Token-Level Edge–Cloud Speculative Decoding for Real-Time Speech Emotion Captioning

Jiajun Lu
Student ID: 1654675

Abstract

Real-time Speech Emotion Captioning (SEC) faces a significant challenge in balancing the high quality of cloud-based Large Language Models (LLMs) against the strict privacy and resource constraints of edge devices. This report proposes and evaluates a token-level edge-cloud collaborative framework using speculative decoding to address this trade-off. Our methodology uses an on-device 3B SLM (Qwen-2.5-Omni-3B) to ”draft” captions and a powerful 7B cloud LLM (Qwen-2.5-Omni-7B) to ”verify” them. Our design ensures privacy by keeping the user's raw audio on the device, while abstracted audio features and drafted token IDs are transmitted for verification only when local uncertainty (measured by entropy) is high.

We evaluated this framework against edge-only and cloud-only baselines using the MER2024 dataset, in a hardware-emulated environment (a 2-core CPU for the edge vs. an A100 GPU for the cloud). Our results show that this hybrid method successfully improves caption quality over the edge-only baseline (e.g., 8.64 vs. 7.72 in BLEU-1). This proves the concept is viable. However, this quality gain is achieved at a significant cost: the total latency was higher than the edge baseline (43.87s vs. 34.89s), and the method consumed maximum resources on both the edge (CPU) and cloud (GPU). We conclude that while the hybrid framework is conceptually sound, the current implementation, which relies on a simple entropy-based gate, presents an inefficient ”quality-for-latency” trade-off. Future work must therefore focus on developing smarter gating mechanisms to identify high-impact tokens and optimize this critical balance.

1 Introduction

In recent years, the rapid advancement of Large Language Models (LLMs) has fundamentally changed our ability to process and understand complex human data. This progress is not limited to text; these models show remarkable capabilities in interpreting nuanced, multi-modal information, including the rich emotional content conveyed in human speech.This opens significant new opportunities for applications that can provide real-time, meaningful feedback to users. At the forefront of this opportunity is the task of Speech Emotion Captioning (SEC) [1].This task represents a significant advancement over traditional Speech Emotion Recognition (SER) [2, 3, 4]. While conventional SER systems typically output a single, fixed category label, such as ”Angry” or ”Happy,” SEC aims to generate a rich, descriptive, and open-ended natural language caption. For example (Figure 1), instead of just ”Happy” or ”Angry” an SEC system might output, ”Her voice trembled slightly, filled with surprise and joy”.This level of detailed, human-like understanding is highly valuable for next-generation applications in areas like mobile health monitoring, advanced assistive communication systems, and sophisticated real-time customer service analysis. However, for such applications to be considered truly viable and trustworthy for real-world use, they must satisfy two non-negotiable constraints: they must operate in real-time to be useful, and they must be fundamentally private to protect sensitive user data.

However, deploying these powerful LLMs to meet the simultaneous demands of real-time performance and user privacy creates a significant engineering and ethical dilemma. This problem manifests as a core conflict between the two primary deployment strategies: powerful, centralized cloud computing versus localized, resource-constrained edge devices. On one hand, a ”Cloud-Only” approach allows us to use state-of-the-art models running on high-performance GPUs. This strategy delivers the gold standard in caption quality and fast computation. But this power comes at an unacceptable price. It introduces significant network latency from the constant round-trip communication between the device and the server, making a truly responsive real-time experience difficult. More critically, this model requires the user's raw, sensitive audio data to be continuously uploaded to a third-party server, creating a severe privacy risk. On the other hand, an "Edge-Only" approach, where a smaller, compact Small Language Model (SLM) runs directly on the user's device, perfectly solves the privacy problem. In this model, the audio data never leaves the local environment. However, this strategy suffers from its own debilitating weaknesses.These smaller models inherently possess lower descriptive power, resulting in captions of significantly reduced quality. Furthermore, the limited computational power of typical edge devices, which often rely on just a few CPU cores, means that inference is extremely slow, failing the real-time performance requirement. It is clear that neither of these binary, "all-or-nothing" solutions can satisfy the complex needs of our target applications.To navigate this landscape and resolve the conflict between quality, privacy, and latency, this report proposes a novel hybrid edge-cloud framework. Our approach is built upon the principles of Speculative Decoding, a "Draft and Verify" mechanism designed to intelligently distribute the computational workload[5]. Its schematic diagram is shown in the Figure 2. The core concept is as follows: we allow the fast, local, but less accurate SLM on the edge device to do the majority of the work. It "drafts" caption segments (text tokens) first. These drafts are then efficiently "verified" by the powerful, high-quality LLM in the cloud. Critically, our system is designed as privacy-first. The raw audio data is never uploaded; it is processed only on the local device to generate the initial text draft. The only data transmitted to the cloud are the non-sensitive, drafted text tokens. This architecture allows us to maintain the strict privacy guarantee of an edge-only system while strategically tapping into the superior intelligence of the cloud LLM.Our central hypothesis is that this hybrid framework can "buy back" a significant portion of the cloud model's high quality, starting from the privacy-preserving edge baseline, by incurring a measurable and hopefully acceptable latency cost.

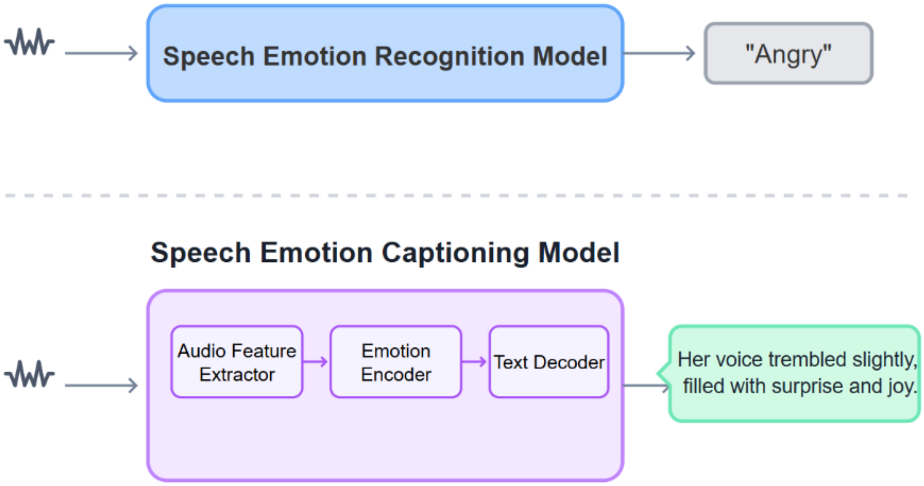
Speech Emotion Recognition Model
"Angry"
Speech Emotion Captioning Model
Audio Feature Extractor
Emotion Encoder
Text Decoder
Her voice trembled slightly, filled with surprise and joy.

Figure 1: Comparison of Speech Emotion Recognition (SER) model and Speech Emotion Captioning (SEC) Model.

This research, therefore, sets out to empirically evaluate this hypothesis and rigorously quantify the associated trade-offs. We aim to answer three key questions. First, can this hybrid frame achieve significantly better captioning quality than a baseline frame that only uses the edges, thus proving the feasibility of the concept? Second, what is the full cost of this quality improvement? We must measure this "price" not only in end-to-end latency (such as Time-To-First-Token and Total Time) but also in total resource consumption (including edge CPU, device RAM, and cloud GPU utilization). Finally, and most importantly, how efficient is this trade-off? Is the extra time and resource expenditure, which we term "collaboration overhead", justified by the corresponding gain in caption quality? This report makes several contributions. We present the design and implementation of a privacy-preserving hybrid framework specifically for the Speech Emotion Captioning task. We conduct a rigorous, empirical analysis of this framework on emulated, realistic hardware (a 2-core CPU edge environment versus an A100 GPU cloud environment). Finally, through this analysis, we identify the primary bottleneck of this approach—an inefficient trade-off—which provides clear direction for future optimization.

To answer these questions, the following sections of this report will be organized. The second section

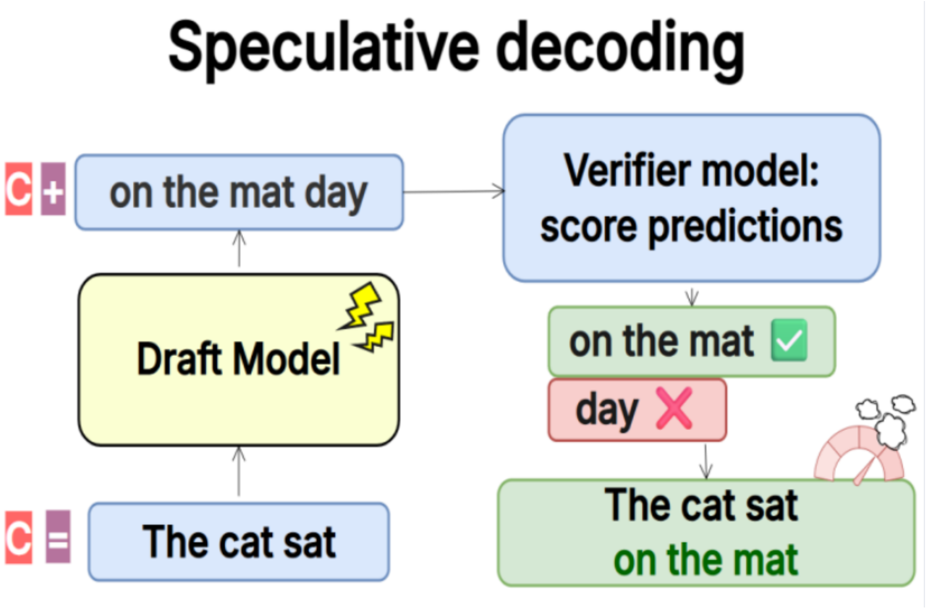

Figure 2: Schematic diagram of speculative decoding.

comprehensively reviews the fundamental literature in areas such as voice emotional captioning, speculative decoding, and edge cloud collaboration systems. Section 3 provides a detailed account of the complete design and approach of the hybrid framework we proposed, including draft formulation, gating, and verification protocols. Section 4 describes the experimental setup, dataset, evaluation metrics, and the specific hardware environment used for the baseline. Section 5 presents and analyzes in detail our experimental results, with a focus on quality and efficiency. Then, it delved into our findings and their impact on the ”quality-delay” trade-off. Finally, Section 6 summarizes this report and Outlines the potential directions for future research.

2 Literature Review

This research builds upon a foundation of recent advancements in large language model (LLM) inference, edge-cloud systems, and speech processing. The core challenge of LLM deployment is mitigating the high computational cost and latency of autoregressive generation, where each token is generated sequentially. A primary breakthrough in addressing this bottleneck is Speculative Decoding, introduced by Leviathan et al. [5]. This technique employs a ”draft-verify” scheme: a lightweight, fast ”drafter” model proposes a block of tokens in parallel. A larger, more powerful ”verifier” model then checks this draft in a single forward pass, accepting the longest ”safe prefix” of tokens that matches its own predictions. This line of inquiry has demonstrated substantial speedups in generation by reducing the number of required decoding steps, all while mathematically preserving the exact output distribution of the original, larger model. This principle of using a fast drafter to accelerate a powerful verifier is the core mechanism upon which our framework is built.

This token-level reasoning was quickly extended from a single-machine optimization into a new architectural pattern for Edge-Cloud Collaborative Inference . Recognizing that the ”drafter” and ”verifier” do not need to be on the same machine, Hao et al. [6] proposed a ”Hybrid SLM-LLM” framework . In this architecture, a Small Language Model (SLM) runs on the user’s local device (the edge), acting as the private and responsive drafter. The powerful LLM, meanwhile, resides in the cloud, acting as the high-quality verifier. This system established that an on-device model can generate the majority of ”easy” tokens, while the cloud LLM is invoked only to verify and correct the ”difficult” ones. This hybrid approach was shown to achieve near-LLM quality at a fraction of the remote cost and token usage, establishing token-granular collaboration as a viable alternative to other complex methods like layer partitioning. This concept of selective, on-demand escalation based on uncertainty is central to our work. Further research into multimodal systems, such as CD-CCA [7], reinforced this principle by proposing methods to ”send only uncertain tokens upstream (UTS),” demonstrating a principled approach to prioritizing what data must cross the edge-cloud boundary.

The successful application of speculative decoding has recently been extended into the speech domain, most notably in Automatic Speech Recognition (ASR). The work on SpecASR [8] specifically adapted the draft-verify mechanism for use with large, LLM-based ASR models. By tailoring parameters like draft length and token recycling to the unique characteristics of speech transcription, this research reported significant 3x latency reductions without any loss in recognition accuracy. This is a critical precedent for our project, as it provides strong empirical evidence that the latency-saving benefits of speculative decoding can transfer from text-only tasks to audio-conditioned speech pipelines. However, the focus of SpecASR was purely on latency reduction for a single, large model. It did not investigate the ”Hybrid SLM-LLM” architecture, nor did it address the benefits of cloud token savings or the critical privacy concerns associated with an edge-cloud system.

Finally, our research applies these concepts to the specific task of Speech Emotion Captioning (SEC). SEC itself is an emerging field that reframes emotion understanding from a simple classification task into an open-ended, descriptive captioning problem. Foundational works such as SE-Cap [1] established the audio-to-caption pipeline, demonstrating the feasibility of generating rich, natural-language descriptions of emotion from audio . Subsequent work like AlignCap [9] further improved the robustness and human-likeness of these captions through better alignment and preference optimization. These studies define the core task, model architectures, and public datasets (such as MER2024[10], IEMOCAP[11], and ESD[12]) that our work will inherit and build upon. While these papers define what to do (the SEC task), they do not address how to deploy these models in a real-time, privacy-preserving manner. This review reveals a clear research gap: while the task (SEC), the mechanism (Speculative Decoding), and the architecture (Hybrid Edge-Cloud) all exist, they have not been integrated. No prior work has designed or evaluated a privacy-first, token-level collaborative framework to balance the quality, latency, and privacy trade-offs specifically for Speech Emotion Captioning. This research aims to fill that gap.

3 Methodology

This research transitions from the theoretical trade-offs identified in the literature review to the design and implementation of a practical, hybrid computational framework. This section provides a comprehensive technical description of our proposed methodology. We detail the system's high-level architecture, the specific protocols that govern the interaction between the edge device and the cloud server, the foundational privacy-first data model, and the rationale behind our key design choices.

3.1 System Architecture Overview

Our proposed solution is a token-level collaborative framework designed to intelligently and dynamically distribute the workload of Speech Emotion Captioning (SEC). The architecture is fundamentally hybrid, comprised of two primary components that operate in a closed loop, as illustrated in our Method Overview Figure 3.

An Edge SLM (Small Language Model): This is a compact, 3-billion parameter model (Qwen-2.5-Omni-3B) intended to run on the user's local, resource-constrained device. It functions as the system's ”drafter,” responsible for generating initial text sequences.

A Cloud LLM (Large Language Model): This is a larger, 7-billion parameter model (Qwen-2.5-Omni-7B) hosted on a high-performance server. It functions as the system's ”verifier,” responsible for ensuring the quality and accuracy of the generated text.

The core operational flow is guided by the principles of speculative decoding , adapted for an edge-cloud architecture. The process begins on the edge device, where audio features are first extracted from the user's speech. The Edge SLM then drafts a small chunk of k tokens. Following this draft, a critical ”Gating” module—the autonomous decision-making brain of the edge device—computes the uncertainty of this draft .

If this uncertainty is below a predefined threshold (i.e., the chunk is ”Easy”), the system accepts the draft locally without any network communication. The Edge SLM's internal state is updated, and it immediately begins drafting the next k tokens. This local-only loop is the system's fast path, designed to handle the majority of ”simple” text generation.

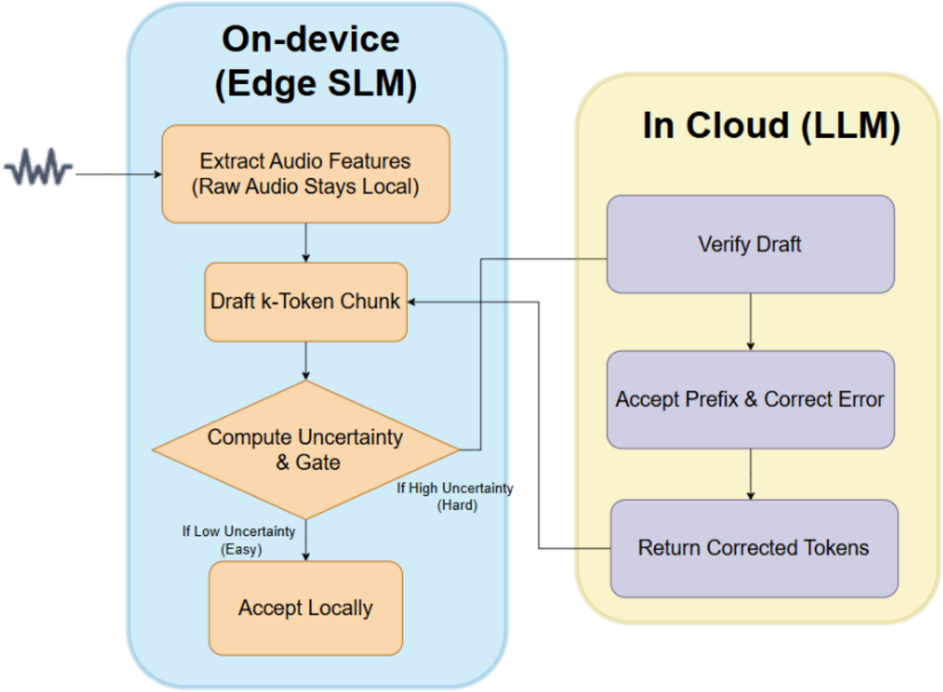

Figure 3: The architecture diagram for our hybrid system.

Conversely, if the uncertainty is high (i.e., the chunk is ”Hard”), the Gating module triggers an escalation. It packages the drafted token IDs and the relevant audio features and sends this multimodal packet to the cloud . The Cloud LLM then performs an audio-conditioned verification. It accepts the correct portion of the draft, corrects the first identified error, and returns the corrected tokens to the edge. The Edge SLM synchronizes its state with this correction and resumes drafting from the new, verified position. This entire architecture is designed to balance the speed and privacy of the edge with the quality of the cloud.

3.2 The Privacy-First Data Model

A non-negotiable, foundational principle of this architecture is its privacy-first data model. This model is designed to directly address the primary failure point of traditional cloud-based systems: the exposure of sensitive user data. The fundamental and non-negotiable rule of our framework is that the user’s raw audio waveform never leaves the edge device.

Raw audio is a unique biometric identifier, and its exposure constitutes a significant privacy breach. In our system, all processing of the raw audio occurs locally on the user’s hardware.

However, to perform a meaningful verification of an audio caption, the cloud verifier must have access to the audio’s content. Our framework resolves this conflict by separating the audio data into two forms:

1. Raw Audio (Private): The original waveform. This stays on the edge.

2. Audio Features (Transmitted): An abstracted, processed representation (e.g., Log-Mel spectrograms or other embeddings). This is sent to the cloud.

This distinction is the key to our privacy model. The audio features are an abstracted representation necessary for the model to perform its task, but they are not the raw, identifiable voice data. This significantly mitigates the privacy risk compared to uploading the original audio file. The data packet transmitted during an escalation event therefore contains the drafted token IDs and their corresponding audio features. This data-minimization approach fulfills the strict privacy requirements for applications in sensitive domains while still enabling powerful, audio-aware cloud verification.

3.3 On-Device (Edge) Module

The On-Device Module, running the Edge SLM, is the user-facing component of the system. It is responsible for all local processing and, most importantly, the autonomous decision-making that governs the entire collaborative process. In this study, this module is represented by the Qwen-2.5-Omni-3B model, chosen for its multimodal capabilities and compact 3-billion parameter size, which is on the upper end of what is plausible for CPU-bound inference on a high-end edge device.

3.3.1 Feature Extraction and k-Token Drafting

The module's pipeline begins when the user speaks. The raw audio waveform is captured and immediately processed locally into an audio feature representation that the Qwen-Omni model can understand (e.g., Log-Mel spectrograms or other internal audio features). This feature set is stored locally. The Edge SLM's text decoder, conditioned on these local audio features, then performs its primary task: k-Token Drafting.

The model autoregressively generates a small, fixed-size chunk of k tokens (e.g., k=5). The choice of k is a critical hyperparameter that defines a core trade-off. A very large k (e.g., k=20) would offer high potential for acceleration, as a single verification step could accept many tokens at once. However, it also carries a high risk of failure; a single error early in the chunk would cause the entire remainder of the draft to be discarded, wasting all the edge computation used to generate it. Conversely, a very small k (e.g., k=1) would be identical to standard autoregressive generation, offering no speedup. Our implementation uses a small k value as a balance, aiming to gain a modest speedup per cycle while minimizing the computational waste from rejected drafts. This drafted chunk, along with the probability distributions (logits) that produced it, is then passed to the gating module.

3.3.2 Uncertainty-Based Gating

This submodule is the "brain" of the local system and the implementation of our first research question. Its sole responsibility is to decide, for each drafted chunk, whether to "Accept Locally" or "Escalate to Cloud." The goal of this gate is to act as an intelligent, cost-saving filter. It must minimize expensive cloud calls by only escalating token chunks that are truly "difficult" or "uncertain" for the 3B model, while confidently accepting "easy" chunks locally.

As outlined in our research proposal, this study's primary implementation uses token-level entropy as the primary uncertainty metric. Entropy, in this context, is a standard information-theoretic measure of a probability distribution's "flatness" or "peakedness." After the k tokens are drafted, the system inspects the logits (the raw probability distributions) generated by the Edge SLM for each of those k steps. It then calculates the entropy for each token's distribution.

A low entropy value (a "peaked" distribution) indicates that the model was highly "confident" in its chosen token, as most of the probability mass was concentrated on that single choice. A high entropy value (a "flat" distribution) indicates the model was "confused" or "uncertain," as the probability was spread thinly across many possible tokens.

The gate's decision logic is straightforward: it finds the maximum entropy value within the k-token chunk. If this maximum entropy exceeds a predefined entropy threshold, the entire chunk is deemed "Hard" and the gating mechanism triggers an escalation. This escalation involves packaging both the drafted token IDs and the corresponding audio features for transmission. If all tokens in the chunk are "confident" (i.e., below the entropy threshold), the chunk is deemed "Easy," accepted locally, and the system loops back, saving both latency and cost.

3.4 In-Cloud (LLM) Verification Module

Once a "Hard" chunk is escalated, the In-Cloud Module, running the powerful Qwen-2.5-Omni-7B model, takes over. This module is designed not for slow, sequential generation, but for high-efficiency, audio-conditioned verification.

3.4.1 Multimodal Verification

A naive verification method would be for the 7B model to autoregressively generate k tokens and see if they match. This would be slow. Our module, as visualized on Figure 4 , is far more efficient. It

performs a Multimodal Verification.
The Cloud LLM receives the full multimodal packet from the edge. This packet contains:
The [Accepted Context] (the text generated so far).
The [k Draft Tokens] (the text being verified).
The [Audio Features] (the audio context for the entire utterance).
The 7B model's architecture allows it to process all three inputs simultaneously. It executes one single forward pass on this entire combined, multimodal sequence . This is the key to its efficiency. This single computation generates the probability distributions (logits) for all token positions at once. This allows the system to efficiently "look ahead" and, in one step, retrieve the 7B model's audio-conditioned prediction for what should have come after token t, t+1, t+2, and so on. These logits are then used to validate the draft.

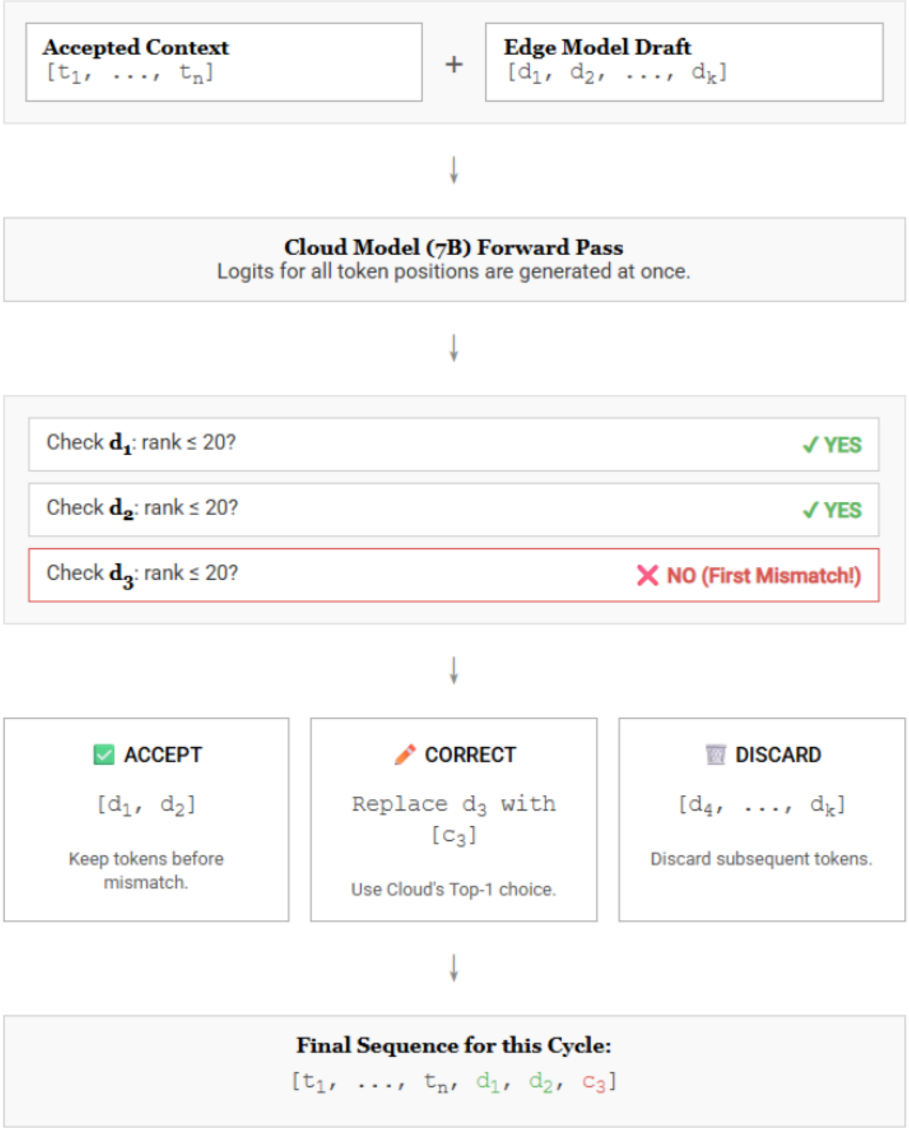
Accepted Context
$[t_1, \ldots, t_n]$
+
Edge Model Draft
$[d_1, d_2, \ldots, d_k]$
Cloud Model (7B) Forward Pass
Logits for all token positions are generated at once.
Check $\mathbf{d_1}$: rank ≤ 20?
✓ YES
Check $\mathbf{d_2}$: rank ≤ 20?
✓ YES
Check $\mathbf{d_3}$: rank ≤ 20?
✕ NO (First Mismatch!)
ACCEPT
$[d_1, d_2]$
Keep tokens before mismatch.
CORRECT
Replace d_3 with $[c_3]$
Use Cloud's Top-1 choice.
DISCARD
$[d_4, \ldots, d_k]$
Discard subsequent tokens.
Final Sequence for this Cycle:
$[t_1, \ldots, t_n, d_1, d_2, c_3]$

Figure 4: The schematic diagram of cloud model verification and correction.

3.4.2 Rank-Based Acceptance Protocol

The validation logic is crucial. As outlined in the proposal , a simple probability threshold is brittle and hard to tune. An even stricter Top-1 match (i.e., "does the draft token d_1 exactly match the cloud's single best prediction given the audio?") would be too severe. The 3B model might generate a valid synonym (e.g., ”pleased” instead of ”happy”), and a strict Top-1 match would wrongly reject it, leading to a high rejection rate and defeating the purpose of drafting.

Instead, we implement a more lenient and robust Rank-Based Acceptance protocol . This protocol is not concerned with what the 3B model chose, but rather how plausible its choice was, according to the 7B model which is also listening to the same audio. For each token d_i in the k-token draft, the system checks its rank within the logits predicted by the Cloud LLM for that position. If the draft token d_i falls within a ”reasonable” range—in our implementation, a Top-20 rank threshold—it is considered a valid continuation and is accepted . This allows the Edge SLM to be ”creative,” as long as its choice is still ”plausible” to the more powerful model.

3.4.3 Correction and Rollback Protocol

This acceptance continues token by token (d_1, d_2, ...) until the first mismatch is found—that is, a draft token that falls outside the Top-20 rank. At this point, the protocol is precise and deterministic:

Accept: All tokens before the mismatched token (e.g., d_1, d_2) are formally accepted.

Correct: The single mismatched token (d_3) is replaced by the Cloud LLM's preferred Top-1 prediction (c_3). This Top-1 prediction is audio-conditioned, representing the 7B model's ”best guess” for that position.

Discard: All remaining tokens in the draft after the first error (e.g., d_4, d_5, ... d_k) are immediately thrown away . This is necessary because the context has now changed (c_3 is the new token), so the rest of the 3B model's draft, which was based on d_3, is now invalid.

The cloud then returns only the accepted and corrected tokens (e.g., [d_1, d_2, c_3]) to the edge device. The Edge SLM receives this correction packet, updates its internal state (including its KV cache) by processing [d_1, d_2, c_3] as if it had generated them itself, and thus ”resynchronizes” its state with the cloud. It is now ready to draft the next k-token chunk starting from this new, verified context.

3.5 Model Selection Rationale

The models for this study were chosen with specific intent. The Qwen-2.5-Omni series was selected as it represents an advanced family of multimodal models. This is a critical, non-negotiable requirement for this methodology. Both the Edge SLM and the Cloud LLM must be able to natively process audio features as part of their context. This allows the Edge SLM to generate an audio-conditioned draft and, more importantly, allows the Cloud LLM to perform an audio-conditioned verification.

The 3B-parameter model was chosen as the Edge SLM, representing a large but plausible model that could run on high-end edge devices (though requiring CPU limitations in our simulation to represent a constrained environment). The 7B-parameter model was chosen as the Cloud LLM. This provides a clear and significant step up in quality and power, creating a meaningful gap for the speculative decoding to bridge, while still being small enough to provide fast verification on an A100 GPU.

Finally, a key technical prerequisite is that both models are from the same family and share the same tokenizer. This shared tokenizer ensures that the token IDs generated by the 3B model (e.g., token 4510 for ”hello”) are perfectly and unambiguously understood by the 7B model. This shared vocabulary is what enables the efficient, privacy-preserving communication protocol, which relies entirely on transmitting numerical token IDs.

4 Experimental Setup

To empirically evaluate the performance, latency, and resource trade-offs of our proposed hybrid framework, we designed a rigorous experimental setup. This section details the comparative baselines, the dataset used for evaluation, the specific hardware environments created to simulate edge and cloud constraints, and the metrics used to measure the outcomes.

4.1 Baselines for Comparison

To effectively quantify the trade-offs of our method, we established three distinct experimental conditions. Our proposed Speculative Decoding framework is compared against two baselines, representing the two extremes of the edge-cloud dilemma.

Edge-Only (SLM Baseline): This baseline consists of the Qwen-2.5-Omni-3B model running in our resource-constrained edge environment. This model performs all tasks locally, from audio processing to text generation. It represents the ”best-case” scenario for privacy and data sovereignty, as no data ever leaves the device. However, it is hypothesized to suffer from the lowest caption quality and highest computational latency due to the small model size and limited CPU resources.

Cloud-Only (LLM Baseline): This baseline consists of the powerful Qwen-2.5-Omni-7B model running in our high-resource cloud environment. This model represents the theoretical ”gold standard” for caption quality. However, it fails our core privacy requirement, as it necessitates uploading the user's raw (or, in this case, fully-featured) audio data to the cloud. It serves as the upper bound for quality against which our hybrid method is measured.

Speculative Decoding (Our Method): This is our proposed hybrid framework, which uses the 3B Edge-Only model as its local drafter and the 7B Cloud-Only model as its remote verifier. This method is designed to test the central hypothesis: that we can approach the Cloud-Only quality while retaining the Edge-Only privacy guarantee, at the cost of a measurable collaboration overhead.

Additionally, we reference the reported scores for Qwen-Audio from the official MER2024 challenge paper [13]. It is important to note that this is not a direct, apples-to-apples comparison. The official paper's results involved a post-processing step where model outputs were refined using GPT-3.5. Our experiments evaluate the raw output of the Qwen models without this refinement step. Therefore, this score is included for context, but our primary analysis focuses on the direct comparison between our three implemented baselines.

4.2 Dataset

For our evaluation, we used the dataset provided by the MER2024 Challenge [10]. This is a publicly available, multi-modal dataset designed for emotion recognition and captioning tasks. For the specific needs of our Speech Emotion Captioning (SEC) project, we extracted the audio component from this larger dataset.

Our final evaluation benchmark consists of 332 audio files. Each audio file is accompanied by one or more human-annotated reference captions that provide a natural-language description of the perceived emotion. These reference captions serve as the ground truth for all our caption quality metrics. The audio files were processed according to the methodology described in Section 3, with features extracted locally on the (simulated) edge device before being fed into the models.

4.3 Hardware Emulation Environment

A critical component of this research was to create a fair and realistic comparison between edge and cloud performance. Since deploying models on physical, resource-constrained mobile devices introduces significant variability, we used a controlled, emulated environment on a high-performance computing (HPC) cluster.

Edge Environment (Resource-Constrained): To simulate the computational limitations of a typical high-end edge device (e.g., a smartphone or laptop running on its CPU), we created a CPU-limited environment on the HPC cluster. As defined in our implementation , each experiment for the Edge-Only and Speculative Decoding models was programmatically restricted to using only 2 CPU cores. This constraint is designed to mimic the processing bottleneck of a mobile CPU and prevent the powerful server-grade processor from skewing the latency results. The 3B model was run in 32-bit floating-point (FP32) precision on these cores.

Cloud Environment (High-Resource): To represent the powerful, ”unlimited” resources of a modern data center, our cloud environment was hosted on the University of Melbourne's Spartan HPC cluster. For the Cloud-Only and the verifier component of our Speculative Decoding method, each experiment was allocated a dedicated, high-performance node. This node was equipped with 1 NVIDIA A100 GPU, 8 CPU cores, and 64 GB of system RAM. This setup ensures that the 7B model's inference is accelerated by a state-of-the-art GPU, providing a realistic measure of a high-performance cloud baseline.

4.4 Evaluation Metrics

To comprehensively answer our research questions, we evaluated our baselines across two distinct categories of metrics: caption quality and system efficiency.

4.4.1 Caption Quality Metrics

To measure the semantic similarity between our models' generated captions and the human reference captions, we employed a standard suite of metrics used in captioning and machine translation:
BLEU (Bilingual Evaluation Understudy)[14]: We report BLEU-1 and BLEU-4, which measure the precision of matching 1-gram (single words) and 4-grams (four-word phrases), respectively. BLEU-1 is a good indicator of lexical overlap, while BLEU-4 is a much stricter measure of grammatical and structural similarity.
METEOR (Metric for Evaluation of Translation with Explicit ORdering)[15]: A more advanced metric that considers precision and recall, as well as stemming and synonym matching, providing a more robust score for semantic similarity.
ROUGE-L (Recall-Oriented Understudy for Gisting Evaluation)[16]: We use ROUGE-L, which measures the Longest Common Subsequence (LCS) between the generated and reference captions. This is useful for evaluating how well the core ”gist” of the caption is captured.

4.4.2 Efficiency & Resource Utilization Metrics

To quantify the ”cost” of each method, we recorded a detailed set of performance and resource metrics. These are critical for understanding the latency and deployment feasibility:
TTFT (s) (Time To First Token): The time in seconds from the start of the request until the first token is generated. This is a crucial metric for measuring user-perceived responsiveness.
ITPS (/s) (Input Tokens Per Second): Measures the speed of the initial processing (prefill) of the input audio features and text prompt.
OET (s) (Output Emission Time): The total time in seconds spent only on generating the output tokens (i.e., total time minus TTFT).
OTPS (/s) (Output Tokens Per Second): The throughput of the generation process (total output tokens divided by OET). A higher OTPS means faster generation.
Total Time (s): The full end-to-end latency, from the initial request to the generation of the final token.
CPU (%) / RAM (GB): The percentage of CPU cores and the gigabytes of system RAM consumed by the model, measured on the respective (edge or cloud) machine.
GPU (%) / GPU Memory (GB): The percentage of GPU utilization and the gigabytes of dedicated GPU VRAM consumed, measured on the cloud machine.

5 Results and Analysis

This section presents the empirical results of our experiments, directly addressing the research questions posed in the introduction. We first evaluate the Caption Quality (RQ1) to determine if our hybrid framework can successfully improve performance over the edge baseline. We then conduct a detailed analysis of the Efficiency, Latency, and Resource Costs (RQ2) to quantify the price of this collaboration. Finally, we synthesize these findings to analyze the Overall Trade-Off Efficiency (RQ3).

5.1 Caption Quality Analysis

The primary goal of this research was to determine if a hybrid edge-cloud framework could improve caption quality while preserving user privacy. To answer this, we compared the outputs of our three baselines against the 332 reference captions from the MER2024 dataset using a standard suite of N-gram-based quality metrics. The results are presented in Table 1.
The results provide a clear and compelling answer. The first key finding is the significant quality gap between the baselines. The Edge-Only model (7.72 BLEU-1) performs substantially worse than the Cloud-Only model (15.26 BLEU-1). This confirms our initial premise: the resource-constrained 3B model is not capable of producing captions of the same quality as the 7B model.

Table 1: Result (Quality): Comparison of Caption Quality Metrics. T, V, and A represent the modalities used: Text (Transcription), Visual, and Audio, respectively.

Model	**T**	**V**	**A**	**BLEU-1**	**BLEU-4**	**METEOR**	**ROUGE-L**
Edge(Qwen-2.5-Omni-3B)	x	x	✓	7.72	0.41	7.17	6.94
Cloud(Qwen-2.5-Omni-7B)	x	x	✓	15.26	0.77	13.06	12.72
Speculative decoding	x	x	✓	8.64	0.45	8.06	8.11
Qwen-Audio[13]	✓	x	✓	21.87	6.55	21.65	20.81

The most critical finding, however, is the performance of our Speculative Decoding method. It achieved a BLEU-1 score of 8.64, which is demonstrably higher than the Edge-Only baseline of 7.72. This trend is consistent across all other metrics, with METEOR rising from 7.17 to 8.06 and ROUGE-L rising from 6.94 to 8.11. This result is the primary success of the experiment: our hybrid framework successfully proved the core hypothesis. It is possible to leverage a cloud verifier to improve the quality of a privacy-preserving edge model.

It is important to address the low BLEU-4 scores across all our methods (0.41 to 0.45). BLEU-4 is an extremely strict metric that requires an exact match of four-word phrases. For an open-ended, creative task like SEC, where many different phrasings can be correct, achieving a high BLEU-4 score is exceptionally difficult. The low scores are therefore expected and are more reflective of the metric's unsuitability for this task than a specific failure of the models. The modest improvement from 0.41 to 0.45 suggests that our method's corrections are more lexical (improving individual words, captured by BLEU-1) than deeply structural.

Finally, we must contextualize our results against the Qwen-Audio SOTA reference. Our highest score (15.26) is clearly lower than the paper's reported 21.87. This discrepancy is expected and is not an apples-to-apples comparison. The official MER2024 paper's methodology included a final refinement step using GPT-3.5 to rewrite and enrich the captions. Our experiment, due to computational (quota) constraints, evaluates the raw output of the Qwen models. Thus, our results are a fair comparison amongst our baselines and successfully answer RQ1: Yes, the hybrid concept works, and quality is demonstrably improved.

5.2 Efficiency and Latency Analysis

The second research question asked to quantify the cost of this quality improvement. Our analysis of the efficiency and latency metrics, presented in Table 2, reveals a complex and critical part of our story.

Table 2: Result (Efficiency): Comparison of Latency and Resource Metrics

Model	**TTFT**	**ITPS**	**OTPS**	**OET**	**Total time**	**CPU(%)**	**RAM**	**GPU(%)**	**GPU Memory**
Edge(3B)	7.74	2327.01	2.10	34.80	34.89	201.56	20.55	0.0	0.86
Cloud(7B)	0.39	1461.04	24.16	3.25	3.38	98.25	4.56	89.42	42.49
Speculative decoding	7.69	2413.77	1.83	43.79	43.87	196.06	24.75	96.05	49.87

The most critical finding of all is the Total Time. The Cloud-Only baseline is the fastest (3.38s). The Edge-Only baseline is, as expected, very slow (34.89s). But our Speculative Decoding method (43.87s) is even slower than the Edge-Only baseline it was meant to improve. This is further confirmed by the OTPS (Output Tokens Per Second), where the hybrid method (1.83) is the slowest of all, processing fewer tokens per second than the edge (2.10).

This finding is not an anomaly; it is a direct measurement of collaboration overhead. The Speculative method is slower because the time saved by occasionally accepting k tokens at once is completely overwhelmed by the time lost to the new steps we introduced: computing uncertainty on the CPU, packaging and sending the data packet (network latency), waiting for the cloud GPU to perform verification, and receiving the correction. This overhead is paid every time the gate escalates, and the cumulative cost results in a net loss of performance compared to the simpler, "dumber" Edge-Only baseline.

5.3 Resource Utilization Analysis

The efficiency table also provides a clear picture of the resource cost (Table 2, rightmost columns).
CPU & RAM: The Edge baseline consumed 201.56% of the CPU and 20.55 GB of RAM. Our Speculative method consumed 196.06% CPU and 24.75 GB of RAM. The near-200% CPU usage confirms our experimental setup was successful: both methods completely saturated the 2 CPU cores we allocated. The higher RAM for the hybrid method is also logical, as it must load the 3B model plus the additional logic for gating, network I/O, and state synchronization.
GPU: The Edge baseline correctly used 0.0% GPU. The Cloud baseline showed a high, sustained utilization of 89.42%. Most telling is the Speculative method's 96.05% GPU utilization. This does not mean it used more total GPU compute, but rather that when it was called for verification, it spiked to near-peak utilization.
This analysis reveals that our hybrid method represents a ”worst of both worlds” scenario in terms of resource consumption. It simultaneously taxes the edge device's CPU and RAM to their absolute limits, while also demanding peak-time access to a powerful cloud GPU. It is, by a significant margin, the most resource-intensive deployment model of the three.

5.4 The Inefficient ”Quality-for-Latency” Trade-Off

The results from Sections 5.1, 5.2, and 5.3 provide a complete, data-driven answer to our third research question. We have established a clear trade-off:
What We Gained (The Reward): A 1-point BLEU-1 increase (8.64 vs 7.72) and a 1-point ROUGE-L increase (8.11 vs 6.94).
What It Cost (The Price): A 9-second increase in total latency (43.87s vs 34.89s), no improvement in TTFT, and the highest combined CPU, RAM, and GPU resource load.
This leads to the central conclusion of this report: The current framework demonstrates an inefficient ”quality-for-latency” trade-off. We are paying a very high price (9 extra seconds of latency, full CPU and GPU load) for a modest reward (a 1-point BLEU-1 gain).
The root cause of this inefficiency is not the concept of speculative decoding, but the implementation of our gating mechanism. Our current gate, based on a simple entropy threshold, is ”trigger-happy.” It is not ”smart” enough to distinguish between a token that is ”uncertain but low-impact” and a token that is ”uncertain and high-impact.” It escalates both. As a result, the system is forced to pay the full, 9-second latency cost for every escalation, but it is likely that many of these escalations are for low-impact tokens that do not significantly change the final quality or meaning of the caption.
This analysis successfully moves our research from ”Does this work?” to a much more precise and valuable question: ”How do we make this efficient?” We have not found a failure, but rather we have precisely identified the real bottleneck: the inefficiency of the gating logic. This insight, which we will explore in the final section, provides a clear, data-driven path for all future work.

6 Conclusion and Future Work

This research set out to address a critical challenge in modern mobile computing: the conflict between the high-quality inference of large language models (LLMs) and the strict, real-world requirements of user privacy and real-time responsiveness. We focused on the task of Speech Emotion Captioning (SEC), where this conflict is particularly pronounced. The ”all-or-nothing” approaches—a privacy-violating Cloud-Only model or a low-quality, slow Edge-Only model—were deemed insufficient for real-world applications . To solve this, we designed, implemented, and rigorously evaluated a novel hybrid framework based on token-level speculative decoding . This system utilized a 3B Edge SLM to draft captions locally, preserving privacy, while strategically escalating ”hard” tokens to a 7B Cloud LLM for verification and correction. Our primary research questions focused on whether this hybrid could improve quality (RQ1), what the full latency and resource costs would be (RQ2), and how efficient this quality-for-latency trade-off was (RQ3).
Our empirical findings provide clear answers to these questions and form the primary conclusions of this report. First, we successfully demonstrated that the core hypothesis is valid. Our Speculative Decoding framework achieved a caption quality score (e.g., 8.64 BLEU-1) that was demonstrably superior to the Edge-Only baseline (7.72 BLEU-1) across all standard metrics. This result confirms that a hybrid, privacy-preserving framework can successfully ”buy back” caption quality by leveraging a

more powerful cloud verifier. This proof of concept is a crucial contribution, validating the architectural approach.

Second, our detailed performance analysis quantified the significant cost of this collaboration. The Speculative Decoding method was, counter-intuitively, slower than the Edge-Only baseline, incurring an additional 9 seconds of total latency. This was attributed to the "collaboration overhead"—the cumulative time spent on local uncertainty computation, network round-trips, and cloud verification. This overhead was larger than any time saved by the speculative acceptance. Furthermore, we found that our hybrid method was the most resource-intensive, simultaneously saturating the 2-core edge CPU (196.06%) and the cloud GPU (96.05%), effectively inheriting the resource burdens of both baselines.

Finally, and most importantly, this research concludes that the proposed framework, in its current implementation, demonstrates an inefficient "quality-for-latency" trade-off . We are paying a very high price (9 extra seconds of latency, full CPU and GPU load) for a modest reward (a 1-point BLEU-1 gain). The root cause of this inefficiency was identified as the gating mechanism itself. A simple, entropy-based gate is not "smart" enough; it "cries wolf" too often, escalating low-impact tokens that do not significantly improve the final caption but still incur the full latency cost of a cloud call. We have not discovered a failure of the hybrid concept, but rather we have precisely identified the bottleneck of our initial implementation.

This key finding provides a clear, data-driven mandate for the next phase of research. The objective must now shift from "proof of concept" to "optimization of the trade-off". Future work must focus on developing a "smarter" framework that minimizes the latency cost while maximizing the quality gain. The first and most critical area of future work is to develop more intelligent gating mechanisms. We must move beyond simple entropy and conduct extensive ablation studies on a suite of different uncertainty metrics , as originally planned in our proposal. This includes investigating token-level log-probability margins (the difference between the top-1 and top-2 predictions) and block-level stability (how much a prediction changes as more context is added). The goal of this investigation is to identify the specific "high-impact hard tokens"—those tokens that are not only uncertain but are also most critical for improving the final caption's quality and meaning. By building a gate that only escalates these high-value tokens, we can drastically reduce the number of cloud calls and, therefore, the total collaboration overhead.

References

[1] Yaoxun Xu et al. "Secap: Speech emotion captioning with large language model". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 38. 17. 2024, pp. 19323–19331.

[2] Moataz El Ayadi, Mohamed S Kamel, and Fakhri Karray. "Survey on speech emotion recognition: Features, classification schemes, and databases". In: *Pattern recognition* 44.3 (2011), pp. 572–587.

[3] Tin Lay Nwe, Say Wei Foo, and Liyanage C De Silva. "Speech emotion recognition using hidden Markov models". In: *Speech communication* 41.4 (2003), pp. 603–623.

[4] Pengxu Jiang et al. "Parallelized convolutional recurrent neural network with spectral features for speech emotion recognition". In: *IEEE access* 7 (2019), pp. 90368–90377.

[5] Yaniv Leviathan, Matan Kalman, and Yossi Matias. "Fast inference from transformers via speculative decoding". In: *International Conference on Machine Learning*. PMLR. 2023, pp. 19274–19286.

[6] Zixu Hao et al. "Hybrid slm and llm for edge-cloud collaborative inference". In: *Proceedings of the Workshop on Edge and Mobile Foundation Models*. 2024, pp. 36–41.

[7] Guanqun Wang et al. "Cloud-device collaborative learning for multimodal large language models". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 12646–12655.

[8] Linye Wei et al. "SpecASR: Accelerating LLM-based Automatic Speech Recognition via Speculative Decoding". In: *arXiv preprint arXiv:2507.18181* (2025).

[9] Ziqi Liang, Haoxiang Shi, and Hanhui Chen. "Aligncap: Aligning speech emotion captioning to human preferences". In: *arXiv preprint arXiv:2410.19134* (2024).

[10] Zheng Lian et al. "Mer 2024: Semi-supervised learning, noise robustness, and open-vocabulary multimodal emotion recognition". In: *Proceedings of the 2nd International Workshop on Multimodal and Responsible Affective Computing.* 2024, pp. 41–48.

[11] Carlos Busso et al. "IEMOCAP: Interactive emotional dyadic motion capture database". In: *Language resources and evaluation* 42.4 (2008), pp. 335–359.

[12] Kun Zhou et al. "Seen and unseen emotional style transfer for voice conversion with a new emotional speech dataset". In: *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE. 2021, pp. 920–924.

[13] Zheng Lian et al. "OV-MER: Towards Open-Vocabulary Multimodal Emotion Recognition". In: *arXiv preprint arXiv:2410.01495* (2024).

[14] Kishore Papineni et al. "Bleu: a method for automatic evaluation of machine translation". In: *Proceedings of the 40th annual meeting of the Association for Computational Linguistics.* 2002, pp. 311–318.

[15] Satanjeev Banerjee and Alon Lavie. "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments". In: *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization.* 2005, pp. 65–72.

[16] Chin-Yew Lin. "Rouge: A package for automatic evaluation of summaries". In: *Text summarization branches out.* 2004, pp. 74–81.