COBRA: CTC-Oriented Blank Removal for Accelerated SpeechLLMs

Anonymous ACL submission

Abstract

Multimodal Large Language Models are transforming audio and visual processing by collapsing traditional pipelines into unified sequence models. In automatic speech recognition, aligning audio and text tokens enables the model to learn a shared latent space where phonetic, lexical, and semantic cues interact. However, processing high-throughput speech tokens with large-scale LLMs incurs prohibitive computational and memory costs, limiting their practical deployment.

To address this bottleneck, we present COBRA (**C**TC-**O**riented **B**lank **R**emoval for **A**ccelerated SpeechLLMs): a novel pruning strategy that leverages an auxiliary CTC head to evaluate the informational value of acoustic frames. By filtering out non-informative 'blank' tokens before they reach the LLM, COBRA significantly reduces the model's sequence length and computational footprint.

Experiments across various tasks and datasets demonstrate that COBRA is highly effective for speech-to-text tasks, maintaining strong performance profiles even under aggressive pruning. Finally, we show that varying the pruning rate during training enables the model to dynamically support multiple performance-efficiency trade-offs at inference time, offering a scalable path toward efficient edge deployment.

1 Introduction

Recently, Multimodal Large Language Models (MLLMs) have demonstrated remarkable benefits across a wide range of tasks (Wu et al., 2023), spanning both vision (Chen et al., 2023) and speech processing (Cui et al., 2025). By integrating information from multiple modalities, these models achieve improved generalization and robustness, enabling progress in tasks such as automatic speech recognition (Ma et al., 2024b; Fathullah et al., 2024), audio-visual speech recognition (Cappellazzo et al., 2025), visual question answering (Antol et al., 2015), multimodal reasoning (Liang et al., 2025b), and so on.

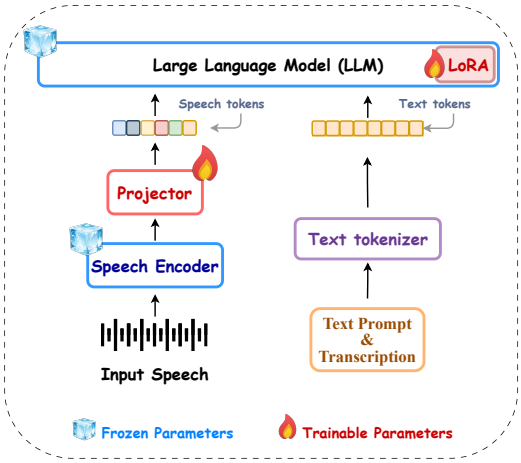

Figure 1: Overview of modern speechLLM architecture: a trainable projector connects a pre-trained speech encoder with a frozen LLM.

A prevalent strategy for adapting text-based LLMs to speech tasks is to add a lightweight projection module (Figure 1) between a pre-trained speech encoder and a pre-trained LLM. This projector maps the encoder's acoustic representations into the LLM's token-level embedding space, enabling the model to directly process speech-derived inputs across tasks. This architecture defines a distinct class of models known as Speech Language Models (SLLMs) (Cui et al., 2025).

Modern LLMs typically contain billions of parameters, demanding substantial computation and memory and making such models impractical for deployment on battery-powered or resource-constrained devices. Furthermore, latency remains a critical challenge: without specialized hardware accelerators such as GPUs or TPUs, the time required to process queries can be prohibitively high, limiting real-time applicability. These issues are further complicated in SLLMs since modern speech encoders produce speech tokens at high

throughput (approximately 50 Hz), generating a huge amount of data to be processed by the LLM. These limitations underscore the need for more efficient architectures and optimization strategies.

This work focuses on optimizing inference time and mitigating latency in SLLMs by proposing a novel token pruning strategy that reduces the number of tokens processed by the LLM. Our approach builds on the observation that standard speech-signal analysis, typically performed at a frame rate of 10-20 milliseconds, produces long and dense sequences of speech embeddings (see Section 3 for details). However, not all embeddings contribute meaningful information to the downstream task. By examining the token sequences produced by a CTC decoder (Fernandez et al., 2007) on the clean test set of *LibriSpeech* (Panayotov et al., 2015), we found that around 65% of the embeddings are classified as 'blank' tokens. These tokens primarily serve to model the alignment between the acoustic signal and its transcription, and therefore contribute little linguistic information. This empirical observation motivates a central research question: **can 'blank' tokens be pruned from the sequence processed by the LLM without degrading the performance?**

We evaluate COBRA across multiple spoken-language tasks, including Automatic Speech Recognition (ASR), Speech Translation (ST), emotion recognition (ER), and topic detection (TD). Our findings demonstrate that our CTC-based blank token filtering strategy remains robust even under substantial pruning, causing only minimal performance degradation. Moreover, we show that varying the pruning rate during training yields versatile models capable of dynamically adapting to different efficiency-performance trade-offs at inference time. The source code and pretrained model weights are available[1].

2 Related Works

Methods for reducing the computational cost and improving the efficiency of SLLMs generally fall into two main categories: *static* and *dynamic* approaches. Table 1 summarizes relevant papers for our work, giving a non-exhaustive overview of recent trends in efficient neural architectures and optimization methods and a concise snapshot of how the community is addressing the rising computational demands of modern neural systems.

[1]upon acceptance

Static methods focus on reducing or compressing the model itself, for example, by shrinking its architecture, pruning parameters, or applying quantization, so that the computational footprint is lowered independently of the input. Examples are: compression/pruning transformations of the parameters that reduce model size (Winata et al., 2020; Wei et al., 2022; Yosuke et al., 2023; Moumen et al., 2026), quantization methods that lower numerical precision (Benoit et al., 2018; Nagel et al., 2021), and knowledge-distillation approaches that transfer information from a large teacher model to a more compact student (Kim and Hassan, 2020).

Dynamic methods, in contrast, adapt the computation to the characteristics of the input sequence. Depending on the input content, these approaches allow the model to allocate compute more selectively through mechanisms such as early-exit strategies (Zhou et al., 2020; Wright et al., 2024), and layer-dropping (Hannan et al., 2026; Someki et al., 2025). Similarly, token-pruning techniques also fall into the same category (Kim et al., 2022; Bonnaerens and Dambre, 2023), offering content-dependent length reduction. Building upon these dynamic pruning paradigms, COBRA selectively discards non-informative audio frames, bringing dynamic sequence compression to the language interface.

2.1 Token pruning

Methods for downsampling the tokens are typically implemented in multimodal LLMs via average pooling (Sun et al., 2017), or by linear concatenation (Ma et al., 2024b) of the adjacent embeddings. However, their efficacy is rather limited.

In the field of Visual Large Language Models (VLLMs) (et al., 2022; Bai et al., 2024; Dai et al., 2023; Liu et al., 2025, 2026), the dynamic image pruning is widely researched for high-resolution inputs, which heavily consume LLM context windows despite just 6% of attention weights driving 95% of the total attention (Li et al., 2025). For example, (Chen et al., 2024) prunes image tokens based on average attention scores. Other approaches exploit the cross-attention values between visual and textual tokens (Zhang et al., 2026), or leverage the LLM's ability to condense visual information into a compact set of latent tokens (Ye et al., 2025). Finally, (Wen et al., 2025) provide a systematic evaluation of token pruning in MLLMs, demonstrating that attention-based metrics suffer from severe positional bias and proposes an infor-

mation theoretic framework.

Concerning speech processing and ASR, (Guo et al., 2025) introduces a fusion mechanism that condenses speech representations by clustering adjacent frames via cosine similarity and merging them through CTC-based weighting. However, unlike our approach, this method does not remove 'blank' tokens but instead uses the CTC layer to control the quality of the merged representations. (Sannigrahi et al., 2026) proposes a sampling method that merges speech embeddings according to a cosine-distance similarity metric. The method has shown efficiency across ASR, ST, and Speech Question Answering (SQA) tasks. The work in (Ma et al., 2025a) instead feeds CTC posteriors, rather than speech embeddings, into the LLM, suppressing 'blank' token probabilities to emphasize more informative tokens.

However, in contrast to our COBRA approach, none of these methods apply CTC directly to prune the speech embeddings. Note that CTC-blank pruning has also been successfully applied in traditional encoder-decoder ASR systems (Grigoryan et al., 2025) and ST models (Gaido et al., 2021). In these settings, removing blank frames reduces redundancy in the encoder representations passed to a decoder that does not rely on a large language model, thereby improving efficiency without significantly degrading recognition or translation quality.

2.2 Other CTC-based methods

Many works leverage CTC information to improve efficiency. (Peng et al., 2026) considers simulated CTC posteriors, derived from text, within TASU (Text-only Alignment for Speech Understanding), achieving competitive zero-shot performance on both ASR and spoken language understanding tasks. To address the length mismatch between speech and text modalities, (Liang et al., 2025a) proposes 'AlignFormer', which leverages a CTC layer to provide alignment information and enables dynamic-window Q-Former layers to generate speech embeddings whose length and structure closely match those of the text sequence. The resulting ASR-trained model also generalizes effectively to other zero-shot speech tasks.

3 Proposed Approach

Figure 2 illustrates the SLLM architecture used in this work together with the proposed token-pruning method, COBRA. We leverage the speech encoder's auxiliary CTC head, which is pre-trained on an ASR task. Based on a pre-defined threshold (τ_{ctc}), embeddings classified by the CTC decoder as 'blank' token (ϵ) with high confidence are removed from the downstream LLM pipeline.

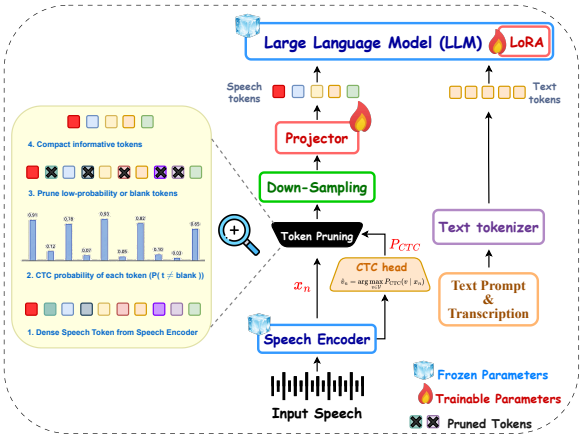

Figure 2: Overview of the CTC-pruning method. Embeddings produced by the encoder are selectively pruned (marked with ✖), based on their CTC scores.

Let $\mathbf{X} = (x_1, \ldots, x_N)$ denote the sequence of speech embeddings output by the speech encoder, where $\mathbf{x}_n \in \mathbb{R}^d$. The subset of retained embeddings $\mathbf{Y} = (y_1, \ldots, y_M)$, where $M \leq N$, is obtained according to the frame-level predictions of the CTC head. For each speech embedding x_n, the most likely token $\hat{v}_n$ from the CTC vocabulary $\mathcal{V}$ is obtained via greedy decoding:

$$\hat{v}_n = \arg\max_{v \in \mathcal{V}} P_{\text{CTC}}(v \mid x_n)$$

An embedding x_n is retained in the pruned sequence $\mathbf{Y}$ if it satisfies the following condition:

$$y_m = x_n \begin{cases} \hat{v}_n \neq \epsilon, \\ \hat{v}_n = \epsilon \ \wedge \ P_{\text{CTC}}(\epsilon \mid x_n) \leq \tau_{\text{ctc}} \end{cases}$$

where ϵ denotes the special '*blank*' token, $\tau_{\text{ctc}} \in [0, 1]$ represents the pre-defined pruning threshold, and the index m increments sequentially for each retained frame. The posterior distribution P_{CTC} is computed by applying a softmax function to the logits produced by the CTC head. The threshold τ_{ctc} can be dynamically tuned to control the efficiency-performance trade-off.

Although highly effective, COBRA is not entirely task-agnostic as it relies explicitly on linguistic content. This makes it ideal for speech-to-text tasks (ASR, ST), but potentially less suited for paralinguistic tasks like emotion recognition. In the

Table 1: Recent research on efficient architectures and optimization methods for contemporary Multimodal LLM.

Ref.	method	approach	models	task
(Benoit et al., 2018; Nagel et al., 2021)	static	quant.	res-net, inception, BERT	image classification, NLU
(Kim and Hassan, 2020)	static	pruning, distill., quant.	roBERTa	NLU
(Winata et al., 2020; Wei et al., 2022)	static	pruning, compres.	in-house	ASR, enhancement
(Kim et al., 2022)	dynamic	pruning	roBERTa	NLU
(Bonnaerens and Dambre, 2023)	dynamic	pruning	Visual Transformer	image classification
(Someki et al., 2025; Hannan et al., 2026)	dynamic	layer-dropping, pruning	OWSM, WavLM	ASR, ST
(Zhou et al., 2020; Wright et al., 2024)	dynamic	early-exit, distillation	BERT, Wav2Vec2, Whisper	NLU, ASR
(Moumen et al., 2026)	dynamic	sparsification	Qwen-2.5, LLAMA-3.1	ASR, ST
(Guo et al., 2025)	dynamic	compres.	LLAMA-3.1	NLU, SQA
(Peng et al., 2026; Liang et al., 2025a)	dynamic	compres. + zero shot	Qwen-2.5, Phi-3.1	NLU, SQA, ASR, ST, emotion
(Moumen et al., 2026)	dynamic	pruning	Qwen-2.5, LLAMA-3.1	ASR, ST
(Zhang et al., 2026; Wen et al., 2025)	dynamic	sparsification	VLLM (LLaVA, QwenVL)	visual QA

latter case, pruning techniques based on cosine similarity (Wen et al., 2025) or average pooling (Sun et al., 2017) may be better suited.

Although pruning can be applied directly to a pre-trained pipeline, in our implementation we **retrain the projector** and fine-tune the LLM. Notably, our framework is entirely complementary to traditional temporal downsampling applied at the projector stage.

4 Experiments and Results

Models. For the speech processing component, we extract the encoder from OpenAI's official pretrained Whisper `large-v3-turbo` model, discarding its native decoder (Radford et al., 2023). Because this architecture does not natively feature a CTC head, we append an auxiliary linear CTC decoder layer on top of it, trained on the clean portion of the LibriSpeech (Panayotov et al., 2015) training set. This CTC head is optimized for the ASR task using a standard English character-based vocabulary. Despite being trained on text targets, this CTC head is utilized exclusively to identify audio embeddings decoded as blank tokens to drive our pruning mechanism, rather than for generating text transcriptions. We consider two distinct configurations for the speech encoder framework: either keeping the underlying Whisper backbone completely frozen or fine-tuning its final four encoder blocks alongside the CTC layer. The motivation for updating the backbone stems from the need to optimize the blank token distribution; as shown in Table 2, fine-tuning the encoder blocks yields a substantially higher percentage of blank tokens (86.28%) compared to the frozen baseline, thereby allowing us to prune the sequence to a greater extent. Additionally, for the ASR task, we also consider Wav2Vec2 (Baevski et al., 2020), which natively provides a pre-trained CTC head trained exclusively on LibriSpeech. As LLM, we considered

Table 2: Comparison of standalone speech encoder configurations on LibriSpeech's test-clean subset.

Model	WER (%)	Blank (%)	# params
Wav2Vec2	**1.80**	64.71	315M
frozen Whisper CTC	36.59	44.95	637M
fine-tuned Whisper CTC	2.75	**86.28**	637M

EuroLLM-1.7B, a multilingual LLM trained on 28 Languages (Martins et al., 2025).

Datasets. We evaluate our proposed method considering 4 tasks: ASR, ST, TD, and ER. For ASR, we used the transcribed portion of VoxPopuli (Wang et al., 2021), which contains parliamentary speeches. Specifically, we employed the English subset within MoSEL (Gaido et al., 2024), comprising approximately 520 hours of training data, 5 hours of development data, and 5 hours of test data. ASR performance was measured using the standard Word Error Rate (WER).

For the ST experiments, we used the EuroParl-ST corpus (Iranzo-Sánchez et al., 2020) in two translation directions: German→English and French→English. Performance is evaluated using the standard normalized BLEU score (Papineni et al., 2002).

For the ER experiments, we used the IEMOCAP dataset (Busso et al., 2008), focusing only on the audio modality. Following standard practice (Ma et al., 2024a), we restrict the task to four core emotion categories:{'happy', 'neutral', 'angry', 'sad'}. Performance is evaluated using Unweighted Accuracy (UWA) via a speaker-independent 5-fold cross-validation protocol. In each fold, eight speakers are utilized for training while the remaining two distinct speakers are reserved for testing, ensuring a rigorous speaker-independent evaluation.

For TD, we use a multilingual subset of the SIB-FLEURS dataset (Adelani et al., 2023) considering: Czech, English, Spanish, Finnish, Galician, and Italian. The train, validation, and test splits for

these languages are merged, resulting in approximately 24 hours of training data, 3.5 hours of validation data, and 8.5 hours of test data. The included topic categories are {'science/technology', 'travel', 'politics', 'sports', 'health', 'entertainment', 'geography'}. Again, the performance is evaluated using the unweighted accuracy (UWA).

Implementation and Training. The projector in Figure 1 consists of two linear layers that map the encoder outputs to the embedding dimension of EuroLLM-1.7B. Following the common practice, the speech embeddings are downsampled before being passed to the projector via concatenation of adjacent frames. Based on previous publications (Ma et al., 2024b), we adopt a downsampling factor of $DS = 5$ at the projector stage. Note that our CTC-pruning is applied before the projector so the final token rate is the result of the combination of the two. We also integrate rank-8 LoRA adapters into the LLM decoder's query and value projections (see Figure 1), configured with an alpha scaling factor of $\alpha = 32$ and a dropout rate of 0.05. We used the Adam optimizer (Kingma and Ba, 2015) with parameters $(\beta 1, \beta 2, \epsilon_{\text{adam}}) = (0.9, 0.999, 1e^{-8})$. We build our codebase on top of the SLAM-LLM framework[2] (Ma et al., 2026). Training follows the SLAM-LLM ASR recipe (Ma et al., 2025b) with dynamic frame batching, where samples are grouped based on audio-frame length using a batch size of 4. A learning rate scheduler with 1,000 linear warm-up steps is used, after which the learning rate is kept constant at 1×10^{-4}. Training runs for a maximum of 10 epochs and is stopped early when the validation loss no longer improves. For all the tasks, we use cross-entropy loss between the LLM-predicted tokens and the ground-truth label tokens. During inference, decoding is performed with a beam-size of 4. In the results reported below, all methods are compared under matched throughput conditions, ensuring a fair evaluation of their efficiency–accuracy trade-offs.

The following task specific prompts are used: for ASR "Transcribe speech to text.", ST "Translate speech to English text.", TD "Classify the topic of the spoken utterance into one of the following labels: science/technology, travel, politics, sports, health, entertainment, geography.", ER "Classify the emotion of the speaker among [neutral, happy, sad, angry]. Answer with exactly one word.".

[2] https://github.com/X-LANCE/SLAM-LLM/

4.1 Comparative baselines

The literature includes a few pruning methods that serve as alternatives to our approach.

Cosine-Distance: (Wen et al., 2025) introduces the concept of *pattern similarity* to quantify redundancy in the speech embeddings $\mathbf{X}$. This criterion computes the cosine distance between adjacent embeddings, retaining a frame only if the distance between consecutive embeddings exceeds a predefined threshold, i.e.:

$$y_m = x_n \iff \left(\frac{x_n \cdot x_{n-1}}{||x_n|| \cdot ||x_{n-1}||} \right) \geq \tau_{\text{cosd}}$$

Similarly to τ_{ctc} in COBRA, τ_{cosd} determines the sequence pruning rate and regulates the corresponding performance-efficiency trade-off.

Average Pooling: Introduced by Sun et al. (2017), this method averages the speech embeddings within a non-overlapping window of length W. To ensure a fair comparison, the window size W is varied to precisely match the resulting token rates of our other configurations, i.e:

$$y_m = \frac{1}{W} \sum_{n=(m-1)\cdot W+1}^{m \cdot W} x_n$$

Random: This method serves as a content-blind control by randomly discarding a designated percentage of speech embeddings while preserving their original temporal order. Mirroring our other baselines, the retention ratio directly dictates the operating token rate and governs the resulting performance-efficiency trade-off.

DS: To vary the token rate, this method adjusts the value of downsampling rate (DS) at the projector stage. Specifically, $DS = 5$ is treated as the baseline configuration where no pruning is applied to the speech encoder output.

4.2 Results on speech to text tasks

Figure 3 reports the ASR performance (WER) on VoxPopuli's test set as a function of token rate, measured in tokens per second (TPS). Note that when no pruning is applied and $DS = 5$, the token rate at the output of the projector is 10 TPS. Consequently, a lower token rate in Figure 3 directly corresponds to an increased degree of pruning.

As shown in the figure, CTC-based pruning (red line) is remarkably robust in comparison to other baselines to the pruning threshold: the WER remains stable across a wide range of token rates,

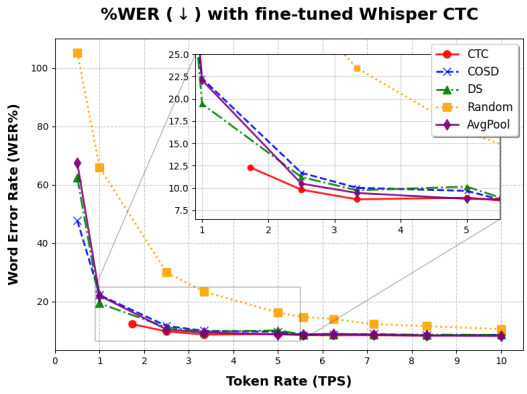

Figure 3: WER (%) vs. token rate on VoxPopuli test set using fine-tuned Whisper CTC encoder in SLLM.

and only begins to increase noticeably when the rate falls to approximately 2.5 TPS (i.e. 75% pruning). Below this point, performance gradually degrades until a token rate of 2 TPS is reached, which corresponds to the removal of almost all blank tokens. Notably, the proposed CTC-based pruning achieves its intrinsic lower bound for the token rate solely by eliminating the blank frames, without any additional lossy compression. In contrast, all other pruning strategies exhibit a consistent degradation in performance once the token rate drops below 5 TPS, which corresponds to discarding more than 50% of the speech embeddings (with the random baseline deteriorating much earlier, as expected). When the token rate falls below 1 TPS, cosine distance based pruning still performs better than both average pooling and DS-based pruning, indicating that it preserves more informative frames under extreme compression.

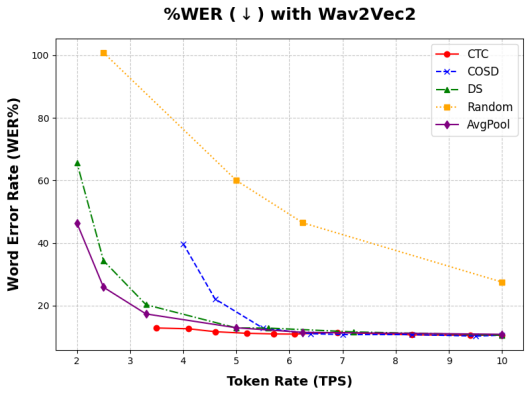

Figure 4: WER (%) vs. token rate on VoxPopuli test set using Wav2Vec2 encoder in SLLM.

To further assess the effectiveness of the proposed approach, we conducted an additional ASR experiment using Wav2Vec2 as the speech encoder. Figure 4 reports the WER obtained on the VoxPopuli evaluation sets. Also in this case, the benefits of pruning blank tokens at different rates are evident: the curve corresponding to CTC-based pruning remains remarkably stable even at low token rates. The behavior of the DS, Random and average-pooling baselines, evaluated with variable window sizes, closely mirrors the trajectories observed in Figure 3, reinforcing the consistency of these findings across the two audio encoders. In this scenario, cosine-distance based pruning performs markedly worse at low token rates, probably because it is more susceptible than CTC based pruning to the mismatch between Wav2Vec2's LibriSpeech training data and the acoustics of the VoxPopuli test set. In addition in Table 2, we saw that our fine-tuned Whisper CTC allows us to prune much more blank tokens than Wav2Vec2 in the case of LibriSpeech's test-clean subset. This trend can also be spotted in Figures 3 and 4, on VoxPopuli's test set. The CTC-based pruning allows for a greater degree of pruning with our fine-tuned Whisper CTC than with Wav2Vec2, reflected by the lower token rate of around 1.7 TPS achieved by the former in Figure 3 against the limit of 3.5 TPS set by the latter in Figure 4.

In summary, CTC-based pruning demonstrates more robust generalization than the other baseline methods, despite its CTC head having been trained on LibriSpeech, whose acoustic conditions differ substantially from those of VoxPopuli.

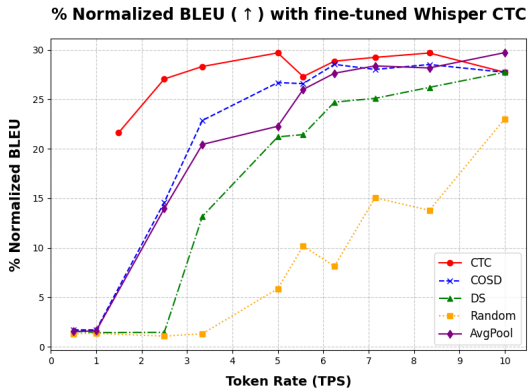

Figure 5: %Normalized BLEU vs. token rate on the 'de2en' and 'fr2en' test subsets of EuroParl-ST using fine-tuned Whisper CTC in SLLM.

Figure 5 reports the performance on the ST task in terms of % normalized BLEU metric. Note that a higher BLEU indicates better performance, in contrast to WER, where lower values are desirable. Also, for the ST task, the advantage of applying

CTC-based pruning clearly stands out compared to the other baselines: the BLEU score remains relatively high and remarkably stable even at its lowest token rate limit ($\approx$ 1.5 TPS).

4.3 Results on Classification tasks

Figures 6 and 7 show TD and ER results using Unweighted Accuracy (UWA), where higher is better. Because these tasks classify whole utterances, pruning affects them differently than ASR and ST. For TD (Figure 6), all pruning methods behave similarly, with little performance loss until reaching the throughput lower bound ($\approx$ 2 TPS). This resilience is rooted in the task's reliance on semantic keyword density rather than fine-grained prosodic variations. Because TD relies primarily on explicit lexical cues to identify key subject-matter words within the utterance, the removal of acoustic blank frames does not alter the underlying semantic content. Performance only degrades at the absolute threshold where semantic keywords themselves begin to be aggressively pruned.

In contrast, for the ER task (Figure 7), discarding speech embeddings classified as blank tokens causes an immediate drop in performance, suggesting that these embeddings carry highly relevant information to paralinguistic classification. The temporal structure of pauses, hesitations, and silences carries vital prosodic markers of a speaker's emotional state. Because our primary pruning mechanism collapses these blank frames to compress the sequence length, the downstream model loses direct access to these duration-based acoustic cues. In this paralinguistic setting, non-selective baseline strategies such as average pooling or random sampling are distinctly preferable, as they uniformly compress the sequence without entirely erasing the temporal footprint of structural silences.

Notably, the Whisper CTC with frozen backbone yields superior baseline performance for both TD and ER, highlighting that a frozen encoder preserves richer global classification features for these tasks compared to the fine-tuned variant.

4.4 Robustness to token-rate mismatch

From Figures 3, 4, and 5, we observe that CTC-based pruning consistently emerges as the most effective strategy for both ASR and ST. While the previous experiments focused exclusively on matched pruning conditions, in the following we examine the robustness of CTC-based pruning under two forms of mismatch: *i)* applying a pruning rate at inference time that differs from the one used during training, and *ii)* training models with variable pruning rates across batches. Figures 8 and 9 show the results for ASR and ST tasks, respectively. In both figures, the upper panel presents a confusion matrix reporting the performance obtained when evaluating the model under different inference-time token rates. The off-diagonal entries correspond to mismatched configurations, i.e., cases in which the token rate used during inference differs from the one used during training. In the bottom part of both figures, we report the results for a model trained with dynamic pruning. In this setting, for each training batch we randomly sample a token-rate from the same set of rates used in the previous experiments, with the goal of making the model robust to variations in the pruning rate at inference time. In both figures, brighter background colors indicate better performance.

From the figures, we observe that both ASR and ST tasks suffer performance degradation under mismatched conditions, with the effect becoming more

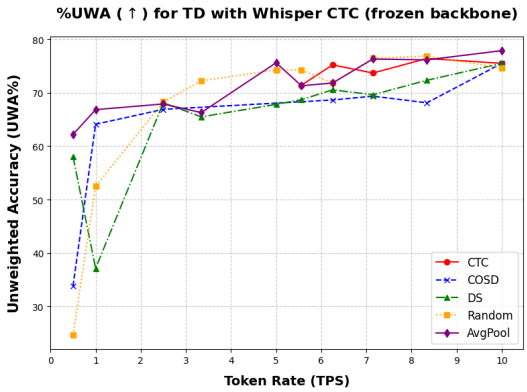

Figure 6: UWA (%) vs. token rate on SIB-Fleurs test set with Whisper CTC (frozen backbone) in SLLM.

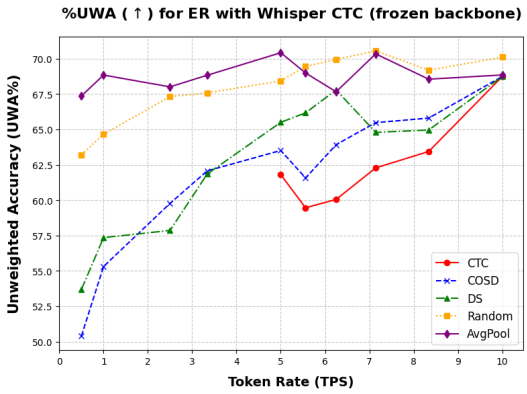

Figure 7: %UWA vs. token rate on IEMOCAP using Whisper CTC (frozen backbone) in SLLM.

pronounced as the mismatch increases. In contrast, this degradation is not observed in the model trained with dynamic pruning, which maintains similarly strong performance across the different pruning rates. This suggests that dynamic pruning is the most effective strategy for achieving flexibility without cutting down on performance.

The ST task is noticeably more sensitive to token-rate mismatches than ASR. Translation depends on broader semantic context and long-range structural patterns, so changes in input token length disrupt the positional mappings learned by the projector layers. ASR, instead, relies on a more monotonic alignment with local phonetic cues, which makes it comparatively more robust to moderate train–inference token-rate deviations.

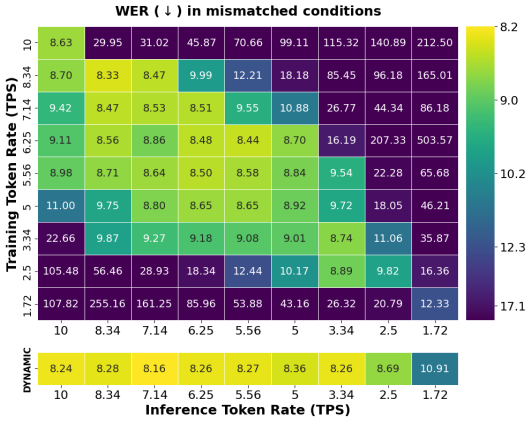

Figure 8: WERs (%) under mismatched token rate conditions and with dynamic pruning on VoxPopuli test set.

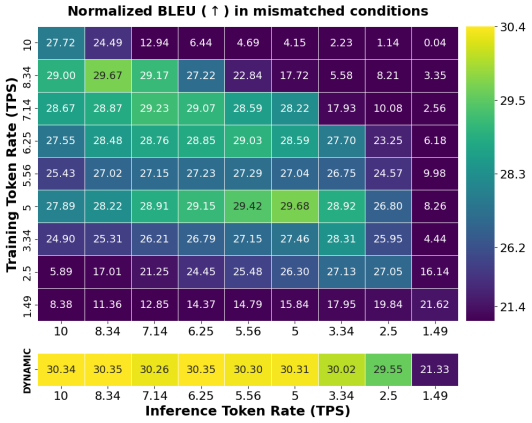

Figure 9: Normalized BLEU under mismatched token rates and with dynamic pruning on EuroParl-ST.

4.5 FLOP reduction

To quantify the efficiency gains of COBRA, we estimate the theoretical floating-point operations

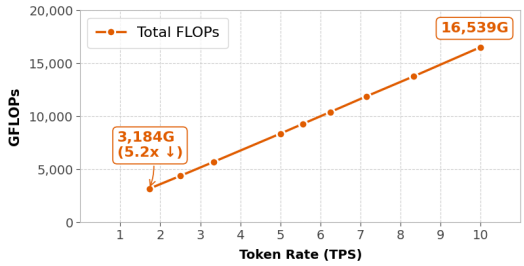

Figure 10: Number of FLOPS vs. token rate.

(FLOPs) of the full pipeline following standard formulations for transformer-based language models (Kaplan et al., 2020; Vaswani et al., 2017; Hoffmann et al., 2022). The total cost reported Figure 10 covers the linear projector and the autoregressive LLM decoding, while encoder FLOPs are excluded as they are identical across all pruning conditions. Since the number of frames surviving pruning scales linearly with the pruning rate, a near-perfect linear relationship holds between pruning rate and FLOPs reduction, with a maximum reduction of 80.7% at an 82.77% pruning rate (from 16538.6 to 3184.1 GFLOPs).

5 Conclusions

We proposed COBRA, a simple and effective CTC-guided token-pruning strategy for Speech-LLMs that removes blank-aligned embeddings before they are sent to the LLM. Experiments across ASR, ST, TD, and ER show that COBRA maintains strong performance even under aggressive compression, with ASR and ST remaining stable at very low token rates. Moreover, training with dynamic pruning enables a single model to operate reliably across multiple token budgets, achieving flexible efficiency–accuracy trade-offs. These properties make COBRA a practical and scalable solution for deployment in resource-constrained settings.

Future research may explore task-adaptive or learned pruning criteria that combine linguistic and paralinguistic cues, integrate COBRA with LLM-side efficiency methods, and extend the approach to multilingual or low-resource settings. Scaling this framework by training our Whisper CTC over massive, diverse corpora could further optimize blank token detection thresholds for complex ASR tasks. Applying COBRA to streaming SLLMs and real-time on-device scenarios is another promising direction.

Limitations

Despite its advantages, COBRA presents several limitations. First, the method is not fully task-agnostic, as it explicitly relies on linguistic information extracted from the CTC head. Tasks that depend on paralinguistic cues, such as emotion recognition, may be negatively affected, since blank-aligned frames can still carry prosodic or affective information. Second, COBRA requires a CTC head trained on ASR, which may not be available for all encoders or languages, and introduces an additional training step. Third, while COBRA significantly compresses the input sequence length to alleviate the LLM's attention overhead, it operates purely at the input stage and does not optimize the LLM's internal architecture, unlike complementary model-side efficiency techniques such as weight sparsification or activation pruning. Finally, pruning decisions are made locally at the frame level, without considering global sequence structure, which may limit performance under extreme compression or in highly noisy conditions.

References

D. I. Adelani, H. Liu, X. Shen, N. Vassilyev, J. O. Alabi, Y. Mao, H. Gao, and A. E. Lee. 2023. Sib-200: A simple, inclusive, and big evaluation dataset for topic classification in 200+ languages and dialects.

S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. 2015. VQA: visual question answering. *arXiv, preprint arXiv:1505.00468*.

A. Baevski, H. Zhou, M. Abdelrahmxundo, and M. Auli. 2020. Wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Proc. of NEURIPS*.

J. Bai, S. Bai, S. Yang, S. Wang, S. Tan, P. Wang, J. Lin, C. Zhou, and J. Zhou. 2024. Qwen-VL: A versatile vision-language model for understanding, localization, text reading, and beyond. In *Proc. of ICCPR*, pages 295–302.

J. Benoit, K. Skirmantas, C. Bo, Z. Menglong, T. Matthew, H. Andrew, A. Hartwig, and K. Dmitry. 2018. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proc. of CVPR*, pages 2704–2713.

M. Bonnaerens and J. Dambre. 2023. Learned thresholds token merging and pruning for vision transformers. *IEEE Trans. on Machine Learning*.

C. Busso, M. Bulut, C-C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. Chang, S. Lee, and S. Narayanan. 2008. Iemocap: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42(4):335–359.

U. Cappellazzo, M. Kim, H. Chen, P. Ma, S. Petridis, D. Falavigna, A. Brutti, and M. Pantic. 2025. Large language models are strong audio-visual speech recognition learners. In *Proc. of ICASSP*, pages 1–5.

L. Chen, H. Zhao, T. Liu, S. Bai, J. Lin, C. Zhou, and B. Chang. 2024. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *Proc. of ECCV*, page 19–35.

Z. Chen, J. Wu, W. Wang, and W. Su et al. 2023. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proc. of CVPR*.

W. Cui, D. Yu, X. Jiao, Z. Meng, G. Zhang, Q. Wang, S. Y. Guo, and I. King. 2025. Recent advances in speech language models: A survey. In *Proc. of ACL*, pages 13943–13970.

W. Dai, J. Li, D. Li, A. Tiong, H. Meng, J. Zhao, W. Wang, B. Li, P. Fung, and S. Hoi. 2023. Instructblip: towards general-purpose vision-language models with instruction tuning. In *Proc. of NEURIPS*.

J-B. Alayrac et al. 2022. Flamingo: a visual language model for few-shot learning. In *Proc. of NEURIPS*.

Y. Fathullah, C. Wu, E. Lakomkin, J. Jia, Y. Shangguan, K. Li, J. Guo, W. Xiong, J. Mahadeokar, O. Kalinli, C. Fuegen, and M. L. Seltzer. 2024. Prompting large language models with speech recognition abilities. In *Proc. of ICASSP*, pages 13351–13355.

S. Fernandez, A. Graves, and J. Schmidhuber. 2007. An application of recurrent neural networks to discriminative keyword spotting. In *Proc. of ICANN*, pages 220—229.

M. Gaido, M. Cettolo, M. Negri, and M. Turchi. 2021. CTC-based compression for direct speech translation. In *Proc. of EACL*, pages 690–696.

M. Gaido, S. Papi, L. Bentivogli, and A. Brutti et al. 2024. MOSEL: 950,000 hours of speech data for open-source speech foundation model training on EU languages. In *Proc. of EMNLP*, pages 13934–13947.

L. Grigoryan, V. Bataev, N. Karpov, A. Andusenko, V. Lavrukhin, and B. Ginsburg. 2025. Flexctc: Gpu-powered ctc beam decodingwith advanced contextual abilities. In *Proc. of ICASSP*, pages 1–7.

S. Guo, S. Zhang, Q. Fang, Z. Ma, M. Zhang, and Y. Deng. 2025. Fastlongspeech: Enhancing large speech-language models for efficient long-speech processing. In *Proc. of NEURIPS*.

A. Hannan, D. Falavigna, S. Nawaz, M. Noman, M. Schedl, and A. Brutti. 2026. Distillation-based layer dropping (dld): effective end-to-end framework for dynamic speech metworks. In *Proc. of ICASSP*.

J. Hoffmann, S. Borgeaud, and A. Mensch et al. 2022. Training compute-optimal large language models. In *Proc. of NEURIPS*.

J. Iranzo-Sánchez, J. A. Silvestre-Cerdà, J. Jorge, N. Roselló, A. Giménez, A. Sanchis, J. Civera, and A. Juan. 2020. Europarl-st: A multilingual corpus for speech translation of parliamentary debates. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8229–8233.

J. Kaplan, S. McCandlish, and T. Henighan et al. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

S. Kim, S. Shen, D. Thorsley, A. Gholami, W. Kwon, J. Hassoun, and K. Keutzer. 2022. Learned token pruning for transformers. In *Proc. of ACM-IGKDD*.

Y.J. Kim and H. Hassan. 2020. FastFormers: Highly efficient transformer models for natural language understanding. In *Proc. of SustaiNLP*, pages 149–158.

D.P. Kingma and J. Ba. 2015. Adam: A method for stochastic optimization. In *Proc. of ICLR*.

Y. Li, H. Jiang, C. Zhang, Q. Wu, X. Luo, S. Ahn, A. H. Abdi, D. Li, J. Gao, Y. Yang, and L. Qiu. 2025. MMInference: Accelerating pre-filling for long-context visual language models via modality-aware permutation sparse attention. In *Proc. of ICML*.

H. Liang, R. Wu, B. Zeng, J. Niu, W. Zhang, and B. Dong. 2025a. Alignformer: Modality matching can achieve better zero-shot instruction-following speech-llm. *IEEE Journal of Selected Topics in Signal Processing*, 19(7):1329–1337.

H. Liang, R. Wu, B. Zeng, J. Niu, W. Zhang, and B. Dong. 2025b. Multimodal reasoning for science: Technical report and 1st place solution to the icml 2025 seephys challenge.

H. Liu, W. Yan, M. Zaharia, and P. Abbeel. 2025. World model on million-length video and language with blockwise ringattention. In *Proc. of ICLR*.

R. Liu, C. Li, T. Ge, T.H. Li, Y. Shan, and G. Li. 2026. Bt-adapter: Video conversation is feasible without video instruction tuning. In *Proc. of CVPR*.

R. Ma, T. Chen, K. Audhkhasi, and B. Ramabhadran. 2025a. Legoslm: Connecting llm with speech encoder using ctc posteriors. In *Proc. of EMNLP 2025*.

Z. Ma, M. Chen, H. Zhang, Z. Zheng, W. Chen, X. Li, J. Ye, X. Chen, and T. Hain. 2024a. Emobox: Multilingual multi-corpus speech emotion recognition toolkit and benchmark. In *Proc. of Interspeech*, pages 1580–1584.

Z. Ma, G. Yang, Y.Yang, Z. Gao, J. Wang, Z. Du, F. Yu, Q. Chen, S. Zheng, S. Zhang, and X. Chen. 2024b. An embarrassingly simple approach for llm with strong asr capacity. *arXiv, preprint arXiv:2402.08846*.

Ziyang Ma, Guanrou Yang, Wenxi Chen, Zhifu Gao, Yexing Du, Xiquan Li, Zhisheng Zheng, Haina Zhu, and 1 others. 2026. Slam-llm: A modular, open-source multimodal large language model framework and best practice for speech, language, audio and music processing. *Proc. IEEE Journal of Selected Topics in Signal Processing*.

Ziyang Ma, Guanrou Yang, Yifan Yang, Zhifu Gao, Jiaming Wang, Zhihao Du, Fan Yu, Qian Chen, Siqi Zheng, Shiliang Zhang, and 1 others. 2025b. Speech recognition meets large language model: Benchmarking, models, and exploration. *Proc. AAAI*.

P.H. Martins, P. Fernandes, J. Alves, N.M. Guerreiro, R. Rei, D.M. Alves, J. Pombal, A. Farajian, M. Faysse, M. Klimaszewski, P. Colombo, B. Haddow, J.G.C. de Souza, A. Birch, and A.F.T. Martins. 2025. Eurollm: Multilingual language models for europe. *Procedia Computer Science*, 255:53–62.

A. Moumen, G. Sun, and P. C. Woodland. 2026. Measuring the redundancy of decoder layers in speechllms. *arXiv, preprint arXiv:2603.05121*.

M. Nagel, M. Fournarakis, R. A. Amjad, Y. Bondarenko, M. van Baalen, and T. Blankevoort. 2021. A white paper on neural network quantization. *arXiv, preprint arXiv:2106.08295*.

V. Panayotov, G. Chen, D. Povey, and S. Khudanpur. 2015. Librispeech: An ASR corpus based on public domain audio books. In *Proc. of ICASSP*, pages 5206–5210.

K. Papineni, S. Roukos, T. Ward, and W-J. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proc. of ACL*, pages 311–318.

J. Peng, Y. Yang, L. Yi, X. Xu, Y., Q. Tang, Y. Fang, J. Li, and K. Yu. 2026. Tasu: Text-only alignment for speech understanding. In *Proc. of ICASSP*, pages 16287–16291.

A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *Proc. of ICML*.

S. Sannigrahi, G. Attanasio, and A.F.T. Martins. 2026. Adats: Adaptive token sampling for efficient speech language models. In *Proc. of ICLR*.

F. D. Schmidt, I. Vulić, G. Glavaš, and D. I. Adelani. 2025. Fleurs-slu: A massively multilingual benchmark for spoken language understanding.

M. Someki, Y. Peng, S. Arora, S. Watanabe, M. Muller, T. Mouchtaris, G. Strimel, and J. Liu. 2025. Context-aware dynamic pruning for speech foundation models. In *Proc. of ICLR*, Singapore.

M. Sun, Z. Song, X. Jjang, J. Pan, and Y. Pang. 2017. Learning pooling for convolutional neural network. *Neurocomputing*, 224:96–104.

Ashish Vaswani, Noam Shazeer, Niki Parmar, and 1 others. 2017. Attention is all you need. In *In Proc. of NEURIPS*, pages 6000–6010.

C. Wang, M. Riviere, A. Lee, and A. Wu et al. 2021. VoxPopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation. In *Proc. of ACL*.

Z. Wei, L. Hao, and X. Zhang. 2022. Model compression by iterative pruning with knowledge distillation and its application to speech enhancement. In *Proc. of Interspeech*, pages 941–945.

Z. Wen, Y. Gao, W. Li, C. He, and L. Zhang. 2025. Token pruning in multimodal large language models: Are we solving the right problem? In *Proc. of ACL*, pages 15537—15549.

G. I. Winata, S. Cahyawijaya, Z. Lin, Z. Liu, and P. Fung. 2020. Lightweight and efficient end-to-end speech recognition using low-rank transformer. In *Proc. of ICASSP*, pages 6144–6148.

G. A. Wright, U. Cappellazzo, S. Zaiem, D. Raj L. O. Yang, D. Falavigna, M. A. Nabih, and A. Brutti. 2024. Training early-exit architectures for automatic speech recognition: Fine-tuning pre-trained models or training from scratch. In *Proc. of ICASSPW*, pages 685–689. IEEE.

J. Wu, W. Gan, Z. Chen, S. Wan, and P. S. Yu. 2023. Multimodal large language models: A survey. In *Proc. of IEEE Int. Conference on Big Data*, pages 2247–2256.

X. Ye, , Y. Gan, X. Huang, , Y. Ge, and Y. Tang. 2025. Voco-llama: Towards vision compression with large language models. In *Proc. of CVPR*, pages 29836–29846.

K. Yosuke, A. Siddhant, and F. Hayato et al. 2023. Tensor decomposition for minimization of E2E SLU model toward on-device processing. In *Proc. of Interspeech*, pages 710–714.

Y. Zhang, C.-K. Fan, J. Ma, W. Zheng, T. Huang, K. Cheng, D. A. Gudovskiy, T. Okuno, Y. Nakata, K. Keutzer, and S. Zhang. 2026. SparseVLM: Visual token sparsification for efficient vision language models inference. In *Proc. of ICML*.

W. Zhou, C. Xu, T. Ge, J. McAuley, K. Xu, and F. Wei. 2020. BERT loses patience: Fast and robust inference with early exit. In *Advances in Neural Information Processing Systems*, volume 33, pages 18330–18341.

A Licenses

Table 3 outlines all the licenses of resources used in this work. The use of all artifacts was consistent with their intended use.

Resource	License
VoxPopuli (Wang et al., 2021)	CC0 1.0
SIB-Fleurs (Adelani et al., 2023; Schmidt et al., 2025)	CC BY-SA 4.0
Iemocap (Busso et al., 2008)	Signed agreement: research only, no redistribution
Europarl-ST (Iranzo-Sánchez et al., 2020)	CC BY-NC 4.0
Whisper-large-v3-turbo (Radford et al., 2023)	MIT
Wav2Vec2 (Baevski et al., 2020)	MIT
EuroLLM-1.7B (Martins et al., 2025)	Apache 2.0
SLAM-LLM (Ma et al., 2026)	MIT
SLAM-ASR (Ma et al., 2025b)	MIT

Table 3: Licenses of datasets, models, and frameworks used in this work.