Adaptive Audio-Visual Speech Recognition via Matryoshka-Based Multimodal LLMs

Anonymous ACL submission

Abstract

Audio-Visual Speech Recognition (AVSR) leverages both audio and visual modalities to enhance speech recognition robustness, particularly in noisy environments. Recent advancements in Large Language Models (LLMs) have demonstrated their effectiveness in speech recognition, including AVSR. However, due to the significant length of speech representations, direct integration with LLMs imposes substantial computational costs. Prior approaches address this by compressing speech representations before feeding them into LLMs. However, higher compression ratios often lead to performance degradation, necessitating a trade-off between computational efficiency and recognition accuracy. To address this challenge, we propose Llama-MTSK, the first Matryoshka-based Multimodal LLM for AVSR, which enables flexible adaptation of the audio-visual token allocation based on specific computational constraints while preserving high performance. Our approach, inspired by Matryoshka Representation Learning, encodes audio-visual representations at multiple granularities within a single model, eliminating the need to train separate models for different compression levels. Moreover, to efficiently fine-tune the LLM, we introduce three LoRA-based Matryoshka strategies using global and scale-specific LoRA modules. Extensive evaluations on the two largest AVSR datasets demonstrate that Llama-MTSK achieves state-of-the-art results, matching or surpassing models trained independently at fixed compression levels.

1 Introduction

Audio-Visual Speech Recognition (AVSR) aims to improve the robustness of speech recognition systems by utilizing both audio and visual signals to recognize human speech. The correlation between audio and lip movements enables the model to focus on relevant speech content while discarding ambient or background noise. With the rising demand for robust speech recognition systems and the widespread availability of cameras (e.g., smartphones), numerous studies have explored advancements in AVSR technology. They have investigated different neural architectures (Dupont and Luettin, 2000; Noda et al., 2015; Afouras et al., 2018a; Petridis et al., 2018; Ma et al., 2021; Hong et al., 2022), training methods (Ma et al., 2023a; Hong et al., 2023), and methods using self-supervised pretraining (Shi et al., 2022; Haliassos et al., 2023, 2024b; Hsu and Shi, 2022; Haliassos et al., 2024a).

Recently, with the growing popularity and versatility of Large Language Models (LLMs), new efforts have emerged to connect LLMs with speech modeling (Lakhotia et al., 2021; Huang et al., 2024; Park et al., 2024). Specifically, in Auditory Speech Recognition (ASR) and Visual Speech Recognition (VSR), researchers have demonstrated the possibility and effectiveness of LLMs in speech recognition (Chen et al., 2024; Hu et al., 2024b; Ma et al., 2024; Yu et al., 2024; Fathullah et al., 2024; Fang et al., 2024a; Lu et al., 2025; Tan et al., 2024; Yeo et al., 2024). By employing multi-modal speech information, a recent work proposes to adapt LLMs in AVSR as well (Llama-AVSR), attaining state-of-the-art recognition performances (Cappellazzo et al., 2025). A common focus of prior works is reducing the sequence length of speech representations before feeding them into the LLM. Since LLMs have a large number of parameters and speech sequences are much longer than text, directly using speech representations imposes a significant computational burden. At the same time, (Cappellazzo et al., 2025) demonstrate that there is a trade-off between how much we compress the audio-visual speech representations and performance: while higher compression ratios enhance computational efficiency, they inevitably lead to a degradation in performance. Therefore, a possible solution is training and distributing different models with compression ratios tailored to individual

users' computational resources.

However, retraining existing models for different compression ratios, each requiring a distinct coarse-to-fine granularity, is time-consuming and impractical. For this reason, we propose to exploit the concept of Matryoshka Representation Learning (MRL) (Kusupati et al., 2022; Kudugunta et al., 2024; Nair et al., 2025) to encode audio-visual information at different granularities using a single model. This concept was recently explored in visual-linguistic understanding and reasoning tasks in (Cai et al., 2024; Hu et al., 2024a), demonstrating that Matryoshka-based large vision-language models can support multi-granular visual processing at inference while achieving performance parity with independently trained models for each compression rate.

For our audio-visual setting, *with the aspiration to flexibly decide between computational efficiency and performance at inference time within the same model*, we propose Llama-Matryoshka (abbreviated as Llama-MTSK in the rest of the paper), a Matryoshka-based Multimodal LLM which caters to different demands based on specific requirements by training simultaneously audio-visual representations of different granularities. Llama-MTSK first produces audio and video tokens using pre-trained encoders, then reduces their length using average pooling or stacking compression methods at multiple compression rates. Then, unlike the previous works using MRL that *directly fine-tune all the LLM's parameters* (Cai et al., 2024; Hu et al., 2024a), we propose three LoRA-based Matryoshka approaches (LoRA 🪆) to *parameter-efficiently fine-tune* the LLM (i.e., Llama (Dubey et al., 2024)), which is responsible to generate the transcriptions given the audio-visual tokens and textual prompt. These approaches either employ a single global LoRA to learn audio-visual feature tokens at multiple scales (`Multi-Scale` LoRA 🪆), or define multiple LoRAs, each of them focusing on scale-specific audio-visual information (`Scale-Specific` LoRA 🪆), or a combination of both (`Multi-Scale-Specific` LoRA 🪆). At inference, only the projector and LoRA modules associated with the desired compression rate are activated, ensuring both flexibility and efficiency. Our comprehensive experiments on the two largest AVSR datasets demonstrate that our three proposed methods achieve comparable or better performance than training separate models for each combination of audio-video compression rates. Overall, Llama-MTSK *exhibits strong performance results, elastic inference, and computational efficiency under a single set of weights*.

Our key contributions are as follows:

- We propose Llama-MTSK, the first Matryoshka-based Multimodal LLM designed for audio-visual speech recognition. By processing audio-visual tokens with multiple compression levels and granularities, and introducing three Matryoshka-based LoRA modules to efficiently fine-tune the pre-trained LLM, Llama-MTSK is able to dynamically adjust the number of tokens processed during inference using a single model, adapting to varying computational resources or desired accuracy levels.

- Llama-MTSK achieves state-of-the-art results on LRS2 and LRS3, the two largest AVSR datasets, consistently exceeding the performance of models independently trained at specific compression levels. This trend is observed for the ASR, VSR, and AVSR tasks, across both the evaluated compression techniques and granularities.

2 Llama-MTSK

The objective of Llama-MTSK is to train an LLM (Llama-based in our setting) that captures audio and visual information at multiple scales, from coarse to fine, thus providing control over the audio-visual granularity during inference. Consequently, a *single* "universal" model allows us to dynamically adjust the performance-efficiency trade-off at inference time, according to specific needs (Cai et al., 2024; Hu et al., 2024a).

Llama-MTSK follows the structure of Llama-AVSR (Cappellazzo et al., 2025), the first Multimodal LLM (MLLM) tailored for audio-visual speech recognition, with ad-hoc modifications to support MRL (Kusupati et al., 2022). Llama-MTSK computes audio and video tokens via modality-specific pre-trained encoders, and then input them as prefix tokens to the LLM (together with the textual tokens). This approach, denoted as decoder-only, is adopted by several architectures due to its versatility and flexibility (Liu et al., 2023; Lin et al., 2024; Fang et al., 2024b; Fan et al., 2024; Zong et al., 2024; Zhang et al., 2025b; Lee et al., 2024; Fini et al., 2024; Li et al., 2024; Tong et al., 2024; Yao et al., 2024).

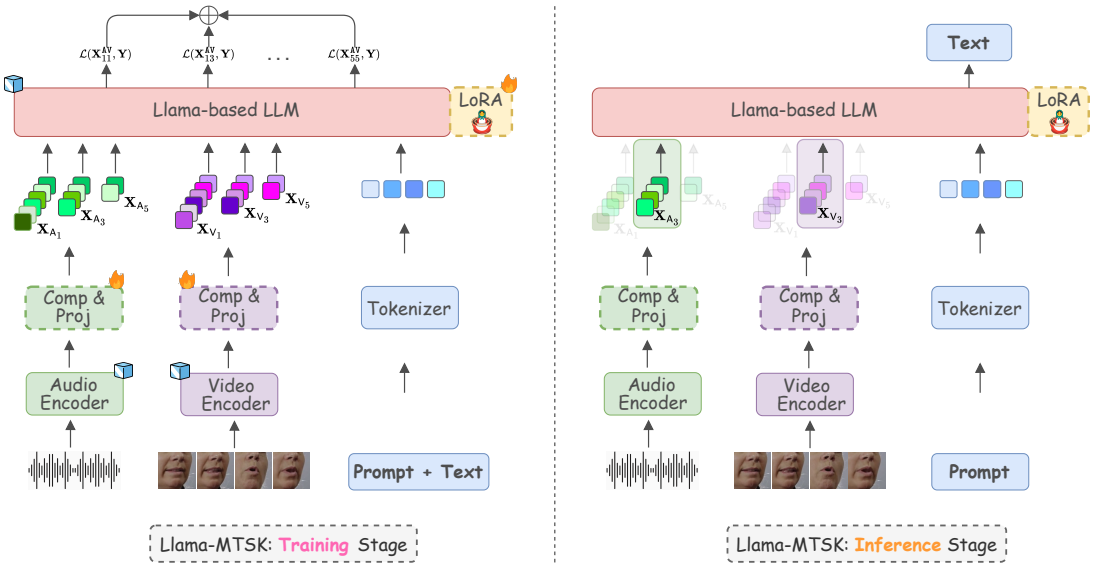

Figure 1: Training and inference stages for Llama-MTSK. (*Left*) During training, we produce audio-visual tokens via pre-trained encoders, followed by specific-scale compression and projection modules. Then, we feed the concatenated audio-visual tokens at multiple scales to the pre-trained Llama-based LLM, which is adapted through one of the three proposed LoRA approaches following the Matryoshka Representation Learning principle. (*Right*) At inference, Llama-MTSK allows us to change on-the-fly the audio-visual compression rates for each input data conditioned on our specific requirements using the same model architecture and weights, enabling high flexibility. Furthermore, only one projector and one LoRA module are activated at inference (in this figure, those associated with the audio and video compression rates equal to 3), guaranteeing model's scalability in training and no extra cost in inference. and represent whether the parameters are trained or kept frozen, respectively.

Llama-MTSK consists of three main components: **1)** pre-trained audio and video encoders, **2)** audio and video compression and projection modules, and **3)** an LLM which is parameter-efficiently fine-tuned via ad-hoc LoRA-based strategies (i.e., LoRA).

2.1 Audio/Video Pre-Trained Encoders

We use pre-trained audio and video encoders to project the input audio and video data into two sets of audio and video tokens. We denote with $\mathbf{X}^{\mathsf{A}} \in \mathbb{R}^{N_{\mathsf{A}} \times d_{\mathsf{A}}}$ and $\mathbf{X}^{\mathsf{V}} \in \mathbb{R}^{N_{\mathsf{V}} \times d_{\mathsf{V}}}$ the audio and video token sequences, respectively, where $N_{\mathsf{A}}/N_{\mathsf{V}}$ is the number of audio/video tokens, and $d_{\mathsf{A}}/d_{\mathsf{V}}$ is the audio/video token dimension. The pre-trained encoders are maintained *frozen* during the training stage (in Figure 1).

2.2 Audio-Visual Compression and Projection

Since the dimensions of audio and video tokens often differ from that of the textual tokens, MLLMs include a projection layer that maps audio and video tokens into the LLM embedding space. It is common to employ either linear projectors (Liu et al., 2023; Luo et al., 2024; Yao et al., 2024; Li et al., 2024; Liu et al., 2024b; Zhang et al., 2025a) or abstractors (e.g., Q-Former, resampler) (Zhu et al., 2023; Li et al., 2023; Cha et al., 2024). In our setting, following (Cappellazzo et al., 2025), we use a two-layer MLP projector.

In addition to this, since the LLM predominantly accounts for the entire computation and memory consumption of the MLLM, it is customary to compress the number of multimodal tokens (in our case audio-visual tokens) by a specific factor in order to find the optimal balance in terms of efficiency and accuracy. For example, (Cappellazzo et al., 2025; Fang et al., 2024a; Ma et al., 2024; Fathullah et al., 2024) stack multiple consecutive tokens along the token hidden dimension to reduce the number of tokens, whereas other methods rely on the Q-Former architecture (Li et al., 2023) using a fixed number of query tokens (Tang et al., 2023; Yu et al., 2024; Zhang et al., 2025b; Cha et al., 2024). However, all these methods need to decide the compression rate to apply beforehand, which means they generate outputs of a single, predetermined length, lacking the ability to modulate the final sequence length. This constraint limits the ability to balance information density and computational efficiency, particularly in resource-constrained deployment scenarios. Alternatively, one could train a separate model for

each desired compression rate, but this approach can be time-consuming and cumbersome in practice.

In contrast, we propose to compress the audio and video tokens using multiple compression rates, leading to token sequences at multiple scales, and thus different granularities. We explore two different compression methods to reduce the token sequence length: 1) *average pooling*, and 2) *hidden size stacking*, where multiple consecutive frames are stacked along the token hidden dimension. Therefore, we decide beforehand a range of $\mathtt{G}$ audio compression rates $\{a_1, a_2, \cdots, a_\mathtt{G}\}$ and T video compression rates $\{v_1, v_2, \cdots, v_\mathtt{T}\}$. We gradually increase the compression rates (i.e., $a_{i+1} > a_i$, $i = 1, \cdots, \mathtt{G}$). With a_i we refer both to the compression rate and the corresponding scale interchangeably (e.g., if $a_i = 4$, then the corresponding sequence would have $\lfloor \frac{N_\mathtt{A}}{4} \rfloor$ tokens). We then compress the audio and video tokens using the chosen rates, producing token sequences at multiple scales: $[\mathbf{X}^\mathtt{A}_{a_1}, \mathbf{X}^\mathtt{A}_{a_2}, \cdots, \mathbf{X}^\mathtt{A}_{a_\mathtt{G}}]$ and $[\mathbf{X}^\mathtt{V}_{v_1}, \mathbf{X}^\mathtt{V}_{v_2}, \cdots, \mathbf{X}^\mathtt{V}_{v_\mathtt{T}}]$.

At this point, each of these sequences are processed by compression rate-specific linear projectors to align the audio-visual and text tokens (see Figure 1).

2.3 LLM Adaptation via LoRA

The LLM is responsible for generating the corresponding ASR transcription in an auto-regressive fashion given the audio, video, and textual tokens. We define $\mathbf{X}^\mathtt{AV}_{ij}$ as the concatenation of audio and video tokens with audio and video compression rates of a_i and v_j, and the prompt textual tokens $\mathbf{X}^P$: $\mathbf{X}^\mathtt{AV}_{ij} = [\mathbf{X}^\mathtt{A}_{a_i}, \mathbf{X}^\mathtt{V}_{v_j}, \mathbf{X}^P]$. To *parameter-efficiently* align the LLM with the multimodal inputs, we use LoRA modules (Hu et al., 2021) to adapt the query and value projection matrices of each layer. In our setting, the LLM is trained on multiple audio-visual tokens with different scales. We investigate *three* different strategies to efficiently fine-tune LLM's pre-trained matrices via LoRA approximation under a MRL setting: **1)** `Multi-Scale` LoRA Matryoshka (`MS` LoRA), **2)** `Specific-Scale` LoRA Matryoshka (`SS` LoRA), and **3)** `Multi-Specific-Scale` LoRA Matryoshka (`MSS` LoRA). These three methods are illustrated in detail in Figure 2.

The `MS` LoRA approach uses a single "*global*" LoRA to approximate the query and value projection matrices of each LLM's self-attention layer, regardless of the chosen scale and shared by all the input token sequences. For a pre-trained weight matrix W, the projection output is computed as follows:

$$\mathbf{H}^\mathtt{AV}_{ij} \leftarrow \mathbf{X}^\mathtt{AV}_{ij} W + s \cdot \mathbf{X}^\mathtt{AV}_{ij} W_\mathtt{MS}, \quad (1)$$

where s is a tunable scalar hyperparameter, $W_{MS} = W^{down}_\mathtt{MS} W^{up}_\mathtt{MS}$, $W^{down}_\mathtt{MS} \in \mathbb{R}^{d \times r}$ and $W^{up}_\mathtt{MS} \in \mathbb{R}^{r \times d}$, and $r \ll d$ (r is the bottleneck dimension).

In contraposition to `MS` LoRA, we propose to learn "*expert*" LoRA modules, which specialize to each scale. We call this approach `Specific-Scale` (`SS`) LoRA. Therefore, we define $\mathtt{G} \cdot \mathtt{T}$ LoRA modules, one for each audio-visual scale. We compute the projection output as follows:

$$\mathbf{H}^\mathtt{AV}_{ij} \leftarrow \mathbf{X}^\mathtt{AV}_{ij} W + s \cdot \mathbf{X}^\mathtt{AV}_{ij} W^{ij}_\mathtt{SS}, \quad (2)$$

where $W^{ij}_\mathtt{SS}$ is the LoRA decomposition matrix defined for the i-th audio scale and j-th video scale, and it is defined as W_{MS}. As we explain in subsection 2.4, while all the LoRA modules are used during the training stage, at inference we only activate one LoRA module, corresponding to the selected audio and video scales.

The third approach, `MSS` LoRA, is a hybrid approach between `MS` and `SS`, which aims to learn both scale-specific and multi-scale audio-visual representations. Consequently, we define both a multi-scale global LoRA module, which is always activated and shared among all the input sequences both at training and at inference, and multiple scale-specific LoRA modules. In this case, the output takes the following form:

$$\mathbf{H}^\mathtt{AV}_{ij} \leftarrow \mathbf{X}^\mathtt{AV}_{ij} W + s \cdot \mathbf{X}^\mathtt{AV}_{ij} W^{ij}_\mathtt{SS} + s \cdot \mathbf{X}^\mathtt{AV}_{ij} W_\mathtt{MS}. \quad (3)$$

Regardless of the LoRA fine-tuning approach we employ, Llama-MTSK is trained by averaging the auto-regressive next token prediction loss for each audio-visual scale ij for each input data. The LLM predicts the response $\mathbf{Y} = \{y_l\}_{l=1}^L$ conditioned on the multimodal input tokens, where L represents the number of tokens of the ground truth transcription to be generated. Accordingly, for each Matryoshka audio-visual representation $\mathbf{X}^\mathtt{AV}_{ij}$, the probability of the target $\mathbf{Y}$ is computed by:

$$p(\mathbf{Y}|\mathbf{X}^\mathtt{AV}_{ij}) = \prod_{l=1}^{L} p_\theta(y_l|\mathbf{X}^\mathtt{AV}_{ij}, y_{<l}), \quad (4)$$

where $y_{<l}$ is the generated output sequence up to token $l-1$, and θ is the trainable parameters, which

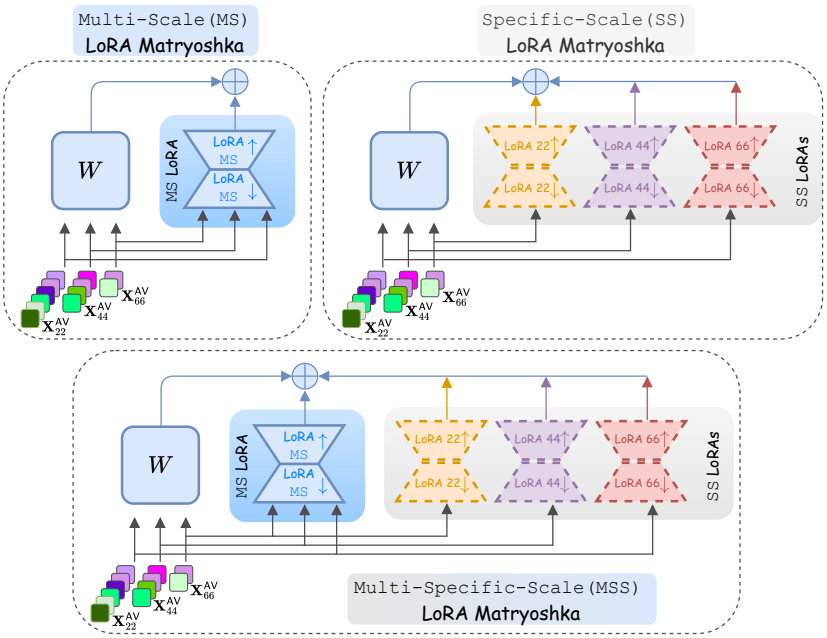

Figure 2: Our three proposed LoRA Matryoshka (LoRA 🪆) approaches. `Multi-Scale` (`MS`) LoRA 🪆 uses a shared global LoRA module for all the audio-visual token scales (in this specific example there are three scales) to fine-tune the pre-trained matrices of the LLM. The `Specific-Scale` (`SS`) variant defines a LoRA module tailored to each scale, learning and specializing to a specific scale. The third approach, `Multi-Specific-Scale` (`MSS`), combines `MS` and `SS` to support both global and specific-scale LoRAs. The global LoRA is responsible to capture relationships that can be shared among different-scale tokens, while specific-scale LoRAs learn tokens based on the specific scale.

comprises the projection layers and the LoRA 🪆 modules according to the LoRA 🪆 fine-tuning approach used.

The final objective is the average over all the audio-visual token scales:

$$\frac{1}{\mathsf{G}\cdot\mathsf{T}}\sum_{i=1}^{\mathsf{G}}\sum_{j=1}^{\mathsf{T}} -\log p(\mathbf{Y}|\mathbf{X}_{ij}^{\mathsf{AV}}). \quad (5)$$

2.4 Llama-MTSK: Training vs Inference

During training, Llama-MTSK learns multiple sets of audio-visual tokens, each progressively incorporating more details as the scale increases. To do so, the LLM processes all the multi-scale audio-visual tokens and concurrently optimize over them using Eq. 5. This means that all the projectors and LoRA 🪆 modules are involved. Instead, at inference time, for each input data, we choose a specific audio-visual scale and we activate only the projector and LoRA module associated with it. This is equivalent to one single Llama-AVSR model trained on the specific scale. This principle is similar to the behaviour of Mixture of Experts-based models (Shazeer et al., 2017; Fedus et al., 2022; Zoph et al., 2022; Mustafa et al., 2022; Puigcerver et al., 2023; Cappellazzo et al., 2024a; Jiang et al., 2024; Muennighoff et al., 2024), which at inference time only activate a small subset of the available experts (in our case the "experts" are the projectors and LoRA 🪆 modules). Figure 1 depicts a schematic comparison of Llama-MTSK training and inference processes.

3 Experiments and Results

3.1 Implementation Details

Datasets. We train and evaluate Llama-MTSK on LRS2 (Son Chung et al., 2017) and LRS3 (Afouras et al., 2018b), the two largest publicly available datasets for audio-visual speech recognition. LRS2 includes 225 hours of video clips from BBC programs. LRS3 contains 433 hours of transcribed English video clips from TED talks.

Pre-Processing. We follow (Ma et al., 2023b; Cappellazzo et al., 2025) for the pre-processing of the datasets. For the video modality, we crop the mouth region of interests (ROIs) through a bounding box of 96×96. Each frame is normalised by subtracting the mean and dividing by the standard deviation of the training set. Audio data only undergo z-normalisation per utterance.

Tasks. The AVSR task is studied for the main results, both for LRS2 and LRS3. We also report the results for the ASR and VSR tasks on LRS3.

Table 1: Comparison between Llama-AVSR and our proposed Llama 🪆 MS, SS, and MSS approaches on LRS2 and LRS3 benchmarks. †Llama-AVSR trains 4 independent models tailored to each configuration of audio-video compression rates.

Method	Compression Rates (A,V)			
	(4,2)	(4,5)	(16,2)	(16,5)
LRS3 Dataset				
Llama-AVSR†	2.4	2.8	3.3	4.1
Llama 🪆 MS	2.6	2.7	3.7	4.1
Llama 🪆 SS	**2.3**	**2.2**	3.3	3.6
Llama 🪆 MSS	2.4	2.4	**3.2**	**3.5**
LRS2 Dataset				
Llama-AVSR	4.1	**4.5**	5.3	8.1
Llama 🪆 MS	4.8	5.9	6.4	8.9
Llama 🪆 SS	**3.4**	4.7	**4.8**	**6.4**
Llama 🪆 MSS	3.6	4.8	6.1	9.0

Table 2: Comparison between Llama 🪆 and multiple SOTA methods on the LRS2 and LRS3 benchmarks. The "**Lab. Hrs.**" column with values X/Y specifies how many labeled hours have been used in training for LRS2 (X) and LRS3 (Y).

Method	Rates (A,V)	Lab. Hrs.	Dataset	
			LRS2	LRS3
CM-seq2seq	(1,1)	380/433	3.7	2.3
Eff. Conf.	(1,1)	818/818	2.3	1.8
auto-avsr	(1,1)	3448/1902	1.5	1.0
W-Flamingo	(1,1)	1982/433	**1.4**	1.1
USR	(1,1)	1982/1759	1.9	1.1
Llama-AVSR	(4,2)	223/433	2.4	**0.9**
Llama 🪆 MS	(4,2)	223/433	**2.1**	1.0
Llama 🪆 SS	(4,2)	223/433	2.4	**0.9**
Llama 🪆 MSS	(4,2)	223/433	2.4	1.2

Llama-MTSK Details. We use Whisper Small and Medium (Radford et al., 2023) as pre-trained audio encoder, whilst AV-HuBERT Large (Shi et al., 2022) for computing the video tokens. Their weights remain frozen throughout the training phase. The projectors consist of two linear layers with ReLU activation in between. As for the LLM, based on the task and dataset, we experiment with 3 base pre-trained models of varying size from the Llama 3 family (Dubey et al., 2024): Llama 3.1-8B, Llama 3.2-3B, and Llama 3.2-1B. Each LoRA module used to fine-tune the query and key projection matrices of each LLM's self-attention layer has a bottleneck dimension r such that the original LLM's hidden size is reduced of a factor 32 for Llama 3.2-3B and 3.2-1B, and 64 for Llama 3.1-8B (e.g., for Llama 3.2-1B, since the hidden size is 2048, the rank is set to $2048/32 = 64$). The hyperparameter s is set to $\frac{1}{8}$.

Audio-Visual Token Compression Rates. We choose the audio and video compression rates to train and evaluate Llama-MTSK carefully, based on the studied tasks. For ASR, we apply compression rates in the range of $\{4, 8, 12, 16, 20\}$. For VSR, since the task is more challenging, we can afford smaller rates: $\{1, 2, 3, 4, 5\}$ (we also include the case in which no compression is applied). For AVSR, we apply audio rates in $\{4, 16\}$ and video rates in $\{2, 5\}$, leading to 4 audio-visual configurations. To compress the audio and video tokens, either we apply average pooling with kernel size and stride equal to the desired compression rate, or we stack consecutive frames along the hidden dimension according to the rate (we denote this as "stacking").

Training/Inference Details. Following (Cappellazzo et al., 2025; Ma et al., 2023b), we augment visual inputs through horizontal flipping, random cropping, and adaptive time masking, while for audio we only apply adaptive time masking. For training, we sample babble noise from the NOISEX dataset (Varga, 1992) using a uniform distribution. We define the textual prompts as in (Cappellazzo et al., 2025): "`Transcribe {task_prompt} to text.`", where `task_prompt` $\in$ {"`speech`", "`video`", "`speech and video`"}. We train our model for 10 epochs with the AdamW optimizer with cosine annealing scheduler and weight decay set to 0.1 using NVIDIA A40 GPUs. The learning rate is set to 1e-3 for ASR and AVSR tasks, and 5e-4 for VSR. For decoding, we use beam search with a beam width of 15 and temperature of 0.6. The evaluation metric for all the experiments is the Word Error Rate (WER, %).

3.2 AVSR Main Results

We report the results achieved by Llama-MTSK MS, SS, and MSS on the LRS2 and LRS3 datasets in Table 1. We replace "MTSK" with 🪆 in the tables and in the following sections to simplify the notation. For both datasets, we use Whisper Small as audio encoder. For the LLM, we use Llama 3.2-1B for LRS3 and Llama 3.2-3B for LRS2. The smaller size of the LRS2 dataset necessitates the larger LLM to mitigate higher WERs. We apply

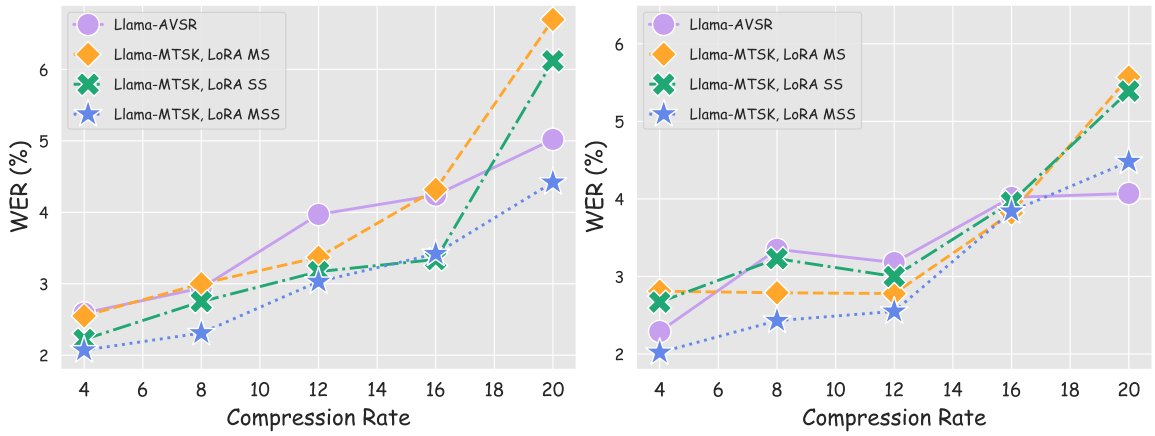

Figure 3: WER results for the *average pooling* (left) and *stacking* (right) compression methods for the ASR task. We use Whisper Small as audio encoder and Llama 3.2-1B as LLM.

audio compression rates of 4 and 16 and video compression rates of 2 and 5, resulting in 4 different compression configurations. We compare these results with those achieved by training Llama-AVSR independently on the 4 configurations, leading to 4 models. During inference, Llama-AVSR employs a separate model trained for each audio-video compression rate. In contrast, our Llama 🪆 uses a single pre-trained model, activating the projector and LoRA 🪆 modules corresponding to the desired compression rate. On the LRS3 dataset, the three proposed Llama 🪆 approaches achieve comparable or superior performance to Llama-AVSR, particularly for the SS and MSS configurations. These two methods use LoRA modules specialized for specific compression rates, which are activated during inference based on specific requirements. On the LRS2 dataset, Llama 🪆 SS outperforms all other approaches across all compression rates.

Llama 🪆 vs SOTA Methods. In Table 2, we compare Llama 🪆 with state-of-the-art (SOTA) methods on LRS2 and LRS3 for the AVSR task. We equip Llama 🪆 with Whisper Medium and Llama 3.1-8B. We report results from 5 recent SOTA AVSR methods: CM-seq2seq (Ma et al., 2021), Efficient Conformer (Burchi and Timofte, 2023), auto-avsr (Ma et al., 2023b), Whisper-Flamingo (Rouditchenko et al., 2024), and USR (Haliassos et al., 2024a). Notably, all these methods do not reduce the token sequence length, whereas Llama-AVSR and Llama 🪆 reduce the number of tokens by a factor 4 for audio and 2 for video. For LRS3, Llama 🪆 achieves SOTA results, with its SS variant surpassing Llama-AVSR, which is trained on those specific compression rates, and outperforming methods like auto-avsr and USR, which use significantly more training hours. For LRS2, Llama 🪆 SS and MSS perform comparably to Llama-AVSR, while MS achieves better results. Additionally, our methods perform as well as or better than CM-seq2seq and Efficient Conformer but slightly underperform other SOTA methods. However, Llama 🪆 is trained only on the 223 hours of LRS2, whereas all competing methods utilize at least 1982 hours. We leave for future work the integration of additional training data to enable a fairer comparison. Finally, more AVSR experiments can be found in the Appendix.

Table 3: Comparison between Llama 🪆 MS and a training-free Llama-AVSR-based approach that reduces the number of tokens via average pooling at inference time for the ASR task on the LRS3 dataset.

Method	Compression Rate				
	2	4	6	8	10
Avg Pooling	4.3	13.5	46.1	89.2	160.0
Llama 🪆 MS	2.5	2.3	2.3	2.7	3.0

3.3 Additional Results

In this section, we extend our analysis to the tasks of ASR and VSR, where only audio or video tokens are fed to the LLM, respectively. We finally present the computational cost analysis of Llama 🪆.

ASR Results. For the ASR task, we consider 5 compression rates in the range {4, 8, 12, 16, 20}. In Figure 3, we report the results on the LRS3 dataset when using average pooling compression (left) and stacking compression (right). With the exception of rate = 20, all the three Llama 🪆 methods outperform separately-trained Llama-AVSR methods. The MSS

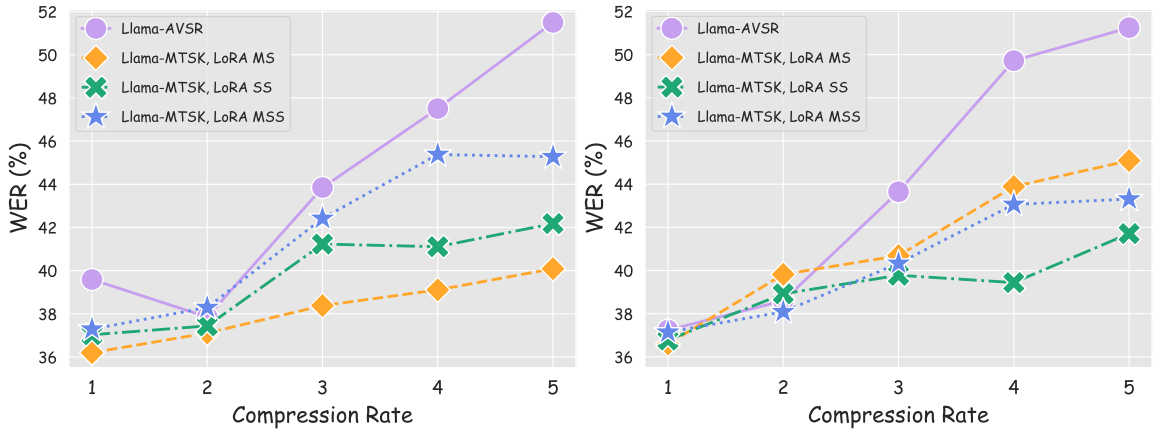

Figure 4: WER results for the *average pooling* (left) and *stacking* (right) compression methods for the VSR task. We use AVHuBERT Large as video encoder and Llama 3.2-3B as LLM.

Table 4: Computational cost analysis of Llama 🪆 MS using different compression rates and Llama 3.1-8B.

(A,V) Rates	# Tokens	TFLOPs
(1,1)	757	11.40
(4,2)	257	3.87
(4,5)	182	2.74
(16,2)	163	2.46
(16,5)	88	1.33

configuration achieves the best WER performance across all the compression rates, even surpassing or equaling the performance of Llama-AVSR trained at the lowest compression rate of 20.

VSR Results. Figure 4 shows WER results for the VSR task, similar to the ASR results in Figure 3. The video rates are {1, 2, 3, 4, 5}, lower than the ASR rates due to the greater complexity of VSR. For both average pooling and stack compression, all three Llama 🪆 approaches outperform Llama-AVSR, with increasing gains at higher rates. The MS and SS approaches using average pooling achieve WER reductions exceeding 10 at the highest rates. We attribute this improvement at higher compression rates to the joint training of multi-scale tokens. The performance of the three LoRA 🪆 approaches varies slightly depending on the compression method, suggesting that no single approach is superior across all configurations. However, all of them significantly outperform Llama-AVSR.

Llama 🪆 vs Avg Pooling at Inference Time. Llama 🪆 trains a single model that supports multiple scales at inference time by applying different compression rates. We compare our method with a training-free approach that trains a single Llama-AVSR model without compression and then applies the desired compression rate at inference on-the-fly by average pooling the tokens. In Table 3, we study the ASR setting with audio compression rates in the range {2, 4, 6, 8, 10}. The performance of the average-pooling baseline is severely impacted by a decrease in the number of tokens, while Llama 🪆 MS is much more robust. These results demonstrate that Llama 🪆 MS can be effectively used with diverse computational resources. Notably, even with limited resources, a compression rate of 8 incurs minimal performance loss.

Computation Cost Analysis. Table 4 presents the computational benefits of using Llama 🪆 . Specifically, we evaluate MS LoRA 🪆 with Llama 3.1-8B as LLM and detail the associated inference costs. Without compression, we assume the LLM processes 500 audio tokens, 250 video tokens (the resolution of the audio encoder is twice that of the video encoder), and 7 tokens for the textual prompt, totaling 757. As shown in the table, our proposed approach yields significant speedups, reducing TFLOPs by over 8x when applying compression rates of 16 and 5 for audio and video, respectively, thus substantially improving efficiency.

4 Conclusion

This work introduces Llama-MTSK, a versatile audio-visual MLLM capable of elastic inference across multiple tasks and computational resources. Llama-MTSK exploits the concept of matryoshka representation learning to adapt the pre-trained LLM through ad-hoc LoRA 🪆 modules, achieving performance comparable to or better than models separately trained on each compression rate while significantly reducing computational costs.

5 Limitations

During training, processing multiple sequences at various granularities increases the LLM's memory requirements. Therefore, selecting the compression rates is crucial and delicate; including too many rates is unfeasible, especially for AVSR, where we theoretically have up to $G \cdot T$ audio-video compression rate combinations. In addition to this, while our study focuses on LoRA for parameter-efficient LLM fine-tuning, other methods exist, such as adapter-tuning (Hu et al., 2023; Pfeiffer et al., 2021; Cappellazzo et al., 2024b) and advanced LoRA-based techniques (Zhang et al., 2023; Ding et al., 2023; Hayou et al., 2024; Liu et al., 2024a), which we did not explore. Extending our method to these approaches is an interesting direction for future work.

References

Triantafyllos Afouras, Joon Son Chung, Andrew Senior, Oriol Vinyals, and Andrew Zisserman. 2018a. Deep audio-visual speech recognition. *IEEE transactions on pattern analysis and machine intelligence*, 44(12):8717–8727.

Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman. 2018b. Lrs3-ted: a large-scale dataset for visual speech recognition. *arXiv preprint arXiv:1809.00496*.

Maxime Burchi and Radu Timofte. 2023. Audio-visual efficient conformer for robust speech recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2258–2267.

Mu Cai, Jianwei Yang, Jianfeng Gao, and Yong Jae Lee. 2024. Matryoshka multimodal models. *arXiv preprint arXiv:2405.17430*.

Umberto Cappellazzo, Daniele Falavigna, and Alessio Brutti. 2024a. Efficient fine-tuning of audio spectrogram transformers via soft mixture of adapters. *arXiv preprint arXiv:2402.00828*.

Umberto Cappellazzo, Daniele Falavigna, Alessio Brutti, and Mirco Ravanelli. 2024b. Parameter-efficient transfer learning of audio spectrogram transformers. In *2024 IEEE 34th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE.

Umberto Cappellazzo, Minsu Kim, Honglie Chen, Pingchuan Ma, Stavros Petridis, Daniele Falavigna, Alessio Brutti, and Maja Pantic. 2025. Large language models are strong audio-visual speech recognition learners. In *ICASSP*.

Junbum Cha, Wooyoung Kang, Jonghwan Mun, and Byungseok Roh. 2024. Honeybee: Locality-enhanced projector for multimodal llm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13817–13827.

C. Chen et al. 2024. It's never too late: Fusing acoustic information into large language models for automatic speech recognition. In *ICLR*.

Ning Ding, Xingtai Lv, Qiaosen Wang, Yulin Chen, Bowen Zhou, Zhiyuan Liu, and Maosong Sun. 2023. Sparse low-rank adaptation of pre-trained language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4133–4145, Singapore. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Stéphane Dupont and Juergen Luettin. 2000. Audio-visual speech modeling for continuous speech recognition. *IEEE transactions on multimedia*, 2(3):141–151.

Xiaoran Fan, Tao Ji, Changhao Jiang, Shuo Li, Senjie Jin, Sirui Song, Junke Wang, Boyang Hong, Lu Chen, Guodong Zheng, et al. 2024. Mousi: Poly-visual-expert vision-language models. *arXiv preprint arXiv:2401.17221*.

Qingkai Fang, Shoutao Guo, Yan Zhou, Zhengrui Ma, Shaolei Zhang, and Yang Feng. 2024a. Llama-omni: Seamless speech interaction with large language models. *arXiv preprint arXiv:2409.06666*.

Rongyao Fang, Chengqi Duan, Kun Wang, Hao Li, Hao Tian, Xingyu Zeng, Rui Zhao, Jifeng Dai, Hongsheng Li, and Xihui Liu. 2024b. Puma: Empowering unified mllm with multi-granular visual generation. *arXiv preprint arXiv:2410.13861*.

Yassir Fathullah, Chunyang Wu, Egor Lakomkin, Junteng Jia, Yuan Shangguan, Ke Li, Jinxi Guo, Wenhan Xiong, Jay Mahadeokar, Ozlem Kalinli, et al. 2024. Prompting large language models with speech recognition abilities. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 13351–13355. IEEE.

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

Enrico Fini, Mustafa Shukor, Xiujun Li, Philipp Dufter, Michal Klein, David Haldimann, Sai Aitharaju, Victor Guilherme Turrisi da Costa, Louis Béthune, Zhe Gan, et al. 2024. Multimodal autoregressive pre-training of large vision encoders. *arXiv preprint arXiv:2411.14402*.

Alexandros Haliassos, Pingchuan Ma, Rodrigo Mira, Stavros Petridis, and Maja Pantic. 2023. Jointly learning visual and auditory speech representations from raw data. In *International Conference on Learning Representations*.

Alexandros Haliassos, Rodrigo Mira, Honglie Chen, Zoe Landgraf, Stavros Petridis, and Maja Pantic. 2024a. Unified speech recognition: A single model for auditory, visual, and audiovisual inputs. In *NeurIPS*.

Alexandros Haliassos, Andreas Zinonos, Rodrigo Mira, Stavros Petridis, and Maja Pantic. 2024b. Braven: Improving self-supervised pre-training for visual and auditory speech recognition. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 11431–11435. IEEE.

Soufiane Hayou, Nikhil Ghosh, and Bin Yu. 2024. Lora+: Efficient low rank adaptation of large models. *arXiv preprint arXiv:2402.12354*.

Joanna Hong, Minsu Kim, Jeongsoo Choi, and Yong Man Ro. 2023. Watch or listen: Robust audio-visual speech recognition with visual corruption modeling and reliability scoring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18783–18794.

Joanna Hong, Minsu Kim, Daehun Yoo, and Yong Man Ro. 2022. Visual context-driven audio feature enhancement for robust end-to-end audio-visual speech recognition. In *Interspeech*, pages 2838–2842.

Wei-Ning Hsu and Bowen Shi. 2022. u-hubert: Unified mixed-modal speech pretraining and zero-shot transfer to unlabeled modality. *Advances in Neural Information Processing Systems*, 35:21157–21170.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Wenbo Hu, Zi-Yi Dou, Liunian Harold Li, Amita Kamath, Nanyun Peng, and Kai-Wei Chang. 2024a. Matryoshka query transformer for large vision-language models. *arXiv preprint arXiv:2405.19315*.

Y. Hu et al. 2024b. Large language models are efficient learners of noise-robust speech recognition. In *ICLR*.

Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria, and Roy Lee. 2023. LLM-adapters: An adapter family for parameter-efficient fine-tuning of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5254–5276, Singapore. Association for Computational Linguistics.

Rongjie Huang, Mingze Li, Dongchao Yang, Jiatong Shi, Xuankai Chang, Zhenhui Ye, Yuning Wu, Zhiqing Hong, Jiawei Huang, Jinglin Liu, et al. 2024. Audiogpt: Understanding and generating speech, music, sound, and talking head. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 23802–23804.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Sneha Kudugunta, Aditya Kusupati, Tim Dettmers, Kaifeng Chen, Inderjit Dhillon, Yulia Tsvetkov, Hannaneh Hajishirzi, Sham Kakade, Ali Farhadi, Prateek Jain, et al. 2024. Matformer: Nested transformer for elastic inference. In *NeurIPS*.

Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, et al. 2022. Matryoshka representation learning. *Advances in Neural Information Processing Systems*, 35:30233–30249.

Kushal Lakhotia, Eugene Kharitonov, Wei-Ning Hsu, Yossi Adi, Adam Polyak, Benjamin Bolte, Tu-Anh Nguyen, Jade Copet, Alexei Baevski, Abdelrahman Mohamed, et al. 2021. On generative spoken language modeling from raw audio. *Transactions of the Association for Computational Linguistics*, 9:1336–1354.

Byung-Kwan Lee, Chae Won Kim, Beomchan Park, and Yong Man Ro. 2024. Meteor: Mamba-based traversal of rationale for large language and vision models. In *NeurIPS*.

Jiachen Li, Xinyao Wang, Sijie Zhu, Chia-Wen Kuo, Lu Xu, Fan Chen, Jitesh Jain, Humphrey Shi, and Longyin Wen. 2024. Cumo: Scaling multimodal llm with co-upcycled mixture-of-experts. In *NeurIPS*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. 2024. Vila: On pretraining for visual language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26689–26699.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Advances in neural information processing systems*, 36.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. 2024a. Dora: Weight-decomposed low-rank adaptation. In *ICML*.

Zhijian Liu, Ligeng Zhu, Baifeng Shi, Zhuoyang Zhang, Yuming Lou, Shang Yang, Haocheng Xi, Shiyi Cao, Yuxian Gu, Dacheng Li, et al. 2024b. Nvila: Efficient frontier visual language models. *arXiv preprint arXiv:2412.04468*.

Ke-Han Lu, Zhehuai Chen, Szu-Wei Fu, Chao-Han Huck Yang, Jagadeesh Balam, Boris Ginsburg, Yu-Chiang Frank Wang, and Hung-yi Lee. 2025. Developing instruction-following speech language model without speech instruction-tuning data. In *ICASSP*.

Gen Luo, Yiyi Zhou, Yuxin Zhang, Xiawu Zheng, Xiaoshuai Sun, and Rongrong Ji. 2024. Feast your eyes: Mixture-of-resolution adaptation for multimodal large language models. *arXiv preprint arXiv:2403.03003*.

Pingchuan Ma, Alexandros Haliassos, Adriana Fernandez-Lopez, Honglie Chen, Stavros Petridis, and Maja Pantic. 2023a. Auto-avsr: Audio-visual speech recognition with automatic labels. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Pingchuan Ma, Alexandros Haliassos, Adriana Fernandez-Lopez, Honglie Chen, Stavros Petridis, and Maja Pantic. 2023b. Auto-avsr: Audio-visual speech recognition with automatic labels. In *ICASSP*.

Pingchuan Ma, Stavros Petridis, and Maja Pantic. 2021. End-to-end audio-visual speech recognition with conformers. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7613–7617. IEEE.

Ziyang Ma, Guanrou Yang, Yifan Yang, Zhifu Gao, Jiaming Wang, Zhihao Du, Fan Yu, Qian Chen, Siqi Zheng, Shiliang Zhang, et al. 2024. An embarrassingly simple approach for llm with strong asr capacity. *arXiv preprint arXiv:2402.08846*.

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Pete Walsh, Oyvind Tafjord, Nathan Lambert, et al. 2024. Olmoe: Open mixture-of-experts language models. *arXiv preprint arXiv:2409.02060*.

Basil Mustafa, Carlos Riquelme, Joan Puigcerver, Rodolphe Jenatton, and Neil Houlsby. 2022. Multimodal contrastive learning with limoe: the language-image mixture of experts. *Advances in Neural Information Processing Systems*, 35:9564–9576.

Pranav Nair, Puranjay Datta, Jeff Dean, Prateek Jain, and Aditya Kusupati. 2025. Matryoshka quantization. *arXiv preprint arXiv:2502.06786*.

Kuniaki Noda, Yuki Yamaguchi, Kazuhiro Nakadai, Hiroshi G Okuno, and Tetsuya Ogata. 2015. Audio-visual speech recognition using deep learning. *Applied intelligence*, 42:722–737.

Se Jin Park, Chae Won Kim, Hyeongseop Rha, Minsu Kim, Joanna Hong, Jeong Hun Yeo, and Yong Man Ro. 2024. Let's go real talk: Spoken dialogue model for face-to-face conversation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Stavros Petridis, Themos Stafylakis, Pingchuan Ma, Georgios Tzimiropoulos, and Maja Pantic. 2018. Audio-visual speech recognition with a hybrid ctc/attention architecture. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 513–520. IEEE.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. AdapterFusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 487–503, Online. Association for Computational Linguistics.

Joan Puigcerver, Carlos Riquelme, Basil Mustafa, and Neil Houlsby. 2023. From sparse to soft mixtures of experts. *arXiv preprint arXiv:2308.00951*.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR.

Andrew Rouditchenko, Yuan Gong, Samuel Thomas, Leonid Karlinsky, Hilde Kuehne, Rogerio Feris, and James Glass. 2024. Whisper-flamingo: Integrating visual features into whisper for audio-visual speech recognition and translation. In *Interspeech*.

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

Bowen Shi, Wei-Ning Hsu, Kushal Lakhotia, and Abdelrahman Mohamed. 2022. Learning audio-visual speech representation by masked multimodal cluster prediction. In *International Conference on Learning Representations*.

Joon Son Chung, Andrew Senior, Oriol Vinyals, and Andrew Zisserman. 2017. Lip reading sentences in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6447–6456.

Weiting Tan, Hirofumi Inaguma, Ning Dong, Paden Tomasello, and Xutai Ma. 2024. Ssr: Alignment-aware modality connector for speech language models. *arXiv preprint arXiv:2410.00168*.

Changli Tang, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, and Chao Zhang. 2023. Salmonn: Towards generic hearing abilities for large language models. *arXiv preprint arXiv:2310.13289*.

Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. 2024. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9568–9578.

A Varga. 1992. Assessment for automatic speech recognition: Ii. noisex-92: A database and an experiment to study the effect of additive noise on speech recognition systems. *Elsevier Speech Commun*, 2(3):247.

Huanjin Yao, Wenhao Wu, Taojiannan Yang, YuXin Song, Mengxi Zhang, Haocheng Feng, Yifan Sun, Zhiheng Li, Wanli Ouyang, and Jingdong Wang. 2024. Dense connector for mllms. In *NeurIPS*.

Jeong Hun Yeo, Seunghee Han, Minsu Kim, and Yong Man Ro. 2024. Where visual speech meets language: Vsp-llm framework for efficient and context-aware visual speech processing. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 11391–11406.

Wenyi Yu, Changli Tang, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, and Chao Zhang. 2024. Connecting speech encoder and large language model for asr. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 12637–12641. IEEE.

Boqiang Zhang, Kehan Li, Zesen Cheng, Zhiqiang Hu, Yuqian Yuan, Guanzheng Chen, Sicong Leng, Yuming Jiang, Hang Zhang, Xin Li, et al. 2025a. Videollama 3: Frontier multimodal foundation models for image and video understanding. *arXiv preprint arXiv:2501.13106*.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. 2023. Adalora: Adaptive budget allocation for parameter-efficient fine-tuning. In *ICLR*.

Shaolei Zhang, Qingkai Fang, Zhe Yang, and Yang Feng. 2025b. Llava-mini: Efficient image and video large multimodal models with one vision token. *arXiv preprint arXiv:2501.03895*.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*.

Zhuofan Zong, Bingqi Ma, Dazhong Shen, Guanglu Song, Hao Shao, Dongzhi Jiang, Hongsheng Li, and Yu Liu. 2024. Mova: Adapting mixture of vision experts to multimodal context. In *NeurIPS*.

Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, and William Fedus. 2022. St-moe: Designing stable and transferable sparse expert models. *arXiv preprint arXiv:2202.08906*.

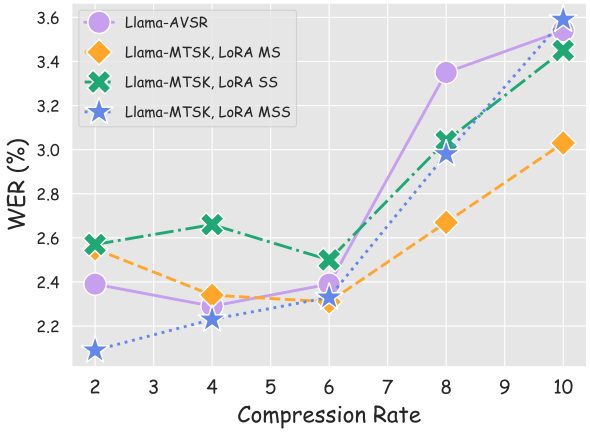

Figure 5: Additional WER results using stacking compression for the ASR task with {2, 4, 6, 8, 10} rates. We use the same configuration as in Figure 3.

A Appendix

A.1 Additional Experiments for ASR

In this section, we report additional results for the ASR task when using compression rates in a different range , specifically {2, 4, 6, 8, 10}. Compared to Figure 3, the increment between two consecutive rates is halved. We argue that it is more useful to use more diverse rates for ASR since we do not observe much deterioration of the WER results when doubling the rate (in Figure 5, the baseline Llama-AVSR achieves similar results when compressing the tokens of a factor 2, 4, and 6). Figure 5 shows that Llama 🪆 MS and MSS achieves comparable or better performance than Llama-AVSR. As for the SS approach, it performs slightly worse than Llama-AVSR for the first compression rates, and we believe this is because having a specific LoRA module for multiple rates which do not show WER deterioration leads to overfitting as one global LoRA is sufficient. This argument also explains why for rates 8 and 10 the MS variant performs better than the other ones.

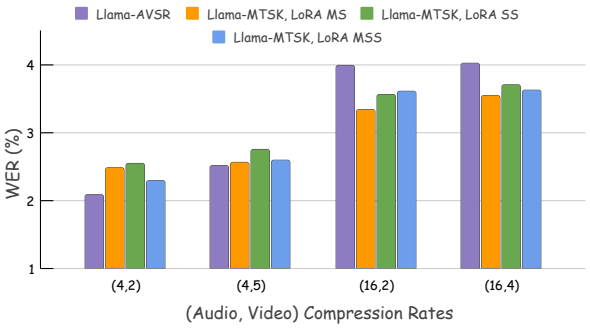

Figure 6: Additional results for Llama 🪆 using stacking compression on the LRS3 dataset.

Table 5: Comparison between Llama-AVSR and our proposed Llama 🪆 MS, SS, and MSS approaches on LRS2 and LRS3 benchmarks. We employ Whisper medium and Llama 3.1-8B. †Llama-AVSR trains 4 independent models tailored to each configuration of audio-video compression rates.

Method	Compression Rates (A,V)			
	(4,2)	(4,5)	(16,2)	(16,5)
LRS3 Dataset				
Llama-AVSR†	**0.9**	**0.9**	1.6	2.1
Llama 🪆 MS	1.0	1.1	**1.5**	**1.6**
Llama 🪆 SS	**0.9**	1.0	1.7	1.8
Llama 🪆 MSS	1.2	1.0	**1.5**	**1.6**
LRS2 Dataset				
Llama-AVSR	2.4	2.2	**2.9**	3.3
Llama 🪆 MS	**2.1**	2.3	**2.9**	3.2
Llama 🪆 SS	2.4	**2.1**	**2.9**	**2.9**
Llama 🪆 MSS	2.4	2.5	3.2	3.4

A.2 AVSR Results with Stacking Compression

We include additional results for AVSR on LRS3 using the stacking compression method in Figure 6. The methods use Whisper Small and Llama 3.2-1B as LLM. Our three proposed Matryoshka approaches performs better than or equally well as Llama-AVSR, especially under conditions of high audio compression, underscoring the effectiveness of our proposed Llama 🪆.

A.3 Full AVSR Results with Whisper Medium and LLama 3.1-8B

In Table 2, we only included for Llama-AVSR and Llama 🪆 the results with audio and video compression rates equal to 4 and 2, respectively. In Table 5, we also report additional configurations of audio-video compression rates. We use Whisper medium as audio encoder and Llama 3.1-8B as LLM. Once more, our proposed methods perform on par or even better than independently-trained Llama-AVSR models for each compression rates configurations. In particular, we highlight the sizeable gains brought by all the three LoRA 🪆 approaches for LRS3 when we apply the highest compression rates configuration (16,5).