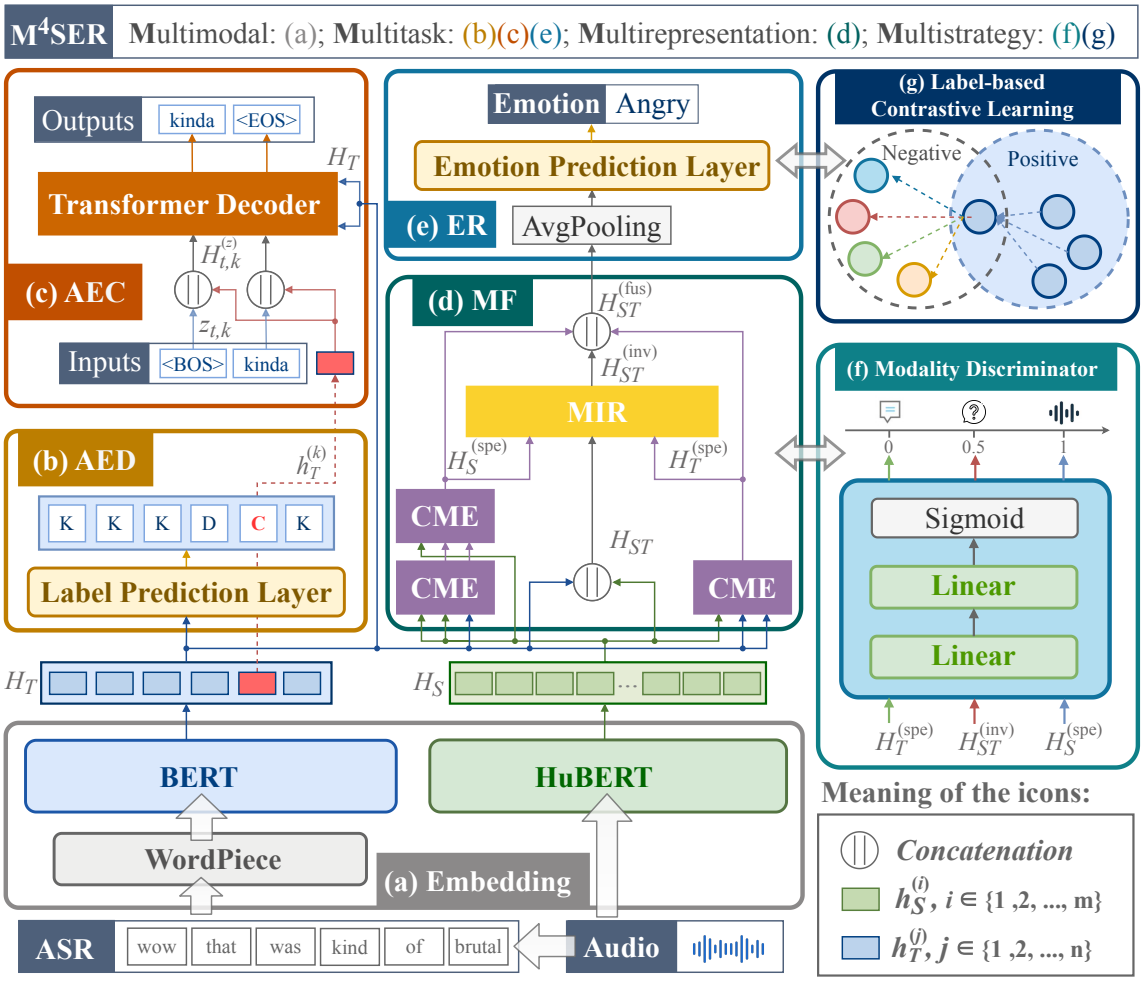


Figure 6.2: Overall architecture of the proposed M^{4}SER model. Specific illustrations of CME and MIR blocks in the (d) MF module are shown in Figs. 6.3 and 6.4, respectively.

6.2 Methodology

6.2.1 Problem Formulation

The proposed M^{4}SER framework can be formalized as a mapping function $f(S,T) = (L,Z)$, where the input consists of two modalities: speech and text. The speech modality is defined as $S = (s_1, s_2, \cdots, s_m) \in \mathbb{R}^m$, representing an utterance encoded into m acoustic frames. The text modality is given by $T = (t_1, t_2, \cdots, t_n) \in \mathbb{R}^n$, which corresponds to n tokens obtained from the ASR hypothesis of the utterance. All tokens are represented using a predefined WordPiece vocabulary [112].

Within this multitask learning framework, the primary task is emotion recognition (ER), formulated as a classification problem with output $L \in \{l_1, l_2, \cdots, l_e\}$, where e denotes the number of emotion categories. In parallel, two auxiliary tasks are introduced: AED and AEC. These tasks produce outputs $Z \in \{Z_1, Z_2, \cdots, Z_k\}$, which capture the GT sequences at positions where ASR hypotheses diverge from human-annotated transcripts. Each element $Z_k = (z_{1,k}, z_{2,k}, \cdots, z_{c,k}) \in \mathbb{R}^c$ represents the k-th error span consisting of c tokens.

As illustrated in Fig. 6.2, the overall architecture of M^{4}SER is composed of five core modules: the embedding module, the AED module, the AEC module, the multimodal fusion (MF) module, and the ER module. In the subsequent section, we provide a detailed description of each of these components.

6.2.2 Embedding Module

The embedding module is designed to generate modality-specific contextual representations for both speech and text inputs, as illustrated in Fig. 6.2(a). It consists of two major components: the acoustic embedding pathway and the token embedding pathway.

Contextual Speech Representations. For the acoustic modality, we adopt HuBERT [19], a pretrained SSL model, as the speech encoder. HuBERT combines convolutional layers with a Transformer-based encoder, enabling it to effectively capture both low-level acoustic features and higher-level contextual dependencies. Given a speech input S, HuBERT produces a sequence of hidden representations: $H_S = (h_S^{(1)}, h_S^{(2)}, \cdots, h_S^{(m)}) \in \mathbb{R}^{m \times d}$, where m denotes the number of acoustic frames and d represents the dimensionality of the hidden representations.

Contextual Token Representations. For the text modality, we utilize the pretrained language model BERT [114] to derive contextualized token embeddings. Given the tokenized ASR hypothesis T, the encoder outputs: $H_T = (h_T^{(1)}, h_T^{(2)}, \cdots, h_T^{(n)}) \in \mathbb{R}^{n \times d}$, where n is the token sequence length. More

formally, the representation can be expressed as:

$$H_T = \text{BERT}(\text{TE}(T) + \text{PE}(T)) \in \mathbb{R}^{n \times d}, \tag{6.1}$$

with $\text{TE}(\cdot)$ denoting token embeddings and $\text{PE}(\cdot)$ representing position embeddings.

6.2.3 ASR Error Detection (AED) Module

The first auxiliary task incorporated into our framework is AED, whose objective is to identify the positions of recognition errors within the ASR hypothesis. Following the alignment strategy described in [113], we align the ASR hypothesis T with the GT transcript C by computing their longest common subsequence. Tokens that belong to the aligned subsequence are assigned the label *KEEP* (**K**), while the remaining tokens are marked either as *DELETE* (**D**) or *CHANGE* (**C**), depending on the required correction operation. An illustrative example of this labeling process is provided in Fig. 6.2(b).

To perform label prediction, we apply a fully connected (FC) classification layer with three output classes corresponding to the operations (**K**, **D**, **C**). Formally, the probability distribution over possible labels for the o-th token is given by:

$$P(y_o \mid h_T^{(o)}) = \text{SoftMax}(\text{FC}(h_T^{(o)})) \in \mathbb{R}^3, \tag{6.2}$$

where $h_T^{(o)}$ denotes the hidden representation of the o-th token, and $P(y_o \mid h_T^{(o)})$ represents the predicted probability distribution over the three editing operations.

The AED module is trained by minimizing the negative log-likelihood loss, defined as:

$$\mathscr{L}_{\text{AED}} = -\sum_{o=1}^{n} \log\left(P(y_o \mid h_T^{(o)})\right). \tag{6.3}$$

6.2.4 ASR Error Correction (AEC) Module

The second auxiliary task is AEC, which is designed to refine the transcription by correcting errors previously identified by the AED module. In contrast to conventional autoregressive decoders that regenerate the entire sequence from scratch, the proposed AEC decoder focuses exclusively on the positions labeled as **C**, thereby improving efficiency and accuracy.

Concretely, for each token marked as erroneous (**C**), the model initiates a dedicated correction sequence. Each sequence begins with a special start symbol $<BOS>$ and generates the corrected span in an autoregressive manner. At every decoding step, the input to the decoder consists of the embeddings of all tokens generated thus far, concatenated with the hidden representation of the cor-

responding erroneous token. All correction sequences are processed simultaneously through a shared transformer-based decoder, which attends to the full ASR hidden representation H_T as memory.

Formally, for the k^{th} erroneous position, the generated correction sequence of length c is denoted as

$$Z_k = z_{1:c,k} = (z_{1,k}, z_{2,k}, \ldots, z_{c,k}),$$

with $z_{1,k}$ initialized by the start token $<BOS>$. The decoder input embeddings for the first t decoding steps are constructed as:

$$H^{(z)}_{1:t,k} = \text{FC}\Big((\text{TE}(z_{1:t,k}) + \text{PE}(z_{1:t,k})) \parallel h^{(k)}_T \Big) \in \mathbb{R}^{t \times d}, \tag{6.4}$$

where "$\parallel$" denotes concatenation along the feature dimension, $\text{TE}(\cdot)$ and $\text{PE}(\cdot)$ are token and positional embeddings, and $h^{(k)}_T$ represents the hidden state of the k^{th} erroneous token, repeated across all time steps $1{:}t$.

A transformer decoder [115] is then applied, with queries given by $H^{(z)}_{1:t,k}$ and both keys and values derived from H_T:

$$O^{(\text{gen})}_{t+1,k} = \text{Transformer}_{\text{Decoder}}(H^{(z)}_{1:t,k}, H_T, H_T) \in \mathbb{R}^d, \tag{6.5}$$

yielding the decoder hidden representation at step $t+1$. The probability distribution over the vocabulary is computed as:

$$P^{(\text{gen})}_{t+1,k} = \text{SoftMax}\big(\text{FC}(O^{(\text{gen})}_{t+1,k})\big) \in \mathbb{R}^{d_{\text{vocab}}}, \tag{6.6}$$

where d_{vocab} is the vocabulary size of the text encoder (e.g., BERT). The next token is predicted as $z_{t+1,k} = \arg\max P^{(\text{gen})}_{t+1,k}$.

Finally, the AEC module is optimized using the negative log-likelihood objective:

$$\mathscr{L}_{\text{AEC}} = -\sum_k \sum_t \log P^{(\text{gen})}_{t,k}. \tag{6.7}$$

6.2.5 Multimodal Fusion (MF) Module

Following the design principles in [9], the MF module is constructed from three cross-modal encoder (CME) blocks and a MIR generator. The purpose of this module is to jointly learn MSRs and MIRs, thereby facilitating more effective integration of speech and text modalities. In the following, we describe the internal mechanisms of the CME blocks and the MIR generator in detail.

Cross-Modal Encoder (CME). Each CME block follows the structure of a transformer layer, consisting of a multi-head cross-attention mechanism [168], residual connections, and feed-forward

layers, as illustrated in Fig. 6.3. To obtain token-aware speech representations, the textual embeddings H_T are used as the query Q, while the speech embeddings H_S serve as the key K and value V inputs:

$$\hat{H}_S^{(spe)} = \mathrm{FC}\big(\text{Cross-Attn}(H_T, H_S, H_S)\big) \in \mathbb{R}^{n\times d}, \tag{6.8}$$

where Cross-Attn$(\cdot)$ denotes the cross-attention operation.

Since the representations $\hat{H}_S^{(spe)}$ obtained at this stage are aligned with token-level embeddings rather than acoustic frames, a second CME block is introduced to re-anchor them to the speech modality. In this block, the original H_S is used as the query, while $\hat{H}_S^{(spe)}$ is treated as both key and value. The resulting output is the token-aware speech representation:

$$H_S^{(\mathrm{spe})} = \mathrm{FC}\big(\text{Cross-Attn}(H_S, \hat{H}_S^{(spe)}, \hat{H}_S^{(spe)})\big) \in \mathbb{R}^{m\times d}. \tag{6.9}$$

In a symmetrical manner, speech-aware token representations are generated by applying a CME block in which the speech embeddings H_S are fed as the query, and the token embeddings H_T serve as key and value:

$$H_T^{(\mathrm{spe})} = \mathrm{FC}\big(\text{Cross-Attn}(H_S, H_T, H_T)\big) \in \mathbb{R}^{m\times d}. \tag{6.10}$$

Through this iterative and reciprocal attention mechanism, the CME blocks effectively capture fine-grained cross-modal interactions, enabling the downstream MIR generator to construct more robust shared representations.

MIR Generator. The MIR generator is designed to distill shared information across modalities from their respective modality-specific representations. As shown in Fig. 6.4(a), the generator employs a hybrid-modal attention (HMA) mechanism, which integrates both acoustic and textual cues into a unified space. Formally, for each modality $i \in \{S, T\}$, the process is defined as

$$H_i^{(b)} = \mathrm{HMA}\big(H_i^{(\mathrm{spe})}, \mathrm{FC}(H_{ST})\big) \in \mathbb{R}^{m\times d}, \tag{6.11}$$

where $H_i^{(\mathrm{spe})}$ denotes the modality-specific representation of modality i, and H_{ST} corresponds to the concatenation of H_S and H_T, projected via a FC layer to ensure dimensional consistency. This step ensures that both speech and text information are embedded within a comparable representational space.

Subsequently, the outputs of the HMA modules are aggregated with H_{ST} and refined through convolutional and normalization layers to yield the final modality-invariant representation:

$$H_{ST}^{(\mathrm{inv})} = \mathrm{Norm}\Big(H_{ST} + \sum_{i\in\{S,T\}} \mathrm{Conv}_{1\mathrm{d}}\big(H_i^{(b)}\big)\Big) \in \mathbb{R}^{m\times d}, \tag{6.12}$$

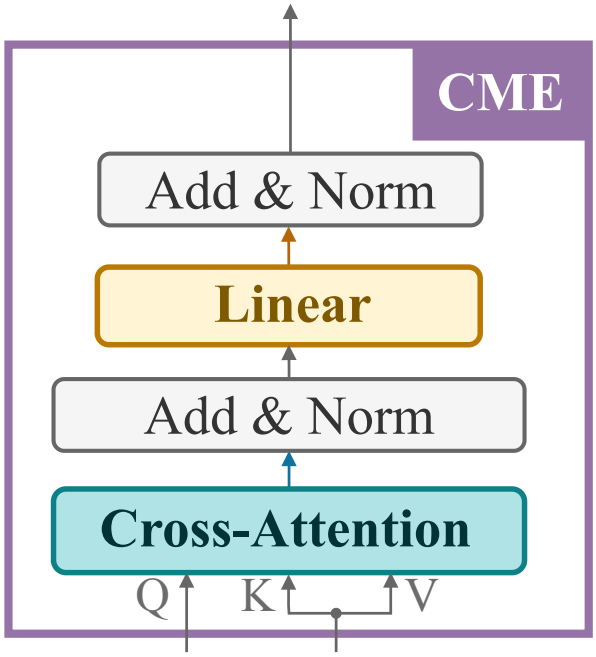


Figure 6.3: Illustration of the CME block.

where $\mathrm{Norm}(\cdot)$ denotes layer normalization [169], and $\mathrm{Conv}_{1\mathrm{d}}(\cdot)$ refers to a 1×1 convolution followed by PReLU activation [170]. For clarity, Eqs. (6.11) and (6.12) can be summarized as

$$H_{ST}^{(\mathrm{inv})} = G\big(H_S^{(\mathrm{spe})}, H_T^{(\mathrm{spe})}, H_{ST}\big) \in \mathbb{R}^{m \times d}, \tag{6.13}$$

where $G(\cdot)$ denotes the overall MIR generation process.

Hybrid-Modal Attention (HMA). As illustrated in Fig. 6.4(b), HMA begins with a cross-attention layer to identify the information shared between the concatenated representation H_{ST} and each modality-specific representation:

$$H_i^{(\mathrm{share})} = \text{Cross-Attn}(H_{ST}, H_i^{(\mathrm{spe})}, H_i^{(\mathrm{spe})}) \in \mathbb{R}^{m \times d}, \quad i \in \{S, T\}. \tag{6.14}$$

To further promote invariance, a parallel convolutional network is employed to generate a gating mask that selectively suppresses modality-dependent information:

$$H_i^{(b)} = H_i^{(\mathrm{share})} \otimes \sigma\Big(\mathrm{Conv}_{1\mathrm{d}}\big(H_i^{(\mathrm{spe})} \parallel H_{ST}\big)\Big) \in \mathbb{R}^{m \times d}, \quad i \in \{S, T\}, \tag{6.15}$$

where $\parallel$ denotes feature-wise concatenation, $\sigma(\cdot)$ denotes the Sigmoid activation function and $\otimes$ indicates element-wise multiplication.

Finally, the MIR is concatenated with the modality-specific representations to form the overall multimodal fusion output:

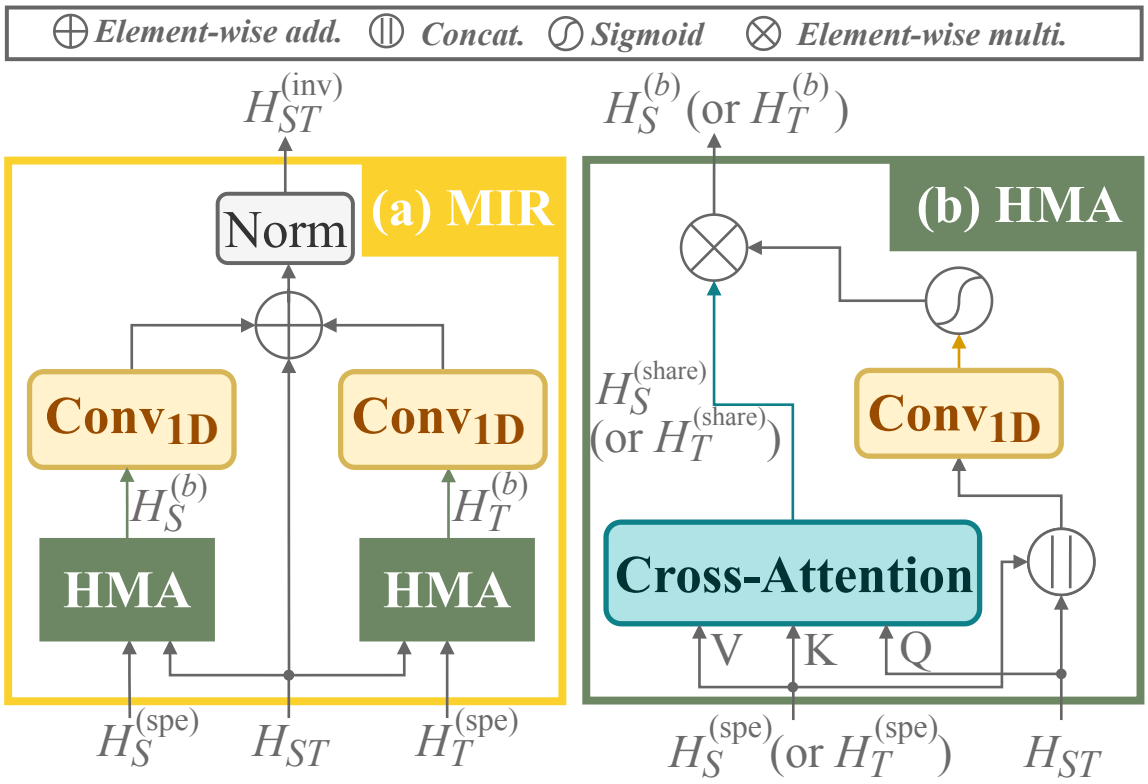


Figure 6.4: Illustration of the MIR generator and HMA blocks.

$$H_{ST}^{(\text{fus})} = H_S^{(\text{spe})} \parallel H_T^{(\text{spe})} \parallel H_{ST}^{(\text{inv})} \in \mathbb{R}^{3m \times d}. \tag{6.16}$$

6.2.6 Emotion Recognition (ER) Module

The emotion recognition task is carried out by applying a temporal average pooling operation to the fused multimodal feature representation $H_{ST}^{(\text{fus})}$ obtained from the MF module. The pooled feature vector is then passed through a FC layer followed by a SoftMax activation function to produce the probability distribution over emotion categories:

$$H_{ST}^{(\text{ap})} = \text{AvgPooling}(H_{ST}^{(\text{fus})}) \in \mathbb{R}^d, \tag{6.17}$$

$$P(y_{\text{emo}} | H_{ST}^{(\text{fus})}) = \text{SoftMax}(\text{FC}(H_{ST}^{(\text{ap})})) \in \mathbb{R}^e, \tag{6.18}$$

where $P(y_{\text{emo}} \mid H_{ST}^{(\text{fus})})$ denotes the predicted probability distribution over emotion classes, and e is the total number of emotion classes. The training objective for this module is defined as the negative

log-likelihood loss:

$$\mathscr{L}_{\text{ER}} = -\sum \log\left(P(y_{\text{emo}}|H_{ST}^{(\text{fus})})\right). \tag{6.19}$$

6.2.7 Modality Discriminator

To further enhance the quality of modality-invariant representations produced by the MIR generator, we introduce a modality discriminator D, illustrated in Fig. 6.2(f). The discriminator consists of two linear layers separated by a ReLU activation, followed by a Sigmoid activation at the output. Its role is to encourage the MIR generator to produce features that are indistinguishable with respect to their modality, thereby strengthening modality invariance.

Specifically, the discriminator D outputs a scalar between 0 and 1 for each temporal frame, where values close to 0 correspond to the text modality, values close to 1 correspond to the speech modality, and values around 0.5 indicate an ambiguous modality. Formally, the discriminator operates on representations $H \in \{H_T^{(\text{spe})}, H_S^{(\text{spe})}, H_{ST}^{(\text{inv})}\}$ as $D(H) \in \mathbb{R}^{m\times 1}$.

During training, the discriminator is optimized to correctly classify frames in the modality-specific representations $H_T^{(\text{spe})}$ and $H_S^{(\text{spe})}$ as text (0) and speech (1), respectively. Conversely, the modality-invariant representation $H_{ST}^{(\text{inv})}$, produced by the MIR generator, is encouraged to yield outputs close to 0.5, reflecting an indistinguishable modality. This adversarial interplay between the discriminator and the generator is captured by the following loss function:

$$\begin{aligned}\mathscr{L}_{\text{GAN}} &= \mathscr{L}_D + \mathscr{L}_G \\ &= \mathbb{E}\Big[\log D(H_S^{(\text{spe})}) + \log\left(1 - D(H_T^{(\text{spe})})\right)\Big] \\ &\quad + \mathbb{E}\Big[-\log D(H_{ST}^{(\text{inv})}) - \log\left(1 - D(H_{ST}^{(\text{inv})})\right)\Big],\end{aligned} \tag{6.20}$$

where $H_{ST}^{(\text{inv})} = G(H_S^{(\text{spe})}, H_T^{(\text{spe})}, H_{ST})$ is the output of the MIR generator as defined in Eq. (6.13), and $\mathbb{E}$ denotes the expectation over all temporal frames in a batch.

6.2.8 Label-based Contrastive Learning (LCL)

To enhance the model's capability to learn emotion features from multimodal data, we employ a label-based contrastive learning task to complement the cross-entropy loss, as shown in Fig. 6.2(g). This task aids the model in extracting emotion-related features when the MF module integrates speech and text data. As depicted in the LCL task in Fig. 6.5, we categorize data in each batch into positive and

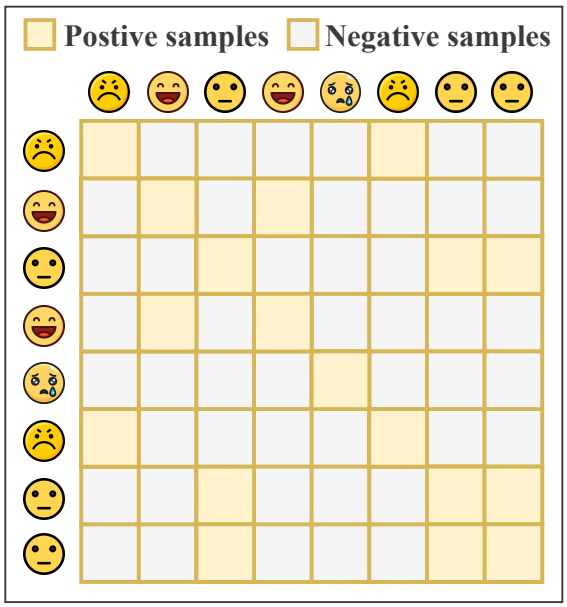


Figure 6.5: Example of constructing positive and negative samples for label-based contrastive learning on the IEMOCAP dataset. In this example, the batch contains eight samples. Yellow squares represent positive samples, whereas white squares represent negative samples.

negative samples on the basis of emotion labels. For instance, in a batch containing eight samples, we compute the set of positive samples for each sample, where those with the same label are considered positive samples (yellow squares), and those with different labels are considered negative samples (white squares). We then calculate the LCL using Eq. 6.21, which promotes instances of a specific label to be more similar to each other than instances of other labels.

$$\begin{aligned} \mathscr{L}_{\text{LCL}} &= \sum_{i \in I} \mathscr{L}_{\text{LCL},i} \\ &= -\frac{1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp(\langle (H_{ST}^{(\text{ap})})_i, (H_{ST}^{(\text{ap})})_p \rangle / \tau)}{\sum_{a \in A(i)} \exp(\langle (H_{ST}^{(\text{ap})})_i, (H_{ST}^{(\text{ap})})_a \rangle / \tau)}, \end{aligned} \tag{6.21}$$

where $\langle \cdot, \cdot \rangle$ denotes cosine similarity and τ is the temperature parameter. For the multimodal representations, let $(H_{ST}^{(\text{ap})})_i$, $(H_{ST}^{(\text{ap})})_p$, and $(H_{ST}^{(\text{ap})})_a$ represent the i^{th} sample, the p^{th} positive sample, and the a^{th} sample, respectively. We define I as the index set of samples, $P(i)$ as the set of positive samples for the i^{th} sample, and $A(i)$ as the set of all samples.

6.2.9 Joint Training

During training, the optimization of the M^{4}SER framework is guided by three task-specific loss functions and two strategy-based objectives. The task-specific losses correspond to ER, AED, and AEC,

denoted as $\mathscr{L}_{\text{ER}}$, $\mathscr{L}_{\text{AED}}$, and $\mathscr{L}_{\text{AEC}}$, respectively. In addition, two auxiliary strategies are incorporated: the adversarial training objective $\mathscr{L}_{\text{GAN}}$ and the label-based contrastive learning loss $\mathscr{L}_{\text{LCL}}$. The complete optimization process of M^{4}SER is summarized in Algorithm 1.

The overall training objective is expressed as the linear combination of these five losses:

$$\mathscr{L} = \mathscr{L}_{\text{ER}} + \alpha \cdot \left(\beta \cdot \mathscr{L}_{\text{AED}} + \mathscr{L}_{\text{AEC}}\right) + \gamma \cdot \mathscr{L}_{\text{GAN}} + \lambda \cdot \mathscr{L}_{\text{LCL}}, \tag{6.22}$$

where α controls the relative importance of the primary task (ER) against the auxiliary tasks (AED and AEC), β balances the two auxiliary tasks, and γ and λ regulate the contributions of the adversarial and contrastive learning strategies, respectively.

Following the adversarial training paradigm proposed by Goodfellow et al. [171], the optimization of the GAN component is divided into two alternating steps. In the first step, the discriminator is updated by maximizing $\mathscr{L}_{\text{GAN}}$, while the parameters of the generator remain fixed. As indicated in Eq. (6.20), maximizing the first term L_D of $\mathscr{L}_{\text{GAN}}$ drives the discriminator to correctly classify modality-specific features, pushing $H_S^{(\text{spe})}$ toward 1 and $H_T^{(\text{spe})}$ toward 0.[1] Conversely, maximizing the second term L_G (equivalently, minimizing $-L_G$) forces $H_{ST}^{(\text{inv})}$ toward 0 or 1,[2] implying that the discriminator perceives $H_{ST}^{(\text{inv})}$ as modality-specific, contrary to the intended modality-invariant property.

In the second step, the discriminator is frozen, and the generator (i.e., the MIR module) is updated by minimizing L_G. This encourages the discriminator's output for $H_{ST}^{(\text{inv})}$ to converge toward 0.5, effectively rendering these representations modality-agnostic by blurring the distinction between text and speech. In parallel, $\mathscr{L}_{\text{ER}}$ is employed to optimize the primary emotion recognition task, while $\mathscr{L}_{\text{AED}}$ and $\mathscr{L}_{\text{AEC}}$ guide the auxiliary tasks of ASR error detection and correction. Furthermore, $\mathscr{L}_{\text{LCL}}$ enforces the label-based contrastive learning strategy. The entire framework is jointly optimized in an E2E manner, with task weights controlled by predefined hyperparameters.

During inference, the AED and AEC modules are omitted. The system thus operates by directly taking speech and ASR transcripts as input, producing emotion classification outputs through the remaining components.

6.3 Experimental Setup

This section describes the datasets employed in our study, the evaluation metrics used for performance assessment, and the corresponding implementation details.

[1] The function $\log(x) + \log(1-y)$, where $x, y \in (0,1)$, reaches its maximum when $x \to 1$ and $y \to 0$.

[2] The function $\log(x) + \log(1-x)$, with $x \in (0,1)$, achieves its maximum at $x = 0.5$ and approaches its minimum as $x \to 0$ or $x \to 1$.

Algorithm 1 M^{4}SER Optimization

Require: Training data $\mathscr{D}$ that contains multimodal inputs, namely, speech and corresponding ASR text pairs (S,T), and outputs, namely, emotion labels, ASR operation labels, and the ground truth transcription (y_e, y_d, y_c). The M^{4}SER network θ that consists of Embedding θ_{Semb} and θ_{Temb}, Encoders θ_{STenc}, MIR generator θ_G, modality discriminator θ_D and downstream emotion recognition module θ_{ER}, ASR error detection module θ_{AED}, and ASR error correction module θ_{AEC}. Hyperparameter weights α, β, γ, λ.

Randomly initialize the entire system θ.
if select self-supervised setting **then**
  Load the pretrained HuBERT model for θ_{Semb} and BERT model for θ_{Temb}.
end if
while not converged **do**
  for $(S,T) \in \mathscr{D}$ **do**
    Forward propagation:
    $H_S = \theta_{Semb}(S), H_T = \theta_{Temb}(T)$ ▷ **Embedding**
    $H_S^{(\text{spe})}, H_T^{(\text{spe})} = \theta_{STenc}(H_S, H_T)$ ▷ **CME**
    $H_{ST} = H_S \parallel H_T$
    $H_{ST}^{(\text{inv})} = \theta_G(H_S^{(\text{spe})}, H_T^{(\text{spe})}, H_{ST})$ ▷ **Generator**
    $\hat{y}_e = \theta_{ER}(H_S^{(\text{spe})} \parallel H_T^{(\text{spe})} \parallel H_{ST}^{(\text{inv})})$ ▷ **SER**
    $\hat{y}_d = \theta_{AED}(H_T)$ ▷ **ASR Error Detection**
    $\hat{y}_c = \theta_{AEC}(H_T)$ ▷ **ASR Error Correction**
    Training Objectives:
    $\mathscr{L}_{\text{GAN}}(\mathscr{L}_D \text{ and } \mathscr{L}_G)$ in Eq. 6.20 ▷ **Discriminator**
    $\mathscr{L}_{\text{LCL}}$ in Eq. 6.21 ▷ **Contrastive learning**
    $\mathscr{L}_{\text{ER}} = \text{CrossEntropy}(\hat{y_e}, y_e)$
    $\mathscr{L}_{\text{AED}} = \text{CrossEntropy}(\hat{y_d}, y_d)$
    $\mathscr{L}_{\text{AEC}} = \text{CrossEntropy}(\hat{y_c}, y_c)$
    Backpropagation: ▷ **Adversarial training**
    Update Discriminator: ▷ **unfreeze θ_D**
    $\theta_D \leftarrow \arg\max \mathscr{L}_{\text{GAN}}$
    Update the Rest Network: ▷ **freeze θ_D**
    $\theta \backslash \theta_D \leftarrow \arg\min \mathscr{L}_{\text{ER}} + \alpha \cdot (\beta \cdot \mathscr{L}_{\text{AED}} + \mathscr{L}_{\text{AEC}}) + \gamma \cdot \mathscr{L}_G + \lambda \cdot \mathscr{L}_{\text{LCL}}$
  end for
end while

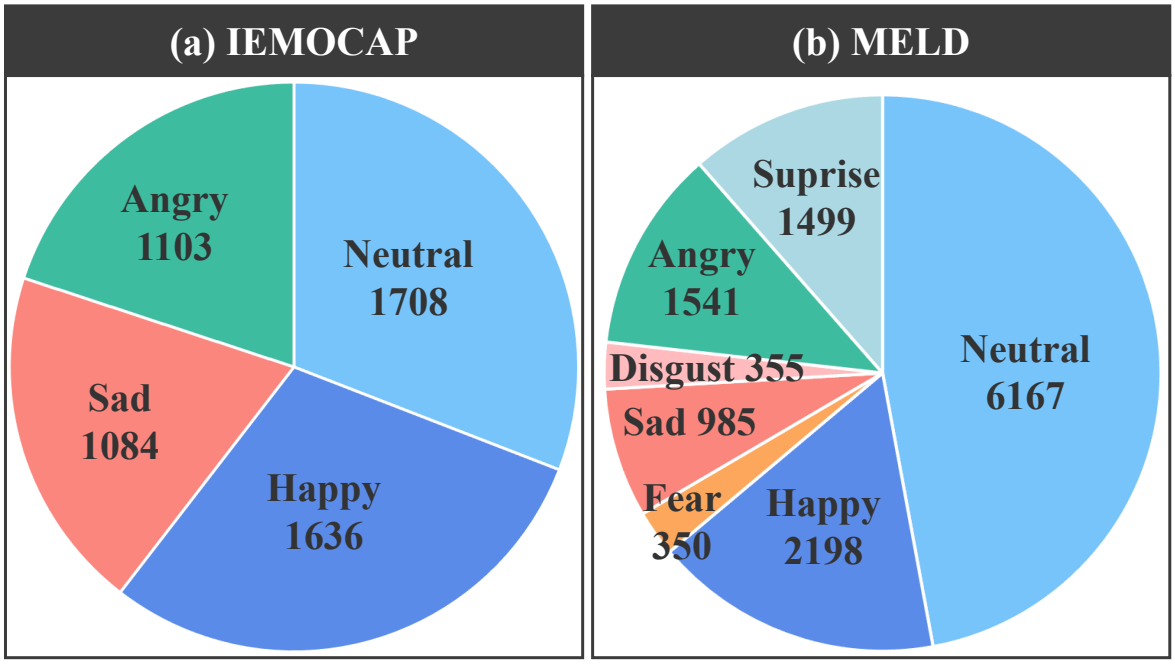


Figure 6.6: Distribution of emotional categories in the IEMOCAP and MELD datasets.

6.3.1 Datasets and Evaluation Metrics

To evaluate the effectiveness of the proposed framework, we conducted experiments on two widely used benchmark datasets: the Interactive Emotional Dyadic Motion Capture (IEMOCAP) corpus [172] and the Multimodal Emotion Lines Dataset (MELD) [173]. The statistics of these datasets are summarized in Fig. 6.6.

IEMOCAP. This dataset contains approximately 12 hours of dyadic speech interactions involving ten speakers across five scripted sessions. Following prior studies [8, 174], we adopted a subset of 5,531 utterances annotated with four emotion categories: "Neutral" (1,708), "Angry" (1,103), "Happy" (including "Excited", 595 + 1,041 = 1,636), and "Sad" (1,084). For evaluation, we applied five-fold leave-one-session-out (LOSO) cross-validation, reporting results in terms of weighted accuracy (WA) and unweighted accuracy (UA).

MELD. This dataset is composed of 13,708 utterances derived from the TV series *Friends*, split into 9,989 for training, 1,109 for validation, and 2,610 for testing. It covers seven emotion classes: "Neutral" (6,167), "Happy" (2,198), "Fear" (350), "Sad" (985), "Disgust" (355), "Angry" (1,541), and "Surprise" (1,499). Hyperparameters were tuned on the validation set, and the final performance was reported on the test set using the best checkpoint, with accuracy (ACC) and weighted F1-score (W-F1) as evaluation metrics.

Table 6.1: Parameter settings for the IEMOCAP and MELD datasets.

Parameter	**IEMOCAP**	**MELD**
Batch size	16	16
Epochs	100	100
Learning rate	$1e^{-5}$	$1e^{-4}$
Dropout	0.5	0.5
d_S	768	768
d_T	768	768
d_{vocab}	30,522	30,522
d	768	768
Attention layers	12	12
Attention heads	12	12
α	0.1	0.1
β	3	3
γ	0.01	0.01
λ	0.1	0.1
τ	0.07	0.07

6.3.2 Implementation Details

The proposed framework was implemented in Python 3.10.0 with PyTorch 1.11.0. All experiments were conducted on a workstation equipped with an Intel(R) Xeon(R) Gold 6248 CPU running at 2.50 GHz, 32 GB RAM, and a single NVIDIA Tesla V100 GPU. The hyperparameter configurations for both datasets are summarized in Table 6.1.

The acoustic encoder was initialized with the `hubert-base-ls960`[3] model, producing acoustic features of dimension $d_S = 768$. For the text modality, we adopted the `bert-base-uncased`[4] model, yielding token embeddings of dimension $d_T = 768$. The vocabulary size was set to $d_{vocab} = 30,522$. Both encoders were fine-tuned during training. Hidden dimensionality d was fixed at 768, with 12 attention layers and 12 heads in both encoders.

For the correction component, the transformer decoder (Fig. 6.2(c)) was implemented as a single-layer transformer with hidden size 768. To obtain ASR hypotheses, we employed Whisper [2], using

[3] `https://huggingface.co/facebook/hubert-base-ls960`
[4] `https://huggingface.co/google-bert/bert-base-uncased`

Table 6.2: Performance Comparison of SOTA Multimodal Models on IEMOCAP (%). "S" and "T" represent Speech and Text modalities, respectively. "ASR" and "GT" denote ASR and ground truth transcripts, respectively. **Bold** indicates the best result, whereas underline signifies the second-best result.

Method	**Year**	**Modality**	**WA**	**UA**
SAWC [174]	2022	S+T(ASR)	76.6	76.8
RMSER-AEA [8]	2023	S+T(ASR)	76.4	76.9
SAMS [175]	2023	S+T(GT)	76.6	78.1
MGAT [95]	2023	S+T(GT)	78.5	79.3
MMER [88]	2023	S+T(GT)	78.1	79.8
IMISA [176]	2024	S+T(GT)	77.4	77.9
MAF-DCT [177]	2024	S+T(GT)	78.5	79.3
FDRL [96]	2024	S+T(GT)	78.3	79.4
MF-AED-AEC [9]	2024	S+T(ASR)	78.1	79.3
M^{4}SER	**2024**	**S+T(ASR)**	**79.2**	**80.1**

the `openai/whisper-medium.en`[5] and `openai/whisper-large-v2`[6] checkpoints. These achieved word error rates (WERs) of 20.48% on IEMOCAP and 37.87% on MELD, respectively.

Optimization was performed using Adam [117] with a batch size of 16. Following empirical results and prior work [183, 184], the learning rate was fixed at $1e^{-5}$ for IEMOCAP and $1e^{-4}$ for MELD. For multitask learning, β was set to 3 and α to 0.1, while for multistrategy learning, γ and λ were set to 0.01 and 0.1, respectively. The temperature τ was fixed at 0.07. To validate these configurations, a one-at-a-time sensitivity analysis of α, β, γ, and λ was conducted, with results reported in Section 6.4.3 and illustrated in Fig. 6.7.

[5] `https://huggingface.co/openai/whisper-medium.en`
[6] `https://huggingface.co/openai/whisper-large-v2`

Table 6.3: Performance Comparison of SOTA Multimodal Models on MELD (%). "V" represents Visual modality. * indicates the result of training with GT texts and testing with ASR texts. We reimplement the method indicated by ° on MELD and obtain the corresponding result.

Method	Year	Modality	ACC	W-F1
Full [178]	2022	S+T(ASR)	-	61.4
HCAM* [179]	2023	S+T(ASR)	-	50.2
DIMMN [180]	2023	S+T(GT)+V	60.6	58.0
MER-HAN [181]	2023	S+T(GT)	62.9	60.2
MLCCT [182]	2023	S+T(GT)	63.2	62.4
SAMS [175]	2023	S+T(GT)	65.4	62.6
MF-AED-AEC° [9]	2024	S+T(ASR)	65.5	64.1
M^{4}SER	**2024**	**S+T(ASR)**	**66.5**	**66.0**

6.4 Experimental Results

6.4.1 Comparison with State-of-the-Art (SOTA) Methods

In this section, we present a comprehensive comparison between the proposed M^{4}SER framework and a range of representative multimodal speech emotion recognition (SER) approaches from the recent literature. The experiments were conducted on the IEMOCAP dataset, and for fairness, the comparison followed established evaluation protocols. The baseline methods include:

- **SAWC** [174], which reduces the impact of ASR errors by assigning confidence-based weights and emphasizing corresponding acoustic segments.
- **RMSER-AEA** [8], which integrates complementary semantic information and introduces an auxiliary task to handle ASR errors, while simultaneously fusing acoustic and textual representations.
- **SAMS** [175], which introduces high-level emotional representations as supervisory signals, enabling multi-spatial learning for each modality and facilitating cross-modal semantic alignment.
- **MGAT** [95], which addresses emotional asynchrony and modality misalignment through a multi-granularity attention mechanism.
- **MMER** [88], which applies early fusion and cross-modal self-attention across acoustic and textual modalities, while incorporating three auxiliary tasks to enhance SER. For fairness, we adopt the version that excludes augmented textual data.

Table 6.4: Ablation study on IEMOCAP and MELD datasets (%). "w/o" means "without". "Concat" denotes "concatenation" operation.

Method	IEMOCAP		MELD	
	WA	**UA**	**ACC**	**W-F1**
M^{4}SER (Full)	**79.2**	**80.1**	**66.5**	**66.0**
A. Impact of Multimodalities				
(1) only Speech Modality	68.9	69.8	51.7	45.1
(2) only Text Modality	65.1	66.4	62.2	58.6
(3) Speech & Text + Concat	73.4	75.2	63.9	60.8
B. Impact of Multirepresentations				
(1) w/o Modality-Specific (MSR)	77.3	78.4	65.0	62.4
(2) w/o Modality-Invariant (MIR)	78.9	79.5	65.6	64.5
(3) w/o MSR & MIR	76.5	77.9	64.8	61.7
C. Impact of Multitasks				
(1) w/o ASR Error Detection	76.9	78.3	64.3	61.7
(2) w/o ASR Error Correction	78.0	79.2	65.9	64.1
(3) w/o AED & AEC	77.3	78.6	65.1	62.6
D. Impact of Multistrategies				
(1) w/o GAN	78.7	79.8	65.8	65.3
(2) w/o LCL	78.2	79.6	65.9	64.8
(3) w/o GAN & LCL	78.1	79.3	65.5	64.1

- **IMISA** [176], which projects speech–text pairs into a shared representation space, extracting modality-specific features and employing contrastive learning for sample-level alignment.
- **MAF-DCT** [177], which leverages a dual framework combining SSL-derived representations with spectral features, and employs a dual cross-modal transformer for effective interaction modeling.
- **FDRL** [96], which introduces modality-shared and modality-private encoders, and incorporates fine-grained alignment and disparity modules to simultaneously improve consistency and diversity across modalities.
- **MF-AED-AEC** [9], our prior work, which introduced two auxiliary tasks—ASR error detection and correction—along with a multimodal fusion method designed to enhance semantic coherence in ASR-derived text and learn shared multimodal representations.

For the MELD dataset, the SAMS and MF-AED-AEC approaches were also employed as baselines. In addition, the following representative methods were considered for comparison:

- **Full** [178], which incorporates contextual cross-modal transformers together with graph convolutional networks to improve emotion representation learning and modality fusion.
- **HCAM** [179], which combines wav2vec-based audio embeddings and BERT-derived text features, applying co-attention mechanisms and recurrent neural networks for multimodal emotion recognition.
- **DIMMN** [180], which employs a dynamic fusion framework that integrates multiview attention layers, temporal convolutional networks, gated recurrent units, and memory networks to model temporal dependencies and contextual interactions.
- **MER-HAN** [181], which introduces a hierarchical attention structure incorporating local intramodal, cross-modal, and global intermodal attention to more effectively capture emotional cues.
- **MLCCT** [182], which combines SSL-based feature extraction with Bi-LSTM layers, cross-modal transformers, and self-attention modules to facilitate multimodal feature interaction and fusion.

Tables 6.2 and 6.3 present a comparison between the proposed M^{4}SER framework and several recent multimodal SER approaches on the IEMOCAP and MELD datasets. Our method achieves a WA of 79.2% and a UA of 80.1% on IEMOCAP, and an ACC of 66.5% and a W-F1 of 66.0% on MELD, surpassing all competing baselines and thereby demonstrating the effectiveness of M^{4}SER. On the IEMOCAP dataset in particular, M^{4}SER yields a 0.7% improvement in WA and a 0.8% improvement in UA compared with the recent MAF-DCT method, establishing a new state-of-the-art performance. These gains can be attributed to M^{4}SER's ability to reduce the detrimental impact of ASR errors, jointly model modality-specific and modality-invariant representations, enhance cross-modal consistency through adversarial learning, and improve discriminative emotion representation learning via the LCL loss.

6.4.2 Ablation Study

Our proposed M^{4}SER model is composed of four types of learning: multimodal learning, multi-representation learning, multitask learning, and multistrategy learning. To determine the impact of different types of learning on the performance of the emotion recognition task, ablation experiments were conducted on the IEMOCAP and MELD datasets, as presented in Table 7.2.

Impact of Multimodalities. To investigate the contribution of multimodal inputs, we carried out single-modal experiments using either speech or text alone. In these experiments, the MF module, LCL loss, and GAN loss were omitted, thereby eliminating intermodal interaction. The results,

presented in Table 7.2(A), indicate that for the IEMOCAP dataset, speech outperforms text, whereas for the MELD dataset, text yields superior results. This discrepancy implies that in IEMOCAP, emotional cues are conveyed more strongly through acoustic signals, while in MELD, textual information plays a more dominant role in expressing emotions. Furthermore, the multimodal baseline, which combines speech and text features via simple concatenation (Table 7.2 A(3)), substantially improves recognition accuracy compared with both single-modal settings, underscoring the indispensable value of multimodal integration for emotion recognition.

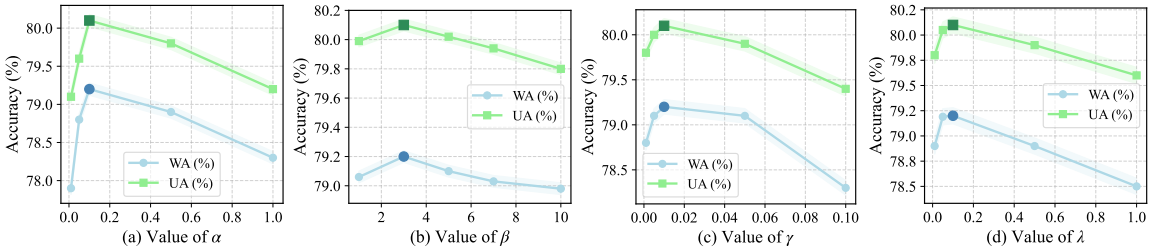


Figure 6.7: Sensitivity analysis of key hyperparameters on the IEMOCAP dataset.

Impact of Multirepresentations. We first examine the role of modality-specific and modality-invariant representations by selectively removing each from the multimodal fusion process. Excluding modality-specific representations results in a notable decline in emotion recognition accuracy across both datasets, confirming their essential role in capturing and leveraging unique information carried by each modality. Conversely, omitting modality-invariant representations also leads to a significant performance drop, underscoring their importance in aligning heterogeneous modalities. As expected, removing both types of representations further exacerbates the degradation, demonstrating that both components are indispensable for robust multimodal emotion recognition.

Impact of Multitasks. Next, we investigate the contributions of the AED and AEC modules. When the AED module is removed and only the AEC module is retained, the model is forced to correct all errors in an utterance without guidance on error positions, rather than focusing solely on detected errors. This setting results in a substantial decrease in recognition performance, highlighting the necessity of AED. Likewise, eliminating the AEC module also degrades performance, confirming its complementary role. Interestingly, as shown in Table 7.2(C)(1), using only the AEC module yields lower WA than removing both AED and AEC altogether (see Table 7.2(C)(3)) in both datasets. This phenomenon arises because applying a neural machine translation (NMT)-style approach directly to AEC may inadvertently increase WER [185]. Unlike typical NMT tasks, where most tokens require modification, AEC involves correcting only a small fraction of tokens—approximately 10% when the ASR WER is 10%. However, these tokens tend to be the most challenging cases, as they are already misrecognized by the ASR system. Consequently, an effective design must account for the

characteristics of ASR outputs, which motivates the joint use of both AED and AEC as auxiliary tasks in our framework.

Impact of Multistrategies. To further examine the contribution of the proposed training strategies, we first remove the adversarial learning strategy described in Alg. 1. As shown in Table 7.2(D)(1), this results in a performance decline on both datasets, confirming the essential role of adversarial learning in fostering modality-invariant representations. The modality discriminator plays a pivotal role by enforcing modality agnosticism and reducing modality-specific biases in the representations generated by the MIR module.

In addition, excluding the label-based contrastive learning (LCL) strategy introduced in Section 6.2.8 also causes a similar decrease in recognition accuracy (Table 7.2(D)(2)). To better illustrate its effectiveness, we further visualize the feature distributions before and after applying LCL, with detailed analysis provided in the subsequent t-SNE section. When both adversarial learning and LCL are omitted, the degradation in performance becomes even more pronounced, indicating that these two strategies complement each other in improving multimodal learning.

The ablation experiments collectively demonstrate that each component of the M^{4}SER framework—multimodality, multirepresentation, multitask, and multistrategy learning—plays a critical role in achieving robust emotion recognition. The integration of these strategies equips M^{4}SER with the ability to maintain high accuracy and robustness, even in the presence of ASR errors, thereby underscoring the overall effectiveness of our proposed framework.

Table 6.5: Trade-off between computational cost and performance on IEMOCAP. Parameter size (Params) is measured in millions (M). Training time is reported over 100 epochs in hours (h), and inference time is averaged per utterance in milliseconds (ms/U).

Type	Model	Modality	Params (M)	Train Time (h)	Infer Time (ms/U)	WA (%)	UA (%)
Single-modal	HuBERT	S	316.2	5.4	23.7 ± 0.1	68.9	69.8
	BERT	T(ASR)	109.5	0.5	6.3 ± 0.1	65.1	66.4
Multi-modal	Baseline (HuBERT + BERT)	S+T(ASR)	426.9	9.6	29.6 ± 0.1	73.4	75.2
	+ MSR & MIR		481.4	12.0	44.8 ± 0.1	76.6	77.4
	+ GAN & LCL		481.6	20.8	44.5 ± 0.6	77.3	78.6
	+ AED & AEC (M^{4}SER)		**512.9**	**21.5**	**44.7 ± 0.2**	**79.2**	**80.1**

6.4.3 Sensitivity Analysis

To better understand the influence of key hyperparameters on model performance, we perform a one-at-a-time sensitivity analysis on four parameters: the auxiliary task weight α, the balance factor

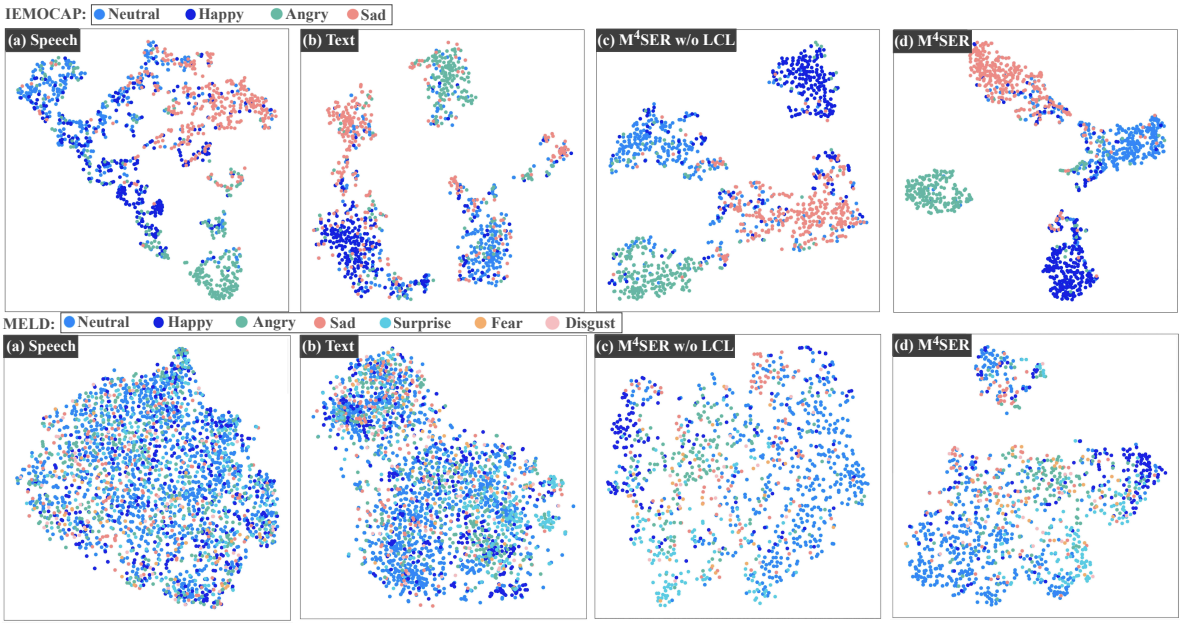


Figure 6.8: t-SNE visualization using IEMOCAP and MELD datasets. We visualize all samples from IEMOCAP and MELD test sets.

between AED and AEC β, the GAN loss weight γ, and the LCL loss weight λ. The results are presented in Fig. 6.7.

The analysis shows that model performance is moderately affected by α and λ, with the best results obtained when $\alpha = 0.1$ and $\lambda = 0.1$. Setting these parameters too low weakens the contribution of auxiliary supervision, whereas excessively large values introduce interference with the primary task.

By contrast, the system demonstrates greater robustness to changes in β, suggesting that the interplay between the AED and AEC objectives remains relatively stable across different values. In comparison, performance is highly sensitive to γ: larger values tend to destabilize training due to the adversarial optimization process, whereas smaller values (e.g., $\gamma = 0.01$) promote stable convergence and yield superior results.

6.4.4 Computational Complexity Analysis

To evaluate the balance between computational efficiency and recognition performance, we report in Table 6.5 the number of parameters, training time, and inference time of each model on the IEMOCAP dataset. All experiments were conducted on a single NVIDIA Tesla V100 GPU.

For single-modal baselines, HuBERT incurs a higher computational cost compared with BERT, which is expected given the complexity of speech encoders. Nevertheless, HuBERT achieves superior

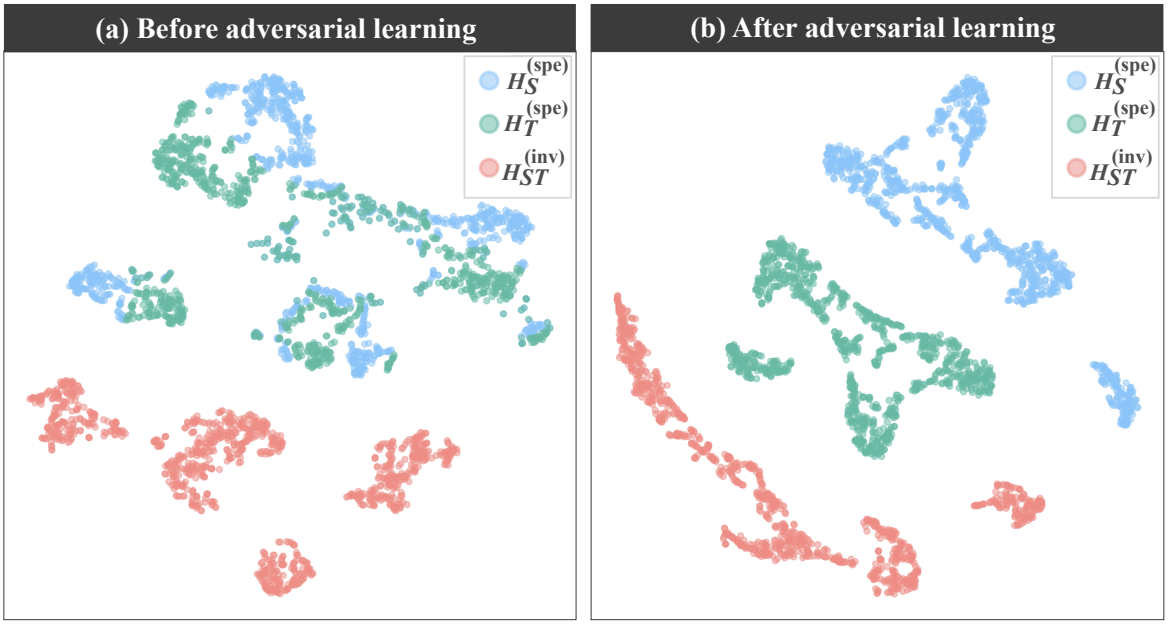


Figure 6.9: t-SNE visualizations of the distribution of the modality-specific and modality-invariant representations before and after adversarial learning on the IEMOCAP dataset.

recognition accuracy, highlighting the trade-off between model complexity and performance.

In the multimodal configuration, the inclusion of additional modules results in a moderate increase in both parameter size and training cost, but consistently contributes to performance improvements. The complete M^{4}SER system achieves the best overall performance, reaching a WA of 79.2% and a UA of 80.1%. While training time extends to 21.5 hours due to the incorporation of adversarial and contrastive learning strategies as well as the AED and AEC auxiliary tasks, it is important to note that these components are only active during training. Consequently, the inference time of M^{4}SER remains almost identical to that of the baseline system containing only MSR and MIR.

Overall, the observed results demonstrate that M^{4}SER achieves a favorable balance between computational overhead and recognition performance, thereby underscoring its practicality for real-world applications.

6.4.5 Visualization Analysis

t-SNE Analysis. To provide an intuitive understanding of the effectiveness of the proposed M^{4}SER model on the IEMOCAP and MELD datasets, we employ the t-distributed stochastic neighbor embedding (t-SNE) algorithm [186] to visualize the learned emotion representations. Specifically, we utilize all samples from session 3 of IEMOCAP and the test set of MELD. The visualization compares four models: the speech-only baseline (Fig. 6.8(a)), the text-only baseline (Fig. 6.8(b)), the proposed

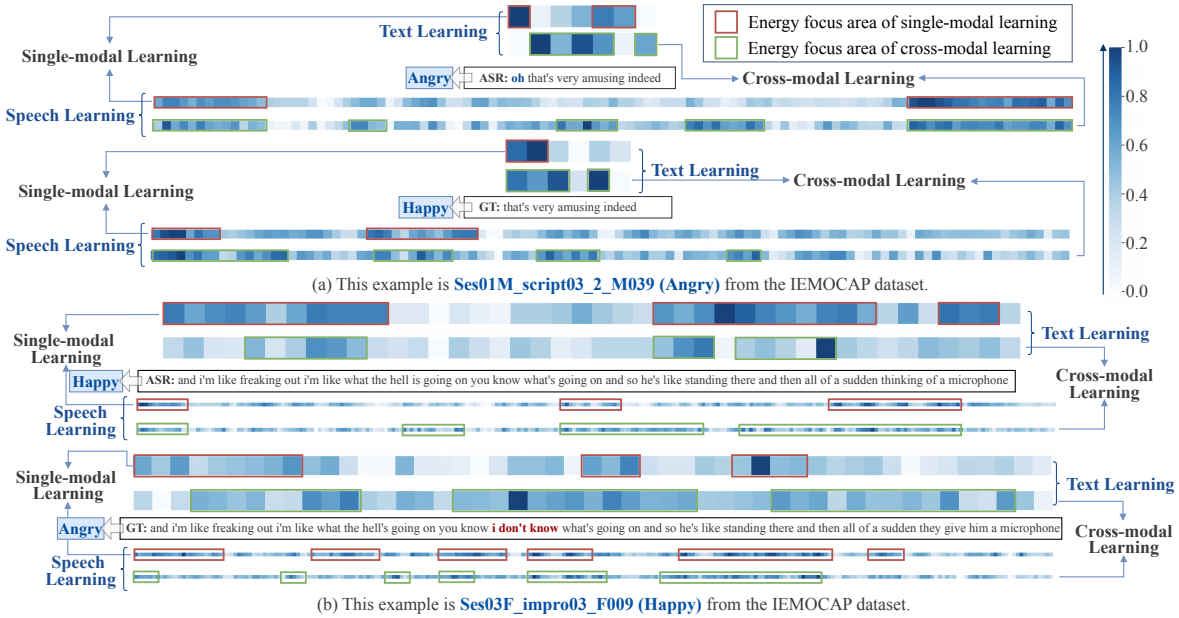


Figure 6.10: Representation weights of temporal-level features under different text input conditions in different types of modal learning. Brighter colors (tending towards blue) indicate higher values, suggesting the coverage of more key information.

M^{4}SER model without LCL (Fig. 6.8(c)), and the full M^{4}SER model (Fig. 6.8(d)).

As illustrated in Fig. 6.8, the single-modal baselines on both IEMOCAP and MELD exhibit substantial overlap among different emotion categories, indicating considerable confusion between labels. In contrast, the two multimodal variants yield more clearly separated emotion clusters, confirming the advantages of multimodal learning in capturing richer and more discriminative emotional representations.

Furthermore, when comparing the M^{4}SER model with and without the LCL strategy, it is evident that LCL facilitates tighter intra-class clustering and clearer inter-class boundaries. On IEMOCAP, for example, the full M^{4}SER model (Fig. 6.8(d)) shows noticeably sharper boundaries between *Sad* and *Angry* categories than the version without LCL (Fig. 6.8(c)). In addition, the distances between emotion categories in the embedding space become more pronounced. Although classification on MELD remains more challenging than on IEMOCAP, the overall separation trend is consistent across datasets. These observations confirm that M^{4}SER effectively integrates modality-specific and modality-invariant information, leading to high-level shared representations across speech and text modalities that are well suited for emotion recognition.

To further examine the role of adversarial learning in M^{4}SER, we visualize the distributions of modality-invariant and modality-specific representations before and after adversarial training using t-SNE (see Fig. 6.9). The results reveal that adversarial learning improves the separation of modality-

Table 6.6: Performance comparison of our method on IEMOCAP using ASR and GT texts (%). When using GT text, the AED and AEC modules are excluded.

Method	Modality	WA	UA
M^{4}SER	S+T(GT)	78.4	79.9
M^{4}SER	**S+T(ASR)**	**79.2**	**80.1**

Table 6.7: Consistency Analysis between Cross-modal Speech and Text Representations Using ASR and GT Texts.

Metric	Example (a)		Example (b)	
	ASR	GT	ASR	GT
DTW ↓	**0.52**	1.89	**1.46**	1.57

specific features while enhancing intra-class cohesion. This indicates that adversarial learning not only increases the diversity of modality-specific representations but also contributes to generating more robust and informative features for downstream emotion recognition tasks.

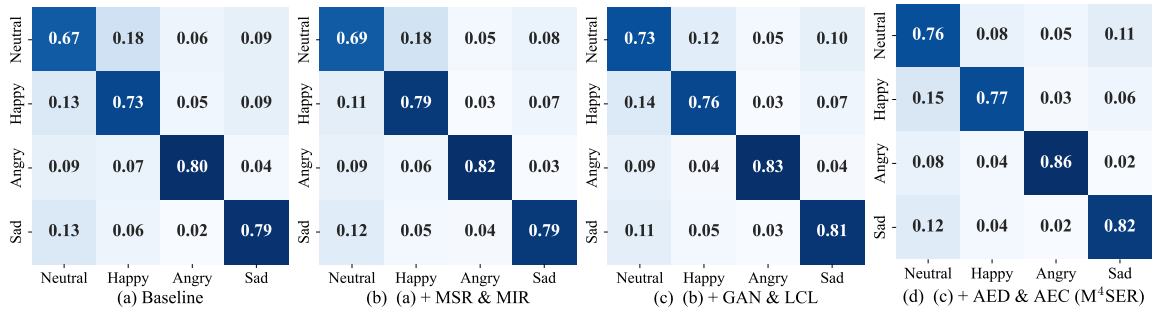


Figure 6.11: Confusion matrices obtained using IEMOCAP datasets. We utilize the results from five-fold cross-validation. Columns represent predicted labels and rows represent true labels.

Representation Analysis. An interesting observation from Table 6.6 is that M^{4}SER achieves superior performance when provided with ASR transcripts compared to GT transcripts. To further investigate this phenomenon, we visualize the representation weights of two utterances from IEMOCAP in Fig. 6.10. Specifically, we project the hidden features H_S, H_T, $\hat{H}_S^{(\text{spe})}$, and $H_T^{(\text{spe})}$ onto a single dimension to display the temporal distribution of representations across speech and text modalities in different learning spaces.

Table 6.8: Analysis of cross-corpus generalization ability on a 4-class emotion classification task (%). We reimplement the method indicated by ° and obtain the corresponding result. *IE* $\rightarrow$ *ME* indicates that the IEMOCAP dataset is used for training and MELD is used for testing. Conversely, *ME* $\rightarrow$ *IE* means the model is trained on MELD and evaluated on IEMOCAP.

Methods	**IE $\rightarrow$ ME**		**ME $\rightarrow$ IE**	
	ACC	**W-F1**	**WA**	**UA**
SAMS° [175]	17.0	11.6	36.3	33.9
MF-AED-AEC° [9]	56.0	54.2	62.5	60.6
Baseline (HuBERT + BERT)	53.0	52.4	58.9	57.4
+ MSR & MIR	54.7	53.2	60.8	59.9
+ GAN & LCL	55.6	53.9	61.9	60.7
+ AED & AEC (M^4SER)	**57.3**	**55.5**	**64.0**	**62.2**
Δ_{Baseline}	4.3↑	3.1↑	5.1↑	4.8↑

As shown in Fig. 6.10(a), in the single-modal case using ASR transcripts, the text modality initially places strong emphasis on the word "oh," while the speech modality concentrates towards the utterance's ending region. After cross-modal fusion, however, the weight assigned to "oh" in the text modality decreases noticeably. This adjustment occurs because the AED and AEC modules detect transcriptional inaccuracies, redistributing attention to other informative words such as "that's." Given the limited emotional cues in ASR text, the model increasingly relies on speech signals, particularly prosodic patterns such as intonation and intensity, which are critical for recognizing anger. In contrast, with GT transcripts, cross-modal learning directs attention in the text modality towards explicitly positive words such as "amusing," while the speech modality distribution remains largely unchanged. Without ASR-induced error signals, the model disproportionately favors textual features, often biasing recognition towards happiness.

Fig. 6.10(b) further illustrates a mismatch between GT transcripts and speech features. Although the GT text contains words like "don't know" that suggest uncertainty, the corresponding speech features reveal strong emotional fluctuations. This inconsistency complicates cross-modal alignment and leads to misclassification as anger. By comparison, ASR transcripts—despite containing recognition errors—highlight emotionally salient expressions such as "freaking out," "what the hell," and "all of a sudden," which align more closely with acoustic cues. The representation weights confirm

Table 6.9: Analysis of few-shot domain adaptation ability for cross-corpus 4-class emotion recognition (%). We reimplement the method indicated by ° and obtain the corresponding result. *IE* → *ME* and *ME* → *IE* denote training on IEMOCAP and MELD, respectively, with the other used for testing.

Methods	**# Shots**	**IE → ME**		**ME → IE**	
		ACC	**W-F1**	**WA**	**UA**
SAMS° [175]	5	51.6	49.0	39.0	37.2
	10	52.5	49.2	38.2	38.3
	20	52.3	52.3	39.9	35.0
	60	59.9	56.8	44.4	45.3
MF-AED-AEC° [9]	5	56.3	54.8	64.4	63.3
	10	57.8	55.7	65.8	65.6
	20	57.5	56.7	67.6	66.6
	60	60.8	60.3	72.5	73.7
M^{4}SER	5	**57.9**	**56.0**	**66.4**	**64.4**
	10	**58.7**	**58.0**	**66.7**	**67.7**
	20	**59.4**	**58.8**	**69.0**	**68.7**
	60	**62.1**	**61.5**	**74.0**	**75.9**

this alignment, showing high consistency between ASR transcripts and speech features across multiple regions, thereby facilitating accurate cross-modal integration and correct classification as "Happy" ("Excited"). To further validate this observation, we apply dynamic time warping (DTW) and demonstrate that ASR transcripts exhibit stronger alignment with speech than GT transcripts (Table 6.7).

In summary, the differences between ASR and GT transcripts can be attributed to two primary factors:

1. **Absence of ASR error signals:** GT transcripts lack ASR errors, thereby excluding auxiliary cues that can aid emotion recognition. This absence may reduce the ability to capture strong emotional patterns.

2. **Differences in feature weight distributions:** The absence of error-driven cues alters the weight distribution in the GT text modality compared to ASR transcripts, which may lead to suboptimal alignment and erroneous emotion predictions.

Confusion Matrix Analysis. To further examine the influence of individual modules on class-wise performance, we present the averaged confusion matrices over five folds on the IEMOCAP dataset in Fig. 6.11. Relative to the baseline system, the incorporation of MSR and MIR leads to a clear improvement in the recognition of the "Happy" class, which is frequently subject to ambiguity due to challenges in aligning acoustic and textual cues.

With the additional integration of the adversarial learning and LCL strategies, we observe more consistent gains across most emotion categories, particularly in "Neutral." Nonetheless, the accuracy for the "Happy" class shows a slight decline. This reduction may be attributed to the heterogeneity between acoustic and textual modalities, especially since "Excited" is merged into "Happy," which complicates local contrastive learning and diminishes its ability to capture class-specific discriminative features.

Finally, when the AED and AEC modules are incorporated, the model achieves the most accurate and balanced predictions overall. Further gains are observed in the "Neutral" and "Sad" classes, while the performance of the "Happy" class also shows partial recovery compared with the previous configuration. These findings demonstrate that the multitask learning design, which explicitly addresses ASR-induced errors, is instrumental in improving class-level robustness and ensuring more reliable emotion recognition across modalities.

6.4.6 Cross-corpus Generalization Ability

To assess the applicability of M^{4}SER in real-world conditions, we evaluate its performance under cross-corpus emotion recognition settings, thereby simulating practical scenarios where domain shift occurs between training and testing data. Following the protocols adopted in prior studies [187–189], we employ a transfer evaluation strategy: one corpus is used exclusively for training, 30% of the target corpus is held out as a validation set for parameter tuning, and the remaining 70% is reserved for final testing. Since the IEMOCAP dataset contains only four emotion categories ("Neutral", "Happy", "Angry", and "Sad"), we restrict the MELD dataset to the same overlapping categories by discarding non-overlapping classes ("Surprise", "Fear", and "Disgust"), thereby ensuring a consistent label space across corpora during both training and evaluation.

The cross-corpus generalization results are summarized in Table 6.8. Compared with the reimplemented SAMS and MF-AED-AEC baselines, our complete M^{4}SER framework achieves the strongest performance in both transfer directions (IE→ME and ME→IE). Specifically, in the IE→ME setting, M^{4}SER surpasses the multimodal baseline by 4.3% in ACC and 3.1% in W-F1. In the ME→IE setting, it further achieves gains of 5.1% in WA and 4.8% in UA.

To further assess adaptability under limited supervision, we perform few-shot domain adapta-

tion experiments by sampling 5, 10, 20, and 60 labeled examples per class from the target-domain validation set. The remainder of the validation data continues to serve for model selection, while the test set remains unchanged across all conditions. As reported in Table 6.9, M^{4}SER consistently outperforms both SAMS and MF-AED-AEC across all few-shot settings. For instance, with only 5 labeled samples per class, M^{4}SER attains 57.9% ACC in the IE→ME transfer, already exceeding the performance of MF-AED-AEC with 20 labeled samples. With 60 labeled samples, the model scales effectively, reaching 62.1% ACC and 61.5% W-F1. In the ME→IE transfer, M^{4}SER achieves 75.9% UA, further underscoring its strong cross-domain adaptability.

In summary, these results highlight that M^{4}SER not only demonstrates robust zero-shot generalization across corpora with minimal parameter tuning, but also exhibits strong scalability and adaptability in few-shot scenarios, confirming its practicality for real-world deployment.

6.5 Summary

In this chapter, we presented M^{4}SER, a novel framework for speech emotion recognition that integrates multimodal, multirepresentation, multitask, and multistrategy learning paradigms. The proposed model incorporates an advanced multimodal fusion module designed to jointly learn modality-specific and modality-invariant representations, thereby capturing both distinctive features of individual modalities and shared characteristics across modalities. To further improve robustness, we introduced a modality discriminator trained with adversarial learning, which enhances modality diversity and mitigates modality mismatch. Moreover, two auxiliary tasks—AED and AEC—were employed to strengthen semantic consistency in the text modality. In addition, a label-based contrastive learning strategy was developed to improve the discriminability of emotion-related features.

Extensive experiments conducted on the IEMOCAP and MELD datasets demonstrate that M^{4}SER consistently outperforms existing state-of-the-art baselines, thereby validating the effectiveness of our proposed framework. Looking ahead, we plan to extend M^{4}SER by incorporating visual modalities and exploring disentangled representation learning to further enhance emotion recognition performance. Furthermore, we intend to investigate the necessity of AED and AEC modules in the context of increasingly powerful LLM-based ASR systems, where transcription accuracy may alleviate the dependence on explicit error detection and correction mechanisms.

Chapter 7

2DP-2MRC: 2-Dimensional Pointer-based Machine Reading Comprehension Method for Multimodal Moment Retrieval

In the previous chapters, we investigated ASR and its downstream application, namely multimodal SER, where both speech signals and ASR transcripts were leveraged to improve affective understanding. Building upon this foundation, we now turn to a broader multimodal retrieval task that integrates not only audio and textual modalities but also visual information: multimodal VMR.

Moment retrieval requires the system to localize the most relevant temporal segment in an untrimmed video given a natural language query. This task poses additional challenges compared with purely speech-based tasks, as it must align heterogeneous information sources—visual content, the corresponding audio track, and the ASR-generated transcripts—to ensure accurate semantic grounding. While moment-based approaches provide fine-grained localization but incur substantial computational costs, clip-based methods offer efficiency but often suffer from imprecise boundary detection.

To address these limitations, in this chapter we introduce the **2-D**imensional **P**ointer-based **M**achine **R**eading **C**omprehension for **M**oment **R**etrieval **C**hoice (2DP-2MRC) framework. Specifically, our method employs an AV-Encoder to jointly capture coarse-grained information at both the video and moment levels, and integrates ASR-derived textual cues to complement visual features. Furthermore, a two-dimensional pointer encoder is designed to enhance temporal boundary prediction. Through extensive experiments on the HiREST dataset, we demonstrate that 2DP-2MRC achieves superior

performance compared with existing baseline models, striking a favorable balance between retrieval accuracy and computational efficiency.

7.1 Introduction

With the rapid proliferation of online video content, the demand for efficient and accurate retrieval of specific information from large-scale video repositories has become increasingly critical [190–192]. To meet this demand, Hendricks et al. [103] and Gao et al. [104] introduced the task of VMR, which requires the system to identify the temporal boundaries of video segments corresponding to a given natural language query in untrimmed videos [105, 193, 194], as illustrated in Fig. 7.1. This task is inherently challenging because it involves aligning complex multimodal content (video and potentially audio) with linguistic queries while simultaneously reasoning over semantics across different modalities [10, 106].

Existing VMR methods can be broadly categorized into moment-based and clip-based approaches, depending on whether predefined candidate moments are explicitly generated [195]. Since clips are the shortest temporal units, their contextual information is naturally a subset of moment-level context. Moment-based approaches typically follow a "propose-then-rank" paradigm [103, 104, 106, 107, 196], where a large set of candidate moments is first generated, each candidate is compared with the query, and the most relevant one is then selected via ranking. While such approaches are intuitive and often effective, they suffer from high computational cost due to dense candidate generation.

In contrast, clip-based approaches directly model the alignment between video clips and queries, predicting their relevance without explicitly enumerating candidate moments [108, 109, 197, 198]. Although this strategy substantially reduces computational complexity, existing methods generally emphasize clip-level alignment while overlooking coarser moment-level and global video-level information. As a result, they frequently yield suboptimal temporal localization and limited retrieval accuracy.

To overcome the limitations of excessive computational cost in moment-based methods and the imprecise localization inherent to clip-based approaches, we propose a novel framework termed **2**-**D**imensional **P**ointer-based **M**achine **R**eading **C**omprehension for **M**oment **R**etrieval **C**hoice (2DP-2MRC) in this chapter. The design of this model is inspired by the process of human reading comprehension. Typically, when answering a question, humans begin by skimming the passage and question to form a preliminary understanding. They then revisit the passage, paying attention to regions most relevant to the query, and progressively integrate contextual cues with the question to identify the correct answer [199–202].

In line with this analogy, our model incorporates mechanisms that jointly enhance coarse-grained

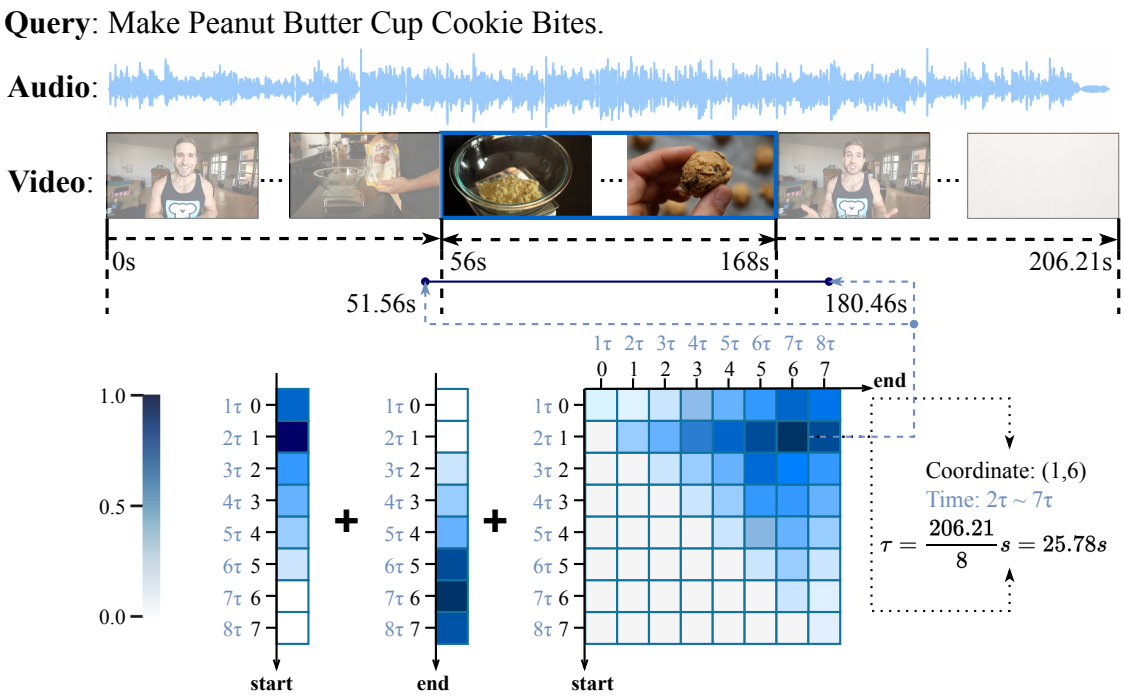


Figure 7.1: An example of our multimodal moment retrieval with a query in an untrimmed video. The most relevant moment is retrieved by two 1D probability matrices and a 2D probability matrix together. Note that the length of the video and the sampling rate determine the value of the short duration τ and the precision of the retrieved moments.

comprehension and fine-grained localization of video content. Specifically, the framework integrates an Audio-Video Encoder (AV-Encoder), a pointer mechanism, a Gated Interactive Attention (GIA) module, and a 2D Probability Encoder (2DP-Encoder). The key contributions of this work are summarized as follows:

- We propose an AV-Encoder designed to capture coarse-grained contextual information at both the moment and video levels, thereby enriching the global understanding of video semantics.
- We design a pointer module that leverages multi-level interactive attention to provide initial predictions for the start and end boundaries of target moments.
- We develop a 2DP-Encoder module that employs a GIA mechanism for fine-grained boundary localization. In this module, an adaptive gating unit is first applied to regulate the contribution of each video clip to the answer. Subsequently, multi-level interactive attention is performed between the query and video context, enabling stronger focus on the target moment. By constructing a 2D positional coordinate matrix across potential boundaries, this module enhances recall of candidate segments and alleviates inconsistencies arising from independent 1D predictions, thus yielding more accurate retrieval.

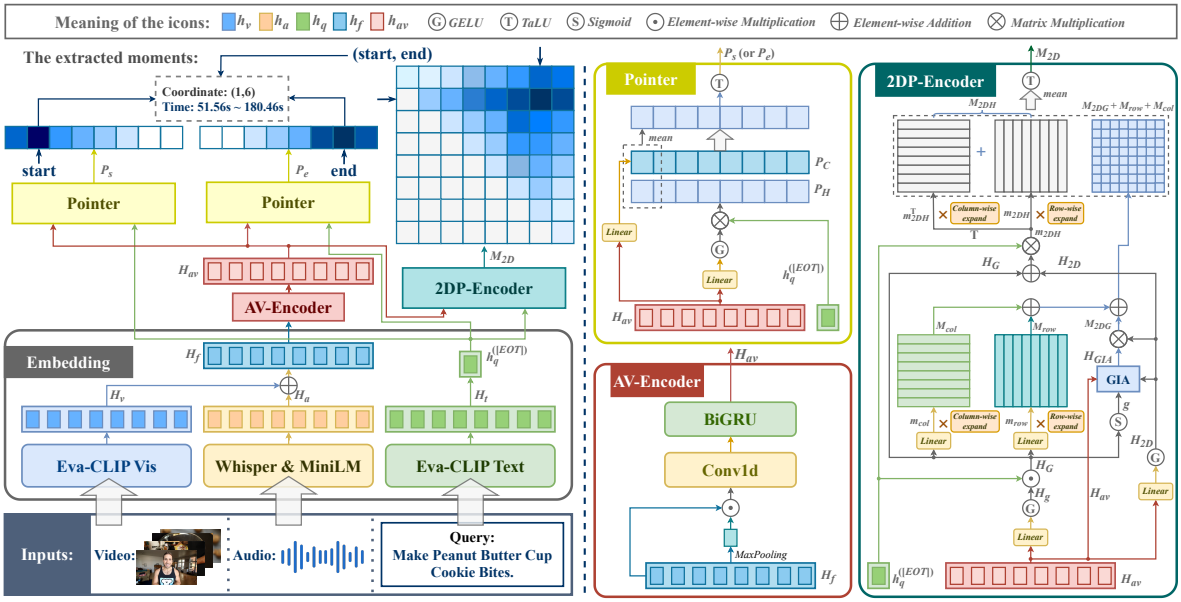


Figure 7.2: Overall architecture of the proposed 2DP-2MRC model.

- We introduce the complete 2DP-2MRC framework, which integrates the AV-Encoder, Pointer, and 2DP-Encoder modules. Benefiting from its parallelizable architecture, the model avoids excessive computational overhead while maintaining high accuracy. Experimental evaluation on the HiREST dataset demonstrates that our proposed 2DP-2MRC achieves substantial improvements over existing baseline approaches.

7.2 Proposed Method

7.2.1 Problem Formulation

We formalize the multimodal moment retrieval task as a mapping function $f(V,A,Q) = (t^s, t^e)$. Here, the video modality is denoted as $V = (v_1, v_2, \cdots, v_m)$, where m refers to the number of video clips. The audio modality, extracted from the corresponding video, is represented as $A = (a_1, a_2, \cdots, a_n)$, consisting of n audio frames. The textual modality is defined as the natural language query $Q = (q_1, q_2, \cdots, q_l)$, comprising l words.

The objective of multimodal moment retrieval is to effectively integrate the visual content V and the acoustic information A in order to localize a temporal segment within the video. This segment is defined by its start and end timestamps (t^s, t^e) and should correspond most closely to the semantics of the query Q.

7.2.2 Embedding Module

In line with prior studies [11], the embedding module of our framework is constructed upon three pretrained backbone models: EVA-CLIP [203], Whisper [204], and MiniLM [205], as illustrated in Fig. 7.2. All pretrained parameters are kept frozen throughout the training phase to preserve their generalization ability.

- **Video Clip Representations.** Given a video composed of o continuous frames, we segment it into short clips, each containing p frames, and uniformly sample m clips at fixed intervals. These clips are then encoded by the EVA-CLIP visual encoder, yielding visual embeddings $H_v = (h_v^{(1)}, h_v^{(2)}, \cdots, h_v^{(m)}) \in \mathbb{R}^{m \times d}$, where d denotes the dimensionality of the hidden representation.
- **Text Query Representations.** The text query is processed by the EVA-CLIP text encoder to produce embeddings $H_q = (h_q^{(1)}, h_q^{(2)}, \cdots, h_q^{(l)}) \in \mathbb{R}^{l \times d}$. In addition, the end-of-text token ($[EOT]$) is employed to summarize the global semantic meaning of the query, represented as $h_q^{([EOT])} \in \mathbb{R}^d$.
- **ASR Representations.** The Whisper model generates speech transcriptions from the audio track, which are subsequently encoded by MiniLM to obtain ASR embeddings $H_a = (h_a^{(1)}, h_a^{(2)}, \cdots, h_a^{(m)}) \in \mathbb{R}^{m \times d}$. To ensure alignment with the m video clips, the ASR embeddings are uniformly sampled or padded to length m.
- **Fused Representations.** Following the design of [11], the fused representation is obtained by applying element-wise addition between the visual embeddings and the ASR embeddings: $H_f = (h_f^{(1)}, h_f^{(2)}, \cdots, h_f^{(m)}) \in \mathbb{R}^{m \times d}$. This fusion provides a joint representation incorporating both visual and audio-textual information for subsequent processing.

7.2.3 AV-Encoder Module

To extract coarse-grained features from the fused representation H_f, we first apply a MaxPooling operation across clip-level features and combine the resulting pooled vector with the original clip-wise representations through element-wise multiplication. This process enriches the fused embedding with global contextual cues while retaining local clip-specific details.

Subsequently, the resulting sequence is passed through a temporal one-dimensional convolutional layer [206], followed by a bidirectional GRU [207]. The convolutional layer captures local temporal patterns among adjacent clips, while the BiGRU models long-range sequential dependencies in both forward and backward directions. The final audio-visual representation is expressed as

$$H_{av} = \text{BiGRU}(\text{Conv1d}(\text{Broadcast}(\text{MaxPooling}(H_f)) \odot H_f)) \in \mathbb{R}^{m \times d}, \tag{7.1}$$

where $\odot$ denotes element-wise multiplication.

7.2.4 Pointer Module

The pointer module is designed to map the fused audio–visual representation H_{av} of video clips into a probability distribution vector of length m, thereby generating preliminary estimates of the start and end positions of the target moment. Distinct from earlier approaches that employ only linear mappings [11], our design further incorporates the global query representation $h_q^{([EOT])}$ to compute multi-level interactive attention between the query, the fused audio–visual features, and contextual information. This mechanism enhances the model's ability to emphasize segments most relevant to the query.

To begin, H_{av} is passed through a linear transformation followed by a GELU activation function [208], yielding a refined representation H:

$$H = \mathrm{GELU}(\mathrm{Linear}(H_{av})) \in \mathbb{R}^{m \times d}. \tag{7.2}$$

Next, the interaction between H and the global query embedding $h_q^{([EOT])}$ produces a candidate probability distribution P_H:

$$P_H = \mathrm{TaLU}(H \cdot h_q^{([EOT])}) \in \mathbb{R}^{m \times 1}, \tag{7.3}$$

where TaLU is a normalized activation function that maps attention scores into $(0,1)$. Compared with the conventional Sigmoid function, TaLU provides a derivative range four times larger, thereby enhancing the model's capacity to differentiate probabilities across positions and improving the precision of boundary detection:

$$\mathrm{TaLU}(x) = \frac{e^x}{e^x + e^{-x}} \in (0,1). \tag{7.4}$$

In parallel, H_{av} is also processed through a linear projection that maps each hidden state to a single scalar score, generating an additional candidate distribution P_C:

$$P_C = \mathrm{TaLU}(\mathrm{Linear}(H_{av})) \in \mathbb{R}^{m}. \tag{7.5}$$

Finally, the preliminary probability distribution for the start (or end) position, denoted P_s (or P_e), is obtained as the mean of P_H and P_C. This combined strategy balances query-guided attention with direct clip-level scoring, thereby providing a more reliable initialization for moment boundary prediction.

7.2.5 2DP-Encoder Module

The 2DP-Encoder is designed to enhance boundary detection and improve recall in moment retrieval by projecting the input representation H_{av} into a two-dimensional probability distribution matrix of

size $m \times m$.

First, H_{av} is passed through two independent linear layers followed by GELU activation [208], producing two separate representations H_g and H_{2D}:

$$H_g = \text{GELU}(\text{Linear}(H_{av})) \in \mathbb{R}^{m \times d}, \tag{7.6}$$

$$H_{2D} = \text{GELU}(\text{Linear}(H_{av})) \in \mathbb{R}^{m \times d}. \tag{7.7}$$

Next, H_g interacts with the global query embedding $h_q^{([EOT])}$ to yield H_G, which is then normalized by a Sigmoid activation to obtain the gating weights g:

$$H_G = H_g \odot h_q^{([EOT])} \in \mathbb{R}^{m \times d}, \tag{7.8}$$

$$g = \text{Sigmoid}(H_G) \in \mathbb{R}^{m \times d}. \tag{7.9}$$

These gating weights adaptively modulate the importance of video clips for moment localization. The gated representation is then obtained as

$$H_{GIA} = g \odot H_{2D} + (1 - g) \odot H_{av} \in \mathbb{R}^{m \times d}. \tag{7.10}$$

Subsequently, we compute the interactive attention between H_{GIA} and H_{2D}, resulting in the first candidate 2D probability distribution matrix M_{2DG}:

$$M_{2DG} = \text{TaLU}(H_{2D} \cdot H_{GIA}^{\top}) \in \mathbb{R}^{m \times m}, \tag{7.11}$$

where T denotes matrix transpose.

In parallel, the query embedding $h_q^{([EOT])}$ also interacts with H_G and H_{2D}. The resulting vector m_{2DH} is expanded row- and column-wise to produce the second candidate distribution M_{2DH}:

$$\begin{aligned} m_{2DH} = \text{TaLU}((H_G + H_{2D}) \cdot h_q^{([EOT])}) \in \mathbb{R}^m \\ \rightarrow M_{2DH} \in \mathbb{R}^{m \times m}. \end{aligned} \tag{7.12}$$

To further enrich the probability representation, the 2DP-Encoder incorporates row- and column-wise expansions of H_G, generating M_{row} and M_{col}:

$$m_{row} = \text{TaLU}(\text{Linear}(H_G^{\text{T}})) \in \mathbb{R}^m \rightarrow M_{row} \in \mathbb{R}^{m \times m}, \tag{7.13}$$

$$m_{col} = \text{TaLU}(\text{Linear}(H_G)) \in \mathbb{R}^m \rightarrow M_{col} \in \mathbb{R}^{m \times m}. \tag{7.14}$$

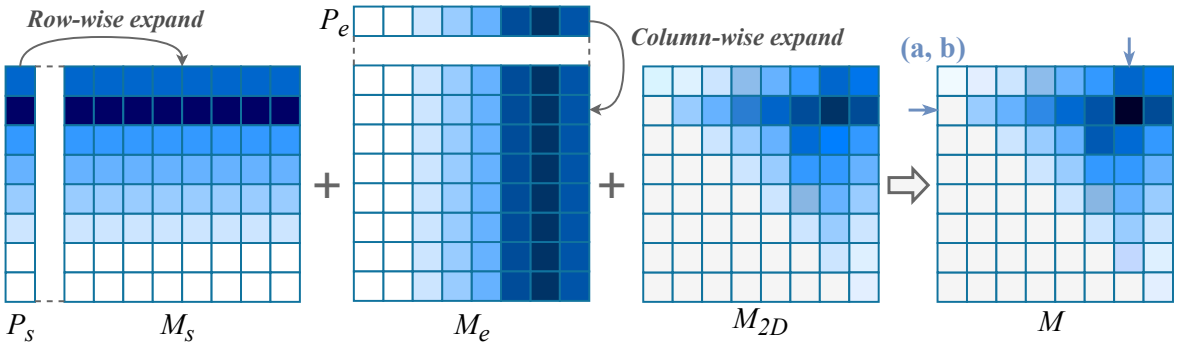


Figure 7.3: The process of score prediction.

Finally, the overall 2D probability distribution M_{2D} is computed as the mean of the four candidate matrices:

$$M_{2D} = \frac{1}{4}(M_{2DG} + M_{2DH} + M_{row} + M_{col}). \tag{7.15}$$

This design integrates query-guided gating, cross-modal interaction, and 2D probability modeling, thereby enhancing moment boundary detection while maintaining computational efficiency.

7.2.6 Score Prediction

During scoring, we expand P_s and P_e into M_s and M_e by row and column, respectively. The final 2D score map M is formed by adding M_s, M_e, and M_{2D} (see Fig. 7.3). The coordinates (a,b) in M represent a moment starting from clip a to b. To maintain validity, the start and end clip indexes a and b must satisfy $a \leq b$. Thus, we set all elements in the lower triangle of M to zero and choose the highest score as the retrieved moment.

7.2.7 Loss Function

The training loss function comprises three weighted loss terms: the loss between the predicted start position P_s by the Pointer module and the target start position distribution Y_s, the loss between the predicted end position P_e by the Pointer module and the target end position distribution Y_e, and the loss between the predicted 2D probability distribution M_{2D} by the 2DP-Encoder module and the target position coordinate distribution Y_m. Here, Y_s, Y_e, and Y_m are one-hot labels.

$$\text{Loss} = \frac{1-\lambda}{2}\left[f(P_s,Y_s) + f(P_e,Y_e)\right] + \lambda \cdot f(M_{2D},Y_m), \tag{7.16}$$

where $f(x,y)$ represents the Binary Cross Entropy (BCE) loss between predictions and ground-truth labels and λ is a hyperparameter to adjust the weight of 2D probability distribution.

7.3 Experiments

7.3.1 Experimental Settings and Evaluation Metrics

The proposed method was implemented in Python 3.10.0 with PyTorch 1.11.0. All experiments were conducted on a workstation equipped with an Intel(R) Xeon(R) Gold 6248 CPU running at 2.50 GHz, 32 GB of RAM, and a single NVIDIA Tesla V100 GPU. The training process employed the AdamW optimizer [156] with a batch size of 16 and a weight decay parameter of 0.01. The dropout rate was fixed at 0.3, and the weighting factor λ for the 2D probability distribution was set to 0.1. The initial learning rate was configured as 1×10^{-3}, and the dimensionality of all hidden states in the model was set to 512.

For evaluation, we followed the commonly adopted protocols in prior studies [11, 201, 209]. Specifically, model predictions were assessed against ground-truth moments using Recall@1 under Intersection-over-Union (IoU) thresholds of 0.5 and 0.7.

7.3.2 Datasets

To validate the effectiveness of the proposed model, we conducted experiments on the HIREST dataset [11], which is designed for a range of multimodal video understanding tasks, including video retrieval, moment retrieval, moment segmentation, and step captioning. In this work, we specifically focus on the moment retrieval task.

The HIREST dataset contains approximately 3.4K text–video pairs, with videos averaging 287 seconds in duration and an overall length of 270 hours. Among these, 1.8K videos are associated with clip-level moment annotations. The average moment length is 148 seconds, corresponding to about 55% of the original video duration. Since multiple videos may correspond to a single query, the dataset is partitioned based on queries rather than videos, yielding splits of 546/292/546 queries (corresponding to 1507/477/1391 videos) for training, validation, and testing, respectively.

7.3.3 Results and Analysis

Experimental Results

We evaluate the proposed model against three representative baselines:

- **BMT** [209]: A moment-based dense video captioning framework pretrained on ActivityNet Captions. It predicts event proposals characterized by center, length, and confidence values. The retrieved moment is then obtained by aggregating proposals through the minimum start time and maximum end time across predicted events.

- **Joint Model** [11]: A clip-based text question answering approach that employs a multimodal encoder followed by two linear layers to directly estimate the start and end boundaries of target moments.
- **MPGN** [201]: A moment-based model that formulates moment retrieval as a multi-choice machine reading comprehension task. It incorporates a fine-grained feature encoder together with a conditioned interaction module, achieving competitive performance.

The comparative results are summarized in Table 7.1. Our proposed method, **2DP-2MRC**, consistently outperforms all baseline models. Specifically, on the HIREST dataset, our model achieves the highest Recall@1 across both IoU thresholds (0.5 and 0.7) and sampling rates (64 and 128), surpassing the strongest baseline (MGPN) by a notable margin of up to 7.25% at the stricter IoU threshold of 0.7. These improvements verify that the proposed 2DP encoding and hierarchical cross-modal reasoning substantially enhance the model's ability to localize and interpret event boundaries in complex audiovisual scenes.

Moreover, the results clearly indicate that removing audio input leads to pronounced performance degradation across all models. For instance, when comparing the "with audio" and "without audio" settings, Recall@1@0.7 drops by 8.46% on average, with our model still maintaining superior performance compared to the best visual-only baseline. This observation underscores the critical role of audio cues in complementing visual information for accurate temporal boundary detection, especially in scenarios where visual transitions are subtle or ambiguous. It also highlights that multimodal alignment effectively captures semantic correlations between visual and auditory streams, which are crucial for robust event understanding.

Ablation Study

To systematically assess the contribution of each component within the proposed 2DP-2MRC framework, we conducted a series of ablation experiments, with results summarized in Table 7.2.

When the **AV-Encoder** is removed (Model 1), performance declines markedly, underscoring the necessity of coarse-grained feature encoding for capturing broader contextual information. Eliminating the **Pointer module** (Model 2) also leads to a notable drop in accuracy, demonstrating the importance of fine-grained interactions between the query and video clips for reliable moment localization. Similarly, removing the **2DP-Encoder** (Model 3) results in degraded performance, confirming its critical role in refining boundary detection and enhancing fine-grained alignment between video clips and the textual query.

Taken together, these findings clearly indicate that each module makes a significant contribution to overall performance. The integration of AV-Encoder, Pointer, and 2DP-Encoder effectively

Table 7.1: Experimental results. 1fps means that the model extracts one frame per second as a candidate moment.

Model	# of Video Clips	Recall@1		Recall@1	
		0.5	0.7	0.5	0.7
		With Audio		**Without Audio**	
BMT [209]	1fps	71.91	39.18	62.6	32.34
Joint [11]	1fps	73.32	32.60	70.70	20.60
Joint [11]	64	73.58	32.12	71.50	26.94
MGPN [201]	64	74.09	43.01	73.34	38.24
2DP-2MRC (ours)	**64**	**75.13**	**54.92**	**74.23**	**46.11**
Joint [11]	128	70.98	35.23	69.43	24.35
MGPN [201]	128	78.76	55.44	74.09	43.01
2DP-2MRC (ours)	**128**	**79.79**	**62.69**	**74.61**	**55.96**

operationalizes the proposed reading comprehension-inspired strategy—namely, skimming, refocusing, and integrating contextual cues—thereby enabling more precise and robust multimodal moment retrieval.

Qualitative Analysis

To further illustrate the effectiveness of the proposed 2DP-2MRC framework, we conducted a qualitative analysis, with representative results shown in Fig. 7.4. The visualization demonstrates that 2DP-2MRC is capable of accurately retrieving temporal segments that are highly consistent with the given language query.

For comparison, we also present qualitative results from Models 1, 2, and 3 introduced in the ablation study. Model 1, which excludes the coarse-grained feature encoder, exhibits limited attention to the overall video context, leading to overly narrow retrieved intervals. Likewise, Model 3, without the 2DP-Encoder module, shows difficulty in precisely determining the end boundary of the target

Table 7.2: Effectiveness of different modules in our 2DP-2MRC.

Model	Modules			Recall@1	
	AV-Encoder	Pointer	2DP-Encoder	0.5	0.7
2DP-2MRC	✓	✓	✓	**79.27**	**62.69**
1	×	✓	✓	74.09	39.38
2	✓	×	✓	74.61	55.96
3	✓	✓	×	78.76	57.00

moment, particularly in cases where visually similar segments occur between 418s and 514s. This issue is also observed in the Joint model, which lacks refined boundary modeling.

Overall, these results confirm that 2DP-2MRC effectively captures multimodal interactions through a coarse-to-fine process. This design mirrors human reading comprehension behavior—first skimming the context and then progressively refining focus—thereby enabling more accurate and robust moment retrieval.

7.4 Summary

In this chapter, we introduced **2-D**imensional **P**ointer-based **M**achine **R**eading **C**omprehension for **M**oment **R**etrieval **C**hoice (2DP-2MRC), a novel framework inspired by human reading comprehension processes. By integrating the AV-Encoder, Pointer, and 2DP-Encoder modules, the proposed model effectively captures multimodal interactions at multiple granularities, thereby enabling more precise localization of target moments. This design parallels the human strategy of first skimming a passage and query, revisiting relevant segments, and then synthesizing the information to identify the correct answer.

Extensive experiments on the HIREST dataset validate the superior performance of 2DP-2MRC over baseline models, confirming its capability in enhancing moment retrieval accuracy. As HIREST encompasses multiple tasks beyond retrieval, such as moment segmentation and video captioning, future research will extend the proposed framework to these tasks, with the goal of further advancing

Query: *Set Vegetable Dyes in Clothing.*

0s	43s	**Ground Truth**	418s	515.2s
	60s	**Joint**		514s
	32s	**MGPN**	418s	
	44s	**2DP-2MRC (ours)**	415s	
	55s	**Model 1 (w/o AV-Encoder)**	399s	
	44s	**Model 2 (w/o Pointer)**	404s	
	60s	**Model 3 (w/o 2DP-Encoder)**		514s

Figure 7.4: A qualitative example of our 2DP-2MRC and ablation models evaluated on the HIREST dataset (with 128 video clips).

multimodal video understanding.

Chapter 8

Conclusions

8.1 Summary of This Thesis

This thesis centers on automatic speech recognition (ASR) and ASR-related downstream understanding in multimodal settings. We target robust recognition and retrieval in real-world conditions that involve rare and homophonous words, overlapping speakers, long-tail domain entities, and noisy contexts. The core question is how to integrate external context, phonetic cues, and large-language-model (LLM) priors—while retaining efficiency and low latency—to mitigate ASR errors and improve downstream performance.

Our contributions span two axes. On the ASR axis, we develop three complementary lines: (i) post-hoc ASR error correction (AEC) with phoneme-augmented multimodal fusion (PMF-CEC); (ii) E2E contextual biasing with phoneme-aware entity modeling and disambiguation (PARCO); and (iii) LLM-driven contextual multi-talker ASR that unifies overlapping-speech recognition with rare-word biasing (CMT-LLM). To facilitate a concise comparison of these context-aware ASR approaches, Table 8.1 summarizes their key characteristics in terms of modeling paradigm, latency, computational complexity, language modeling capability, and recognition performance. On the downstream axis, we study two ASR-aware multimodal tasks: (iv) multimodal speech emotion recognition (SER) that leverages and repairs ASR hypotheses via multitask and contrastive/adversarial learning (M^4SER); and (v) multimodal video moment retrieval (VMR) that fuses video, audio-derived transcripts, and text queries with a two-dimensional pointer reading-comprehension mechanism (2DP-2MRC).

In this thesis, the core body of work is divided into two parts. Chapters 3, 4, and 5 are devoted to automatic speech recognition (ASR) itself, covering AEC, E2E contextual biasing, and LLM based multi-talker recognition. Chapters 6 and 7 then turn to ASR-related downstream tasks, namely multimodal SER and VMR, where the outputs of ASR are further exploited and refined for higher-level

Table 8.1: Comparison of proposed context-aware ASR methods.

Aspect	**PMF-CEC**	**PARCO**	**CMT-LLM**
Chapter	Ch. 3	Ch. 4	Ch. 5
ASR paradigm	Post-correction	E2E decoding	LLM-based decoding
Context injection stage	After decoding	During decoding	Prompt-based decoding
Target problem	1. Rare words 2. Homophones	1. Entity completeness 2. Homophones	1. Multi-talker 2. Large biasing lists
Phoneme modeling	✓	✓	×
Multi-token entity modeling	✓	✓	✓
Multi-talker support	×	×	✓
Scalability to large bias lists	Medium	Medium	High
Inference latency	Low	Medium	High
Key advantages	1. Lightweight 2. Deployable	1. Tight ASR integration 2. Strong generalization ability	1. Long-context modeling 2. Strong reasoning ability
Main limitations	Domain-specific finetuning	Training complexity	Computational cost

understanding.

In Chapter 3, we investigated AEC, where we introduced the PMF-CEC framework. This approach integrates phoneme-augmented multimodal fusion and error-specific selective decoding with a retention probability mechanism, enabling robust correction of rare words and homophones without excessive overcorrection.

In Chapter 4, we proposed PARCO, an E2E contextual ASR method. By introducing phoneme-augmented representations, contrastive entity disambiguation, and entity-level supervision, PARCO addresses the challenge of distinguishing confusable contextual terms and demonstrated significant robustness across domain-shift conditions.

In Chapter 5, we developed CMT-LLM, a contextual multi-talker ASR system that combines serialized output training (SOT) with prompt-based LLM decoding. This system introduces a two-stage rare-word filtering strategy that incorporates both coarse decoding and semantic retrieval, enabling more effective contextual biasing under overlapping speech conditions.

In Chapter 6, we explored the downstream task of multimodal SER. We proposed M^4SER, which integrates multimodal fusion of speech and ASR transcripts with auxiliary tasks for error detection and correction, as well as adversarial and label-based contrastive learning strategies. The model

demonstrated strong resilience to ASR errors and state-of-the-art performance on benchmark datasets.

In Chapter 7, we extended the scope to multimodal VMR, where ASR transcripts provide additional semantic grounding for video content. We introduced the 2DP-2MRC framework, which employs an AV-Encoder, a pointer mechanism, and a two-dimensional probability encoder to achieve fine-grained boundary detection. This model aligns with human reading-comprehension strategies and significantly outperformed prior baselines on the HIREST dataset.

Overall, the thesis demonstrates that integrating multimodal information with contextual modeling, auxiliary tasks, and reading-comprehension-inspired strategies leads to robust and efficient systems for speech recognition, emotion understanding, and video moment retrieval.

8.2 Future Work

Building upon the findings of this thesis, several future research directions can be pursued:

- **Extension to more powerful foundation models:** With the rapid evolution of LLMs and audio-language models, future work will explore integrating advanced LLM-based ASR systems to reassess the necessity of auxiliary modules such as AED and AEC.
- **Incorporation of additional modalities:** Beyond speech, text, and video, incorporating visual cues such as facial expressions and physiological signals may further enhance emotion recognition and multimodal understanding.
- **Towards real-time and low-latency systems:** Optimizing the proposed frameworks for streaming inputs and deployment on resource-constrained devices will be critical for practical applications such as dialogue systems, video retrieval platforms, and affective computing.
- **Generalization to cross-domain and low-resource scenarios:** Future studies will investigate domain adaptation and self-supervised learning techniques to improve model robustness under domain shift and limited training data conditions.
- **Applications in real-world multimodal AI systems:** Extending the proposed frameworks to real-world applications such as conversational assistants, video content management, and multimedia search engines represents a promising direction for broader impact.

In conclusion, this thesis presents methodological advances that bridge ASR, multimodal SER, and VMR. These contributions not only push forward academic research but also lay the foundation for future multimodal systems capable of robust and efficient understanding of human communication in real-world scenarios.

Acknowledgement

First and foremost, I would like to express my deepest gratitude to my doctoral supervisor, Prof. Tomoki Toda, for his continuous guidance and support throughout my Ph.D. studies, which enabled me to keep exploring in the field of speech and language technology. He devoted countless efforts to refining my research ideas, directions, and dissertation writing, while consistently encouraging and inspiring me to pursue academic excellence. I am sincerely thankful to him for granting me the opportunity to undertake this challenging yet rewarding doctoral research. His academic integrity and professionalism have set a remarkable example, leaving a profound influence on my research and future career.

I would also like to extend my heartfelt thanks to our laboratory secretaries, Ms. Noro and Ms. Hayashi, whose thoughtful administrative support greatly facilitated my research life. My gratitude also goes to all my laboratory colleagues for their friendship and for fostering an inspiring and collaborative research environment. I would like to extend special thanks to my Chinese friends in the lab—Rui Wang, Ding Ma, Shaowen Chen, Cheng-hung Hu, Shuming Luan, Xiaohan Shi, Haopeng Geng, Li Li, Chao Xie, Fengji Li, Jinyi Mi, Jingyi Feng, and Jiachen Wang—for their continuous support and encouragement. I am especially indebted to my laboratory tutor, Rui Wang, who not only offered invaluable academic advice but also provided generous care and assistance in many aspects of daily life.

During my doctoral course, I also had the privilege to conduct an internship at CyberAgent, Inc. I am deeply grateful to my company tutors, Naoki Sawada and Koichi Miyazaki, for their tremendous guidance and support in both research and career development. My heartfelt thanks also go to Li Li, my senior at the university, who kindly helped me during the internship and in daily life, making it much easier for me to adapt to a new environment.

Beyond academia, I am truly thankful to the friends I met in Japan as well as my close friends in China. In particular, I would like to thank Chen Yang, Zhu Peng, Lin Fu, Huimin Zhang, Shirong Fu, Wenbo Xie, and Liming Zhang for their companionship, understanding, and encouragement, which brought warmth and meaning to my life abroad. In times of difficulty or helplessness, their comfort and support have always been a great source of strength.

Finally, I wish to express my profound gratitude to my parents, my sister, and all my relatives and close friends. Their unwavering and selfless love and support have allowed me to pursue my ideals and academic goals without hesitation. This dissertation is not only the outcome of my own efforts but also the fruit of the collective support, help, and companionship of all those who have stood by me.

References

[1] Duc Le, Mahaveer Jain, Gil Keren, Suyoun Kim, Yangyang Shi, Jay Mahadeokar, Julian Chan, Yuan Shangguan, Christian Fuegen, Ozlem Kalinli, et al. Contextualized streaming end-to-end speech recognition with trie-based deep biasing and shallow fusion. In Proc. Interspeech, pp. 1772–1776, 2021.

[2] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In Proc. ICML, pp. 28492–28518, 2023.

[3] Ruizhe Huang, Mahsa Yarmohammadi, Sanjeev Khudanpur, and Daniel Povey. Improving neural biasing for contextual speech recognition by early context injection and text perturbation. In Proc. Interspeech, pp. 752–756, 2024.

[4] Zekun Yang, Jiajun He, and Tomoki Toda. Multi-modal video summarization based on two-stage fusion of audio, visual, and recognized text information. In Proc. APSIPA ASC, pp. 1–6, 2024.

[5] Meishan Zhang, Bin Wang, Hao Fei, and Min Zhang. In-context learning for few-shot nested named entity recognition. In Proc. ICASSP, pp. 10026–10030, 2024.

[6] Morten Kolbæk, Dong Yu, Zheng-Hua Tan, and Jesper Jensen. Multi-talker speech separation with utterance-level permutation invariant training of deep recurrent neural networks. IEEE/ACM Transactions on Audio, Speech, and Language Processing, Vol. 25, No. 10, pp. 1901–1913, 2017.

[7] Hao Shi, Yuan Gao, Zhaoheng Ni, and Tatsuya Kawahara. Serialized speech information guidance with overlapped encoding separation for multi-speaker automatic speech recognition. In Proc. SLT, pp. 1–7, 2024.

[8] Binghuai Lin and Liyuan Wang. Robust multi-modal speech emotion recognition with ASR error adaptation. In Proc. ICASSP, pp. 1–5, 2023.

[9] Jiajun He, Xiaohan Shi, Xingfeng Li, and Tomoki Toda. MF-AED-AEC: Speech emotion recognition by leveraging multimodal fusion, ASR error detection, and ASR error correction. In Proc. ICASSP, pp. 11066–11070, 2024.

[10] Lei Chen, Zhen Deng, Libo Liu, and Shibai Yin. Multilevel semantic interaction alignment for video–text cross-modal retrieval. IEEE Transactions on Circuits and Systems for Video Technology, Vol. 34, No. 7, pp. 6559–6575, 2024.

[11] Abhay Zala, Jaemin Cho, Satwik Kottur, Xilun Chen, Barlas Oguz, Yashar Mehdad, and Mohit Bansal. Hierarchical video-moment retrieval and step-captioning. In Proc. CVPR, pp. 23056–23065, 2023.

[12] Yousef O Sharrab, Hani Attar, Mohammad Ali H Eljinini, Yasmin Al-Omary, Wala'aAl-Momani. Advancements in speech recognition: A systematic review of deep learning transformer models, trends, innovations, and future directions. IEEE Access, Vol. 13, pp. 46925–46940, 2025.

[13] Kai-Fu Lee. On large-vocabulary speaker-independent continuous speech recognition. Speech communication, Vol. 7, No. 4, pp. 375–379, 1988.

[14] George E Dahl, Dong Yu, Li Deng, and Alex Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. IEEE Transactions on Audio, Speech, and Language Processing, Vol. 20, No. 1, pp. 30–42, 2011.

[15] Kanishka Rao, Haşim Sak, and Rohit Prabhavalkar. Exploring architectures, data and units for streaming end-to-end speech recognition with RNN-transducer. In Proc. ASRU, pp. 193–199, 2017.

[16] Liang Lu, Xingxing Zhang, Kyunghyun Cho, and Stephen Renals. A study of the recurrent neural network encoder-decoder for large vocabulary speech recognition. In Proc. Interspeech, pp. 3249–3253, 2015.

[17] Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. wav2vec: Unsupervised pre-training for speech recognition. In Proc. Interspeech, pp. 3465–3469, 2019.

[18] Li-Wei Chen and Alexander Rudnicky. Exploring wav2vec 2.0 fine tuning for improved speech emotion recognition. Proc. ICASSP, pp. 1–5, 2023.

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

[20] Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. WavLM: Large-scale self-supervised pre-training for full stack speech processing. IEEE Journal of Selected Topics in Signal Processing, Vol. 16, pp. 1505–1518, 2022.

[21] Ziyang Ma, Guanrou Yang, Yifan Yang, Zhifu Gao, Jiaming Wang, Zhihao Du, Fan Yu, Qian Chen, Siqi Zheng, Shiliang Zhang, et al. An embarrassingly simple approach for LLM with strong ASR capacity. In arXiv:2402.08846, pp. 1–11, 2024.

[22] Yoshua Bengio, et al. Markovian models for sequential data. Neural computing surveys, Vol. 2, No. 199, pp. 129–162, 1999.

[23] Shipra J Arora and Rishi Pal Singh. Automatic speech recognition: a review. International Journal of Computer Applications, Vol. 60, No. 9, pp. 34–44, 2012.

[24] Ken H Davis, R Biddulph, and Stephen Balashek. Automatic recognition of spoken digits. The Journal of the Acoustical Society of America, Vol. 24, No. 6, pp. 637–642, 1952.

[25] Harry F Olson and Herbert Belar. Phonetic typewriter. The Journal of the Acoustical Society of America, Vol. 28, No. 6, pp. 1072–1081, 1956.

[26] James W Forgie and Carma D Forgie. Results obtained from a vowel recognition computer program. The Journal of the Acoustical Society of America, Vol. 31, No. 11, pp. 1480–1489, 1959.

[27] Toshiyuki Sakai and Shuji Doshita. An automatic recognition system of speech sounds. 音声科学研究, Vol. 2, pp. 83–95, 1962.

[28] Biing-Hwang Juang and Lawrence R Rabiner. Automatic speech recognition–a brief history of the technology development. Georgia Institute of Technology. Atlanta Rutgers University and the University of California. Santa Barbara, Vol. 1, No. 67, p. 1, 2005.

[29] Takao Murakami, Kazutaka Maruyama, Nobuaki Minematsu, and Keikichi Hirose. Japanese vowel recognition based on structural representation of speech. In Proc. Interspeech, pp. 1261–1264, 2005.

[30] K Nagata. Spoken digit recognizer for Japanese language. NEC research & development, No. 6, 1963.

[31] Fumitada Itakura. A statistical method for estimation of speech spectral density and formant frequencies. Transaction of IEICE of Japan, Vol. 53, No. 1, pp. 36–43, 1970.

[32] Taras K Vintsyuk. Speech discrimination by dynamic programming. Cybernetics, Vol. 4, No. 1, pp. 52–57, 1968.

[33] Bishnu S Atal and Suzanne L Hanauer. Speech analysis and synthesis by linear prediction of the speech wave. The Journal of the Acoustical Society of America, Vol. 50, No. 2B, pp. 637–655, 1971.

[34] Lawrence Rabiner and Biing-Hwang Juang. Fundamentals of speech recognition. Prentice-Hall, Inc., p. 507, 1993.

[35] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In Proc. ICML, pp. 369–376, 2006.

[36] Alex Graves. Sequence transduction with recurrent neural networks. In Proc. ICML, pp. 235–242, 2012.

[37] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Proc. NeurIPS, Vol. 27, pp. 1–9, 2014.

[38] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. Attention-based models for speech recognition. In Proc. NeurIPS, pp. 577–585, 2015.

[39] Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio. End-to-end attention-based large vocabulary speech recognition. In Proc. ICASSP, pp. 4945–4949, 2016.

[40] William Chan, et al. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In Proc. ICASSP, pp. 4960–4964, 2016.

[41] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. Conformer: Convolution-augmented transformer for speech recognition. In Proc. Interspeech, pp. 5036–5040, 2020.

[42] Dong Wang, Xiaodong Wang, and Shaohe Lv. An overview of end-to-end automatic speech recognition. Symmetry, Vol. 11, No. 8, p. 1018, 2019.

[43] John R Hershey, Zhuo Chen, Jonathan Le Roux, and Shinji Watanabe. Deep clustering: Discriminative embeddings for segmentation and separation. In Proc. ICASSP, pp. 31–35, 2016.

[44] Yi Luo, Zhuo Chen, and Nima Mesgarani. Speaker-independent speech separation with deep attractor network. IEEE/ACM Transactions on Audio, Speech, and Language Processing, Vol. 26, No. 4, pp. 787–796, 2018.

[45] Dong Yu, Morten Kolbæk, Zheng-Hua Tan, and Jesper Jensen. Permutation invariant training of deep models for speaker-independent multi-talker speech separation. In Proc. ICASSP, pp. 241–245, 2017.

[46] DeLiang Wang and Jitong Chen. Supervised speech separation based on deep learning: An overview. IEEE/ACM Transactions on Audio, Speech, and Language Processing, Vol. 26, No. 10, pp. 1702–1726, 2018.

[47] Jharna Agrawal, Manish Gupta, and Hitendra Garg. A review on speech separation in cocktail party environment: challenges and approaches. Multimedia Tools and Applications, Vol. 82, No. 20, pp. 31035–31067, 2023.

[48] George Saon, Hagen Soltau, David Nahamoo, and Michael Picheny. Speaker adaptation of neural network acoustic models using i-vectors. In Proc. ASRU, pp. 55–59, 2013.

[49] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. X-vectors: Robust dnn embeddings for speaker recognition. In Proc. ICASSP, pp. 5329–5333, 2018.

[50] Georg Heigold, Ignacio Moreno, Samy Bengio, and Noam Shazeer. End-to-end text-dependent speaker verification. In Proc. ICASSP, pp. 5115–5119, 2016.

[51] Srikanth Raj Chetupalli and Sriram Ganapathy. Speaker conditioned acoustic modeling for multi-speaker conversational asr. In Proc. Interspeech, pp. 3834–3838, 2021.

[52] Dong Yu, Xuankai Chang, and Yanmin Qian. Recognizing multi-talker speech with permutation invariant training. In Proc. Interspeech, pp. 2456–2460, 2017.

[53] Sue E Tranter and Douglas A Reynolds. An overview of automatic speaker diarization systems. IEEE Transactions on Audio, Speech, and Language Processing, Vol. 14, No. 5, pp. 1557–1565, 2006.

[54] Xavier Anguera, Simon Bozonnet, Nicholas Evans, Corinne Fredouille, Gerald Friedland, and Oriol Vinyals. Speaker diarization: A review of recent research. IEEE Transactions on Audio, Speech, and Language Processing, Vol. 20, No. 2, pp. 356–370, 2012.

[55] Adam Janin, Don Baron, Jane Edwards, Dan Ellis, David Gelbart, Nelson Morgan, Barbara Peskin, Thilo Pfau, Elizabeth Shriberg, Andreas Stolcke, et al. The ICSI meeting corpus. In Proc. ICASSP, Vol. 1, pp. 1–4, 2003.

[56] Aonan Zhang, Quan Wang, Zhenyao Zhu, John Paisley, and Chong Wang. Fully supervised speaker diarization. In Proc. ICASSP, pp. 6301–6305, 2019.

[57] Naoyuki Kanda, Yashesh Gaur, Xiaofei Wang, Zhong Meng, and Takuya Yoshioka. Serialized output training for end-to-end overlapped speech recognition. In Proc. Interspeech, pp. 2797–2801, 2020.

[58] Naoyuki Kanda, Guoli Ye, Yashesh Gaur, Xiaofei Wang, Zhong Meng, Zhuo Chen, and Takuya Yoshioka. End-to-end speaker-attributed ASR with transformer. In Proc. Interspeech, pp. 4413–4417, 2021.

[59] Zhiyun Fan, Linhao Dong, Jun Zhang, Lu Lu, and Zejun Ma. SA-SOT: Speaker-aware serialized output training for multi-talker ASR. In Proc. ICASSP, pp. 9986–9990, 2024.

[60] Naoyuki Kanda, Guoli Ye, Yu Wu, Yashesh Gaur, Xiaofei Wang, Zhong Meng, Zhuo Chen, and Takuya Yoshioka. Large-scale pre-training of end-to-end multi-talker ASR for meeting transcription with single distant microphone. In Proc. Interspeech, pp. 3430–3434, 2021.

[61] Petar S Aleksic, Mohammadreza Ghodsi, Assaf Hurwitz Michaely, Cyril Allauzen, Keith B Hall, Brian Roark, David Rybach, and Pedro J Moreno. Bringing contextual information to Google speech recognition. In Proc. Interspeech, pp. 468–472, 2015.

[62] Ian McGraw, Rohit Prabhavalkar, Raziel Alvarez, Montse Gonzalez Arenas, Kanishka Rao, David Rybach, Ouais Alsharif, Haşim Sak, Alexander Gruenstein, Françoise Beaufays, et al. Personalized speech recognition on mobile devices. In Proc. ICASSP, pp. 5955–5959, 2016.

[63] Ding Zhao, Tara N Sainath, David Rybach, Pat Rondon, Deepti Bhatia, Bo Li, and Ruoming Pang. Shallow-fusion end-to-end contextual biasing. In Proc. Interspeech, pp. 1418–1422, 2019.

[64] Ian Williams, Anjuli Kannan, Petar S Aleksic, David Rybach, and Tara N Sainath. Contextual speech recognition in end-to-end neural network systems using beam search. In Interspeech, pp. 2227–2231, 2018.

[65] Jiyang Tang, Kwangyoun Kim, Suwon Shon, Felix Wu, and Prashant Sridhar. Improving ASR contextual biasing with guided attention. In Proc. ICASSP, pp. 12096–12100, 2024.

[66] Tsendsuren Munkhdalai, Zelin Wu, Golan Pundak, Khe Chai Sim, Jiayang Li, Pat Rondon, and Tara N Sainath. NAM+: Towards scalable end-to-end contextual biasing for adaptive ASR. In Proc. SLT, pp. 190–196, 2023.

[67] Xuandi Fu, Kanthashree Mysore Sathyendra, Ankur Gandhe, Jing Liu, Grant P Strimel, Ross McGowan, and Athanasios Mouchtaris. Robust acoustic and semantic contextual biasing in neural transducers for speech recognition. In Proc. ICASSP, pp. 1–5, 2023.

[68] Guangzhi Sun, Chao Zhang, and Philip C Woodland. Tree-constrained pointer generator for end-to-end contextual speech recognition. In Proc. ASRU, pp. 780–787, 2021.

[69] Guangzhi Sun, Chao Zhang, and Philip C Woodland. Tree-constrained pointer generator with graph neural network encodings for contextual speech recognition. In Proc. Interspeech, pp. 2043–2047, 2022.

[70] Guangzhi Sun, Chao Zhang, and Philip C Woodland. Graph neural networks for contextual ASR with the tree-constrained pointer generator. IEEE/ACM Transactions on Audio, Speech, and Language Processing, pp. 2407–2417, 2024.

[71] Golan Pundak, Tara N Sainath, Rohit Prabhavalkar, Anjuli Kannan, and Ding Zhao. Deep context: end-to-end contextual speech recognition. In Proc. SLT, pp. 418–425, 2018.

[72] Zhehuai Chen, Mahaveer Jain, Yongqiang Wang, Michael L Seltzer, and Christian Fuegen. Joint grapheme and phoneme embeddings for contextual end-to-end ASR. In Proc. Interspeech, pp. 3490–3494, 2019.

[73] Antoine Bruguier, Rohit Prabhavalkar, Golan Pundak, and Tara N Sainath. Phoebe: Pronunciation-aware contextualization for end-to-end speech recognition. In Proc. ICASSP, pp. 6171–6175, 2019.

[74] Feng-Ju Chang, Jing Liu, Martin Radfar, Athanasios Mouchtaris, Maurizio Omologo, Ariya Rastrow, and Siegfried Kunzmann. Context-aware transformer transducer for speech recognition. In Proc. ASRU, pp. 503–510, 2021.

[75] Kanthashree Mysore Sathyendra, Thejaswi Muniyappa, Feng-Ju Chang, Jing Liu, Jinru Su, Grant P Strimel, Athanasios Mouchtaris, and Siegfried Kunzmann. Contextual adapters for personalized speech recognition in neural transducers. In Proc. ICASSP, pp. 8537–8541, 2022.

[76] Yui Sudo, Muhammad Shakeel, Yosuke Fukumoto, Yifan Peng, and Shinji Watanabe. Contextualized automatic speech recognition with attention-based bias phrase boosted beam search. In Proc. ICASSP, pp. 10896–10900, 2024.

[77] Fan Yu, Haoxu Wang, Xian Shi, and Shiliang Zhang. LCB-Net: Long-context biasing for audio-visual speech recognition. In Proc. ICASSP, pp. 10621–10625, 2024.

[78] Xiaoqiang Wang, Yanqing Liu, Sheng Zhao, and Jinyu Li. A light-weight contextual spelling correction model for customizing transducer-based speech recognition systems. In Proc. Interspeech, pp. 1982–1986, 2021.

[79] Jin Jiang, Xunjian Yin, Xiaojun Wan, Wei Peng, Rongjun Li, Jingyuan Yang, and Yanquan Zhou. Contextual modeling for document-level ASR error correction. In Proc. LREC-COLING, pp. 3855–3867, 2024.

[80] Xiaoqiang Wang, Yanqing Liu, Jinyu Li, Veljko Miljanic, Sheng Zhao, and Hosam Khalil. Towards contextual spelling correction for customization of end-to-end speech recognition systems. IEEE/ACM Transactions on Audio, Speech, and Language Processing, Vol. 30, pp. 3089–3097, 2022.

[81] Ling Dong, Wenjun Wang, Zhengtao Yu, Yuxin Huang, Junjun Guo, and Guojiang Zhou. Pronunciation guided copy and correction model for ASR error correction. International Journal of Machine Learning and Cybernetics, pp. 1–13, 2024.

[82] Ye Bai, Jingping Chen, Jitong Chen, Wei Chen, Zhuo Chen, Chuang Ding, Linhao Dong, Qianqian Dong, Yujiao Du, Kepan Gao, et al. Seed-ASR: Understanding diverse speech and contexts with LLM-based speech recognition. In arXiv:2407.04675, pp. 1–20, 2024.

[83] Guanrou Yang, Ziyang Ma, Fan Yu, Zhifu Gao, Shiliang Zhang, and Xie Chen. MaLa-ASR: Multimedia-assisted LLM-based ASR. In Proc. Interspeech, pp. 2405–2409, 2024.

[84] Chen Chen, Yuchen Hu, Chao-Han Huck Yang, Sabato Marco Siniscalchi, Pin-Yu Chen, and Eng-Siong Chng. Hyporadise: An open baseline for generative speech recognition with large language models. In Proc. NeurIPS, pp. 31665–31688, 2023.

[85] Chao-Han Huck Yang, Yile Gu, Yi-Chieh Liu, Shalini Ghosh, Ivan Bulyko, and Andreas Stolcke. Generative speech recognition error correction with large language models and task-activating prompting. In Proc. ASRU, pp. 1–8, 2023.

[86] Chao-Han Huck Yang, Taejin Park, Yuan Gong, Yuanchao Li, Zhehuai Chen, Yen-Ting Lin, Chen Chen, Yuchen Hu, Kunal Dhawan, Piotr Żelasko, et al. Large language model based generative error correction: A challenge and baselines for speech recognition, speaker tagging, and emotion recognition. In Proc. SLT, pp. 371–378, 2024.

[87] Yuchen Hu, Chen Chen, Chao-Han Huck Yang, Ruizhe Li, Chao Zhang, Pin-Yu Chen, and En-Siong Chng. Large language models are efficient learners of noise-robust speech recognition. In Proc. ICLR, pp. 1–25, 2024.

[88] Sreyan Ghosh, Utkarsh Tyagi, S Ramaneswaran, Harshvardhan Srivastava, and Dinesh Manocha. MMER: Multimodal multi-task learning for speech emotion recognition. In Proc. Interspeech, pp. 1209–1213, 2023.

[89] Weidong Chen, Xiaofen Xing, Xiangmin Xu, Jianxin Pang, and Lan Du. SpeechFormer: A hierarchical efficient framework incorporating the characteristics of speech. Proc. Interspeech, pp. 346–350, 2022.

[90] Weiquan Fan, Xiangmin Xu, Bolun Cai, and Xiaofen Xing. ISNet: Individual standardization network for speech emotion recognition. IEEE/ACM Transactions on Audio, Speech, and Language Processing, Vol. 30, pp. 1803–1814, 2022.

[91] Xiaohan Shi, Yuan Gao, Jiajun He, Jinyi Mi, Xingfeng Li, and Tomoki Toda. A study on multimodal fusion and layer adapter in emotion recognition. In Proc. APSIPA ASC, pp. 1–6, 2024.

[92] Jinyi Mi, Xiaohan Shi, Ding Ma, Jiajun He, Takuya Fujimura, and Tomoki Toda. Two-stage framework for robust speech emotion recognition using target speaker extraction in human speech noise conditions. In Proc APSIPA ASC, pp. 1–6, 2024.

[93] Qifei Li, Yingming Gao, Yuhua Wen, Ziping Zhao, Ya Li, and Björn W Schuller. SeeNet: A soft emotion expert and data augmentation method to enhance speech emotion recognition. IEEE Transactions on Affective Computing, Vol. 16, No. 3, pp. 2142–2156, 2025.

[94] Santosh Kumar Bharti, S Varadhaganapathy, Rajeev Kumar Gupta, Prashant Kumar Shukla, Mohamed Bouye, Simon Karanja Hingaa, and Amena Mahmoud. Text-based emotion recognition using deep learning approach. Computational Intelligence and Neuroscience, Vol. 2022, No. 1, pp. 1–8, 2022.

[95] Weiquan Fan, Xiaofen Xing, Bolun Cai, and Xiangmin Xu. MGAT: Multi-granularity attention based transformers for multi-modal emotion recognition. In Proc. ICASSP, pp. 1–5, 2023.

[96] Haoqin Sun, Shiwan Zhao, Xuechen Wang, Wenjia Zeng, Yong Chen, and Yong Qin. Fine-grained disentangled representation learning for multimodal emotion recognition. In Proc. ICASSP, pp. 11051–11055, 2024.

[97] Weiquan Fan, Xiangmin Xu, Guohua Zhou, Xiaofang Deng, and Xiaofen Xing. Coordination attention based transformers with bidirectional contrastive loss for multimodal speech emotion recognition. Speech Communication, pp. 1–10, 2025.

[98] Jiajun He, Jinyi Mi, and Tomoki Toda. GIA-MIC: Multimodal emotion recognition with gated interactive attention and modality-invariant learning constraints. In Proc. Interspeech, pp. 1–5, 2025.

[99] Jennifer Santoso, Takeshi Yamada, Shoji Makino, Kenkichi Ishizuka, and Takekatsu Hiramura. Speech emotion recognition based on attention weight correction using word-level confidence measure. In Proc. Interspeech, pp. 1947–1951, 2021.

[100] Yuanchao Li, Peter Bell, and Catherine Lai. Speech emotion recognition with ASR transcripts: A comprehensive study on word error rate and fusion techniques. In Proc. SLT, pp. 518–525, 2024.

[101] Yuanchao Li, Pinzhen Chen, Peter Bell, and Catherine Lai. Crossmodal ASR error correction with discrete speech units. In Proc. SLT, pp. 431–438, 2024.

[102] Yuanchao Li, Yuan Gong, Chao-Han Huck Yang, Peter Bell, and Catherine Lai. Revise, reason, and recognize: LLM-based emotion recognition via emotion-specific prompts and ASR error correction. In Proc. ICASSP, pp. 1–5, 2025.

[103] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In Proc. ICCV, pp. 5803–5812, 2017.

[104] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal activity localization via language query. In Proc. ICCV, pp. 5267–5275, 2017.

[105] Meng Liu, Liqiang Nie, Yunxiao Wang, Meng Wang, and Yong Rui. A survey on video moment localization. ACM Computing Surveys, Vol. 55, No. 9, pp. 1–37, 2023.

[106] Bolin Zhang, Bin Jiang, Chao Yang, and Liang Pang. Dual-channel localization networks for moment retrieval with natural language. In Proc. ICMR, pp. 351–359, 2022.

[107] Xin Sun, Jialin Gao, Yizhe Zhu, Xuan Wang, and Xi Zhou. Video moment retrieval via comprehensive relation-aware network. IEEE Transactions on Circuits and Systems for Video Technology, Vol. 33, No. 9, pp. 5281–5295, 2023.

[108] Shoubin Yu, Jaemin Cho, Prateek Yadav, and Mohit Bansal. Self-chained image-language model for video localization and question answering. Proc. NeurIPS, Vol. 36, pp. 76749–76771, 2023.

[109] Shuwei Huo, Yuan Zhou, Ruolin Wang, Wei Xiang, and Sun-Yuan Kung. Semantic relevance learning for video-query based video moment retrieval. IEEE Transactions on Multimedia, Vol. 25, pp. 9290–9301, 2023.

[110] Ziji Zhang, Zhehui Wang, Rajesh Kamma, Sharanya Eswaran, and Narayanan Sadagopan. PATCorrect: Non-autoregressive phoneme-augmented transformer for ASR error correction. In Proc. Interspeech, pp. 3904–3908, 2023.

[111] Jiajun He, Zekun Yang, and Tomoki Toda. ED-CEC: Improving rare word recognition using ASR postprocessing based on error detection and context-aware error correction. In Proc. ASRU, pp. 1–6, 2023.

[112] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv:1609.08144, pp. 1–23, 2016.

[113] Jiajun He, Zekun Yang, and Tomoki Toda. ED-CEC: Improving rare word recognition using ASR postprocessing based on error detection and context-aware error correction. In Proc. ASRU, pp. 1–6, 2023.

[114] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proc. NAACL-HLT, pp. 4171–4186, 2019.

[115] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Proc. NeurIPS, Vol. 30, pp. 6000–6010, 2017.

[116] Linh The Nguyen, Thinh Pham, and Dat Quoc Nguyen. XPhoneBERT: A pre-trained multilingual model for phoneme representations for text-to-speech. In Proc. Interspeech, pp. 5506–5510, 2023.

[117] Diederik P Kingma and Jimmy Ba. ADAM: A method for stochastic optimization. In Proc. ICLR, pp. 1–15, 2015.

[118] Mukuntha Narayanan Sundararaman, Ayush Kumar, and Jithendra Vepa. Phoneme-BERT: Joint language modelling of phoneme sequence and ASR transcript. In Proc. Interspeech, pp. 3236–3240, 2021.

[119] Chao-Wei Huang and Yun-Nung Chen. Learning ASR-robust contextualized embeddings for spoken language understanding. In Proc. ICASSP, pp. 8009–8013, 2020.

[120] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. The kaldi speech recognition toolkit. In Proc. ASRU, Vol. 1, pp. 1–5, 2011.

[121] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. LibriSpeech: An ASR corpus based on public domain audio books. In Proc. ICASSP, pp. 5206–5210, 2015.

[122] Zhuoyuan Yao, Di Wu, Xiong Wang, Binbin Zhang, Fan Yu, Chao Yang, Zhendong Peng, Xiaoyu Chen, Lei Xie, and Xin Lei. WeNet: Production oriented streaming and non-streaming end-to-end speech recognition toolkit. In Proc. Interspeech, pp. 4054–4058, 2021.

[123] Hemant Yadav, Sreyan Ghosh, Yi Yu, and Rajiv Ratn Shah. End-to-end named entity recognition from english speech. In Proc. Interspeech, pp. 4268–4272, 2020.

[124] Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, et al. ESPnet: End-to-end speech processing toolkit. In Proc. Interspeech, pp. 2207–2211, 2018.

[125] Abner Hernandez and Seung Hee Yang. Multimodal corpus analysis of autoblog 2020: lecture videos in machine learning. In Proc. SPECOM, pp. 262–270, 2021.

[126] Mirco Ravanelli, Titouan Parcollet, Peter Plantinga, Aku Rouhe, Samuele Cornell, Loren Lugosch, Cem Subakan, Nauman Dawalatabad, Abdelwahab Heba, Jianyuan Zhong, et al. SpeechBrain: A general-purpose speech toolkit. arXiv:2106.04624, pp. 1–34, 2021.

[127] Yun Zhao, Xuerui Yang, Jinchao Wang, Yongyu Gao, Chao Yan, and Yuanfu Zhou. BART based semantic correction for Mandarin automatic speech recognition system. In Proc. Interspeech, pp. 2017–2021, 2021.

[128] Sam Shleifer and Alexander M Rush. Pre-trained summarization distillation. arXiv:2010.13002, pp. 1–10, 2020.

[129] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. OpenAI blog, Vol. 1, No. 8, p. 9, 2019.

[130] Srijith Radhakrishnan, Chao-Han Huck Yang, Sumeer Ahmad Khan, Rohit Kumar, Narsis A Kiani, David Gomez-Cabrero, and Jesper N Tegner. Whispering LLaMa: A cross-modal generative error correction framework for speech recognition. In Proc. EMNLP, pp. 10007–10016, 2023.

[131] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. LLaMa: Open and efficient foundation language models. In arXiv:2302.13971, pp. 1–16, 2023.

[132] Suyoun Kim, Takaaki Hori, and Shinji Watanabe. Joint CTC-attention based end-to-end speech recognition using multi-task learning. In Proc. ICASSP, pp. 4835–4839, 2017.

[133] Takafumi Moriya, Tsubasa Ochiai, Shigeki Karita, Hiroshi Sato, Tomohiro Tanaka, Takanori Ashihara, Ryo Masumura, Yusuke Shinohara, and Marc Delcroix. Self-distillation for improving CTC-transformer-based ASR systems. In Proc. Interspeech, pp. 546–550, 2020.

[134] Christian Huber, Juan Hussain, Sebastian Stüker, and Alexander Waibel. Instant one-shot word-learning for context-specific neural sequence-to-sequence speech recognition. In Proc. ASRU, pp. 1–7, 2021.

[135] Hui Bu, Jiayu Du, Xingyu Na, Bengu Wu, and Hao Zheng. AISHELL-1: An open-source mandarin speech corpus and a speech recognition baseline. In Proc. O-COCOSDA, pp. 1–5, 2017.

[136] Dong Wang and Xuewei Zhang. THCHS-30: A free Chinese speech corpus. arXiv:1512.01882, pp. 1–6, 2015.

[137] Minglun Han, Linhao Dong, Shiyu Zhou, and Bo Xu. CIF-based collaborative decoding for end-to-end contextual speech recognition. In Proc. ICASSP, pp. 6528–6532, 2021.

[138] Shilin Zhou, Zhenghua Li, Yu Hong, Min Zhang, Zhefeng Wang, and Baoxing Huai. CopyNE: Better contextual ASR by copying named entities. In Proc. ACL, pp. 2675–2686, 2024.

[139] Zhengyi Zhang and Pan Zhou. End-to-end contextual ASR based on posterior distribution adaptation for hybrid CTC/attention system. arXiv:2202.09003, pp. 1–5, 2022.

[140] Boli Chen, Guangwei Xu, Xiaobin Wang, Pengjun Xie, Meishan Zhang, and Fei Huang. AISHELL-NER: Named entity recognition from Chinese speech. In Proc. ICASSP, pp. 8352–8356, 2022.

[141] Liang Lu, Naoyuki Kanda, Jinyu Li, and Yifan Gong. Streaming end-to-end multi-talker speech recognition. IEEE Signal Processing Letters, Vol. 28, pp. 803–807, 2021.

[142] Jiajun He, Zekun Yang, and Tomoki Toda. Enhancing recognition of rare words in ASR through error detection and context-aware error correction. IEICE Technical Report, Vol. 123, No. 292, pp. 13–18, 2023.

[143] Duc Le, Gil Keren, Julian Chan, Jay Mahadeokar, Christian Fuegen, and Michael L Seltzer. Deep shallow fusion for RNN-T personalization. In Proc. SLT, pp. 251–257, 2021.

[144] Guangzhi Sun, Chao Zhang, and Philip C. Woodland. Graph neural networks for contextual ASR with the tree-constrained pointer generator. IEEE/ACM Transcations Audio, Speech and Language Processing, Vol. 32, pp. 2407–2417, 2024.

[145] Kaixun Huang, Ao Zhang, Zhanheng Yang, Pengcheng Guo, Bingshen Mu, Tianyi Xu, and Lei Xie. Contextualized end-to-end speech recognition with contextual phrase prediction network. In Proc. Interspeech, pp. 4933–4937, 2023.

[146] Jiajun He and Tomoki Toda. PMF-CEC: Phoneme-augmented multimodal fusion for context-aware asr error correction with error-specific selective decoding. IEEE Transactions on Audio, Speech and Language Processing, Vol. 33, pp. 2402–2417, 2025.

[147] Lingwei Meng, Shujie Hu, Jiawen Kang, Zhaoqing Li, Yuejiao Wang, Wenxuan Wu, Xixin Wu, Xunying Liu, and Helen Meng. Large language model can transcribe speech in multi-talker scenarios with versatile instructions. In Proc. ICASSP, pp. 1–5, 2025.

[148] Mohan Shi, Zengrui Jin, Yaoxun Xu, Yong Xu, Shi-Xiong Zhang, Kun Wei, Yiwen Shao, Chunlei Zhang, and Dong Yu. Advancing multi-talker ASR performance with large language models. In Proc. SLT, pp. 14–21, 2024.

[149] Guanrou Yang, Ziyang Ma, Fan Yu, Zhifu Gao, Shiliang Zhang, and Xie Chen. MaLa-ASR: Multimedia-assisted LLM-based ASR. In Proc. Interspeech, pp. 2405–2409, 2024.

[150] Shujie Hu, Long Zhou, Shujie Liu, Sanyuan Chen, Lingwei Meng, Hongkun Hao, Jing Pan, Xunying Liu, Jinyu Li, Sunit Sivasankaran, Linquan Liu, and Furu Wei. WavLLM: Towards robust and adaptive speech large language model. In Proc. EMNLP, pp. 4552–4572, 2024.

[151] Zhanheng Yang, Sining Sun, Xiong Wang, Yike Zhang, Long Ma, and Lei Xie. Two stage contextual word filtering for context bias in unified streaming and non-streaming transducer. In Proc. Interspeech, pp. 3257–3261, 2023.

[152] Kaixun Huang, Ao Zhang, Zhanheng Yang, Pengcheng Guo, Bingshen Mu, Tianyi Xu, and Lei Xie. Contextualized end-to-end speech recognition with contextual phrase prediction network. In Proc. Interspeech, pp. 4933–4937, 2023.

[153] Tianyi Xu, Zhanheng Yang, Kaixun Huang, Pengcheng Guo, Ao Zhang, Biao Li, Changru Chen, Chao Li, and Lei Xie. Adaptive contextual biasing for transducer based streaming speech recognition. In Proc. Interspeech, pp. 1668–1672, 2023.

[154] Guanrou Yang, Ziyang Ma, Zhifu Gao, Shiliang Zhang, and Xie Chen. CTC-assisted LLM-based contextual ASR. In Proc. SLT, pp. 126–131, 2024.

[155] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. See https://vicuna. lmsys. org (accessed 14 April 2023), Vol. 2, No. 3, p. 6, 2023.

[156] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In Proc. ICLR, pp. 1–8, 2017.

[157] Pengcheng Guo, Xuankai Chang, Shinji Watanabe, and Lei Xie. Multi-speaker ASR combining non-autoregressive conformer CTC and conditional speaker chain. In Proc. Interspeech, pp. 3720–3724, 2021.

[158] Desh Raj, Daniel Povey, and Sanjeev Khudanpur. SURT 2.0: Advances in transducer-based multi-talker speech recognition. IEEE/ACM Transcations Audio, Speech and Language Processing, Vol. 31, p. 3800–3813, 2023.

[159] Joris Cosentino, Manuel Pariente, Samuele Cornell, Antoine Deleforge, and Emmanuel Vincent. Librimix: An open-source dataset for generalizable speech separation. arXiv:2005.11262, pp. 1–5, 2020.

[160] Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, et al. The AMI meeting corpus: A pre-announcement. In International Workshop on Machine Learning for Multimodal Interaction, pp. 28–39. Springer, 2005.

[161] Gordon Wichern, Joe Antognini, Michael Flynn, Licheng Richard Zhu, Emmett McQuinn, Dwight Crow, Ethan Manilow, and Jonathan Le Roux. Wham!: Extending speech separation to noisy environments. In Proc. Interspeech, pp. 1368–1372, 2019.

[162] Xavier Anguera, Chuck Wooters, and Javier Hernando. Acoustic beamforming for speaker diarization of meetings. IEEE/ACM Transcations Audio, Speech and Language Processing, Vol. 15, No. 7, pp. 2011–2022, 2007.

[163] AV Geetha, T Mala, D Priyanka, and E Uma. Multimodal emotion recognition with deep learning: advancements, challenges, and future directions. Information Fusion, Vol. 105, pp. 102218–102256, 2024.

[164] Jingguang Tian, Desheng Hu, Xiaohan Shi, Jiajun He, Xingfeng Li, Yuan Gao, Tomoki Toda, Xinkang Xu, and Xinhui Hu. Semi-supervised multimodal emotion recognition with consensus decision-making and label correction. In Proc. MRAC, pp. 67–73, 2023.

[165] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: a robustly optimized BERT pretraining approach. arXiv:1907.11692, pp. 1–13, 2019.

[166] J. Santoso, T. Yamada, et al. Speech emotion recognition based on self-attention weight correction for acoustic and text features. IEEE Access, Vol. 10, pp. 115732–115743, 2022.

[167] Dingkang Yang, Shuai Huang, Haopeng Kuang, Yangtao Du, and Lihua Zhang. Disentangled representation learning for multimodal emotion recognition. In Proc. ACMMM, pp. 1642–1651, 2022.

[168] Yao-Hung Hubert Tsai, Shaojie Bai, Paul Pu Liang, J Zico Kolter, Louis-Philippe Morency, and Ruslan Salakhutdinov. Multimodal transformer for unaligned multimodal language sequences. In Proc. ACL, pp. 6558–6569, 2019.

[169] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. arXiv:1607.06450, pp. 1–14, 2016.

[170] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proc. ICCV, pp. 1026–1034, 2015.

[171] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Proc. NeurIPS, pp. 1–9, 2014.

[172] Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N Chang, Sungbok Lee, and Shrikanth S Narayanan. IEMOCAP: Interactive emotional dyadic motion capture database. Language Resources and Evaluation, pp. 335–359, 2008.

[173] Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. MELD: A multimodal multi-party dataset for emotion recognition in conversations. In Proc. ACL, pp. 527–536, 2019.

[174] Soumya Dutta and Sriram Ganapathy. Multimodal transformer with learnable frontend and self attention for emotion recognition. In Proc. ICASSP, pp. 6917–6921, 2022.

[175] Mixiao Hou, Zheng Zhang, Chang Liu, and Guangming Lu. Semantic alignment network for multi-modal emotion recognition. IEEE Transactions on Circuits and Systems for Video Technology, pp. 5318–5329, 2023.

[176] Yusong Wang, Dongyuan Li, and Jialun Shen. Inter-modality and intra-sample alignment for multi-modal emotion recognition. In Proc. ICASSP, pp. 8301–8305, 2024.

[177] Yanfeng Wu, Pengcheng Yue, Leyuan Qu, Taihao Li, and Yu-Ping Ruan. Multi-modal emotion recognition using multiple acoustic features and dual cross-modal transformer. In Proc. ICASSP, pp. 10496–10500, 2024.

[178] Dingkang Yang, Shuai Huang, Yang Liu, and Lihua Zhang. Contextual and cross-modal interaction for multi-modal speech emotion recognition. IEEE Signal Processing Letters, Vol. 29, pp. 2093–2097, 2022.

[179] Soumya Dutta and Sriram Ganapathy. HCAM–hierarchical cross attention model for multi-modal emotion recognition. arXiv:2304.06910, pp. 1–11, 2023.

[180] Jintao Wen, Dazhi Jiang, Geng Tu, Cheng Liu, and Erik Cambria. Dynamic interactive multi-view memory network for emotion recognition in conversation. Information Fusion, Vol. 91, pp. 123–133, 2023.

[181] Shiqing Zhang, Yijiao Yang, Chen Chen, Ruixin Liu, Xin Tao, Wenping Guo, Yicheng Xu, and Xiaoming Zhao. Multimodal emotion recognition based on audio and text by using hybrid attention networks. Biomedical Signal Processing and Control, Vol. 85, No. 105052, pp. 1–10, 2023.

[182] Peizhu Gong, Jin Liu, Zhongdai Wu, Bing Han, Y Ken Wang, and Huihua He. A multi-level circulant cross-modal transformer for multimodal speech emotion recognition. Computers, Materials & Continua, Vol. 74, No. 2, pp. 4203–4220, 2023.

[183] Qifei Li, Yingming Gao, Yuhua Wen, Cong Wang, and Ya Li. Enhancing modal fusion by alignment and label matching for multimodal emotion recognition. In Proc. Interspeech, pp. 4663–4667, 2024.

[184] Jiang Li, Xiaoping Wang, Yingjian Liu, and Zhigang Zeng. CFN-ESA: A cross-modal fusion network with emotion-shift awareness for dialogue emotion recognition. IEEE Transactions on Affective Computing, Vol. 15, No. 4, pp. 1919–1933, 2024.

[185] Yichong Leng, Xu Tan, Linchen Zhu, Jin Xu, Renqian Luo, Linquan Liu, Tao Qin, Xiangyang Li, Edward Lin, and Tie-Yan Liu. Fastcorrect: Fast error correction with edit alignment for automatic speech recognition. In Proc. NeurIPS, pp. 21708–21719, 2021.

[186] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. Journal of Machine Learning Research, Vol. 9, No. 11, pp. 2579–2605, 2008.

[187] Siddique Latif, Rajib Rana, Sara Khalifa, Raja Jurdak, and Björn W Schuller. Multitask learning from augmented auxiliary data for improving speech emotion recognition. IEEE Transactions on Affective Computing, Vol. 14, No. 4, pp. 3164–3176, 2022.

[188] Siddique Latif, Muhammad Usama, Muhammad Ibrahim Malik, and Björn W Schuller. Can large language models aid in annotating speech emotional data? uncovering new frontiers [research frontier]. IEEE Computational Intelligence Magazine, Vol. 20, No. 1, pp. 66–77, 2025.

[189] Abdul Rehman, Zhen-Tao Liu, Min Wu, Wei-Hua Cao, and Cheng-Shan Jiang. Speech emotion recognition based on syllable-level feature extraction. Applied Acoustics, Vol. 211, pp. 109444–109456, 2023.

[190] Sihan Chen, Handong Li, Qunbo Wang, Zijia Zhao, Mingzhen Sun, Xinxin Zhu, and Jing Liu. Vast: A vision-audio-subtitle-text omni-modality foundation model and dataset. In Proc. NeurIPS, Vol. 36, pp. 72842–72866, 2024.

[191] Shuwei Huo, Yuan Zhou, Wei Xiang, and Sun-Yuan Kung. Weakly-supervised content-based video moment retrieval using low-rank video representation. Knowledge-Based Systems, Vol. 277, p. 110776, 2023.

[192] Henghao Zhao, Kevin Qinghong Lin, Rui Yan, and Zechao Li. DiffusionVMR: Diffusion model for joint video moment retrieval and highlight detection. IEEE Transactions on Neural Networks and Learning Systems, Vol. 36, No. 8, pp. 14522–14535, 2025.

[193] Dezhao Luo, Jiabo Huang, Shaogang Gong, Hailin Jin, and Yang Liu. Zero-shot video moment retrieval from frozen vision-language models. In Proc. WACV, pp. 5464–5473, 2024.

[194] Tongbao Chen, Wenmin Wang, Zhe Jiang, Ruochen Li, and Bingshu Wang. Cross-modality knowledge calibration network for video corpus moment retrieval. IEEE Transactions on Multimedia, pp. 3799–3813, 2023.

[195] Bolin Zhang, Chao Yang, Bin Jiang, and Xiaokang Zhou. Video moment retrieval with hierarchical contrastive learning. In Proc. ACMMM, pp. 346–355, 2022.

[196] Pandeng Li, Chen-Wei Xie, Hongtao Xie, Liming Zhao, Lei Zhang, Yun Zheng, Deli Zhao, and Yongdong Zhang. MomentDiff: Generative video moment retrieval from random to real. Proc. NeurIPS, Vol. 36, pp. 65948–65966, 2024.

[197] Yi Wang, Kun Li, Guoliang Chen, Yan Zhang, Dan Guo, and Meng Wang. Spatiotemporal contrastive modeling for video moment retrieval. World Wide Web, Vol. 26, No. 4, pp. 1525–1544, 2023.

[198] Zhifang Tan, Fei Dong, Xinfang Liu, Chenglong Li, and Xiushan Nie. Vmlh: Efficient video moment location via hashing. Electronics, Vol. 12, No. 2, p. 420, 2023.

[199] Zhuosheng Zhang, Junjie Yang, and Hai Zhao. Retrospective reader for machine reading comprehension. In Proc. AAAI, pp. 14506–14514, 2021.

[200] Yukun Zheng, Jiaxin Mao, Yiqun Liu, Zixin Ye, Min Zhang, and Shaoping Ma. Human behavior inspired machine reading comprehension. In Proc. SIGIR, pp. 425–434, 2019.

[201] Xin Sun, Xuan Wang, Jialin Gao, Qiong Liu, and Xi Zhou. You need to read again: Multi-granularity perception network for moment retrieval in videos. In Proc. SIGIR, pp. 1022–1032, 2022.

[202] Xiaobo Jiang, Kun He, Jiajun He, and Guangyu Yan. A new entity extraction method based on machine reading comprehension. arXiv:2108.06444, pp. 1–12, 2021.

[203] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. In Proc. CVPR, pp. 19358–19369, 2023.

[204] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In Proc. ICML, pp. 28492–28518, 2023.

[205] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In Proc. EMNLP, pp. 3982–2992, 2019.

[206] Serkan Kiranyaz, Onur Avci, Osama Abdeljaber, Turker Ince, Moncef Gabbouj, and Daniel J Inman. 1D convolutional neural networks and applications: A survey. Mechanical Systems and Signal Processing, Vol. 151, p. 107398, 2021.

[207] R. Jozefowicz, W. Zaremba, and I. Sutskever. An empirical exploration of recurrent network architectures. In Proc. ICML, pp. 2342–2350, 2015.

[208] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GELUs). arXiv:1606.08415, pp. 1–10, 2016.

[209] Vladimir Iashin and Esa Rahtu. A better use of audio-visual cues: Dense video captioning with bi-modal transformer. In Proc. BMVC, pp. 1–16, 2020.

List of Publications

Journal Papers

[1] **J. He** and T. Toda, "PMF-CEC: Phoneme-augmented multimodal fusion for context-aware asr error correction with error-specific selective decoding," IEEE Transactions on Audio, Speech and Language Processing, vol. 33, pp. 2402-4173, 2025.

[2] **J. He**, X. Shi, C.-H. Hu, J. Mi, X. Li, and T. Toda, "M^{4}SER: Multimodal, multirepresentation, multitask, and multistrategy learning for speech emotion recognition," IEEE Transactions on Audio, Speech and Language Processing, vol. 33, pp. 4055-4070, 2025.

[3] X. Shi, **J. He**, X. Li, and T. Toda, "A Comprehensive Study on the Effectiveness of ASR Representations for Noise-Robust Speech Emotion Recognition," IEEE Transactions on Audio, Speech and Language Processing, pp. 1-16, 2026 (Early Access).

International Conference Papers

[1] **J. He**, Z. Yang, and T. Toda, "ED-CEC: Improving rare word recognition using ASR postprocessing based on error detection and context-aware error correction," in Proc. ASRU, 2023, pp. 1–6.

[2] **J. He**, X. Shi, X. Li, and T. Toda, "MF-AED-AEC: Speech emotion recognition by leveraging multimodal fusion, ASR error detection, and ASR error correction," in Proc. ICASSP, 2024, pp. 11 066–11 070.

[3] **J. He** and T. Toda, "2DP-2MRC: 2-dimensional pointer-based machine reading comprehension method for multimodal moment retrieval," in Proc. Interspeech, 2024, pp. 5073–5077.

[4] **J. He**, N. Sawada, K. Miyazaki, and T. Toda, "CMT-LLM: Contextual multi-Talker ASR utilizing large language models," in Proc. Interspeech, 2025, pp. 2575–2579.

[5] **J. He**, J. Mi, and T. Toda, "GIA-MIC: Multimodal emotion recognition with gated interactive attention and modality-invariant learning constraints," in Proc. Interspeech, 2025, pp. 2695–2699.

[6] **J. He**, N. Sawada, K. Miyazaki, and T. Toda, "PARCO: Phoneme-augmented robust contextual ASR via contrastive entity disambiguation," in Proc. ASRU, 2025, pp. 1–7.

[7] J. Tian, D. Hu, X. Shi, **J. He**, X. Li, Y. Gao, T. Toda, X. Xu, and X. Hu, "Semi-supervised multimodal emotion recognition with consensus decision-making and label correction," in Proc. MRAC, 2023, pp. 67–73.

[8] X. Shi, Y. Gao, **J. He**, J. Mi, X. Li, and T. Toda, "A study on multimodal fusion and layer adapter in emotion recognition," in Proc. APSIPA ASC, 2024, pp. 1–6.

[9] Z. Yang, **J. He**, and T. Toda, "Multi-modal video summarization based on two-stage fusion of audio, visual, and recognized text information," in Proc APSIPA ASC, 2024, pp. 1–6.

[10] J. Mi, X. Shi, D. Ma, **J. He**, T. Fujimura, and T. Toda, "Two- stage framework for robust speech emotion recognition using tar- get speaker extraction in human speech noise conditions," in Proc. APSIPA ASC, 2024, pp. 1–6.

Technical Report

[1] **J. He**, Z. Yang, and T. Toda, "Enhancing Recognition of Rare Words in ASR through Error Detection and Context-Aware Error Correction," IEICE Tech. Rep., vol. 123, no. 292, pp. 13–18, 2023.

Awards

[1] IEEE Nagoya Section Conference Presentation Award, for the paper "ED-CEC: Improving Rare Word Recognition Using ASR Post-Processing Based on Error Detection and Context-Aware Error Correction", presented at IEEE ASRU 2023, April 2024.

[2] Shortlisted for ISCA Best Student Paper Award 2024, INTERSPEECH 2024, August 2024.