Dynamic routing-based Mixture of Layer-wise Tokens with Token-Difficulty for Robust Audio-Visual Speech Recognition

Changhan Oh[1,2] **Yeeun Jo**[1,2] **Ran Han**[2] **Jeomja Kang**[2]
Woo Yong Choi[2] **Hwa Jeon Song**[2] **Kiyoung Park**[1,2]
[1]University of Science and Technology (UST), South Korea
[2]Electronics and Telecommunications Research Institute (ETRI), South Korea
{ochh508, jyen, ran.han, jjkang, wychoi4, songhj, pkyoung}@etri.re.kr

Abstract

Audio-Visual Speech Recognition (AVSR) systems typically consume only the final-layer representation of their modality-specific encoders. In self-supervised (SSL) encoders, lower and upper layers encode acoustic and more abstract linguistic cues respectively, so the most informative layer may differ from token to token. We propose Mixture of Layer-wise Tokens (MoLT), which aggregates the per-layer outputs of the AV-HuBERT encoder per token through an MoE-style router. A sparse top-k MoLT (S-MoLT) selects the most useful layers per token, and Dynamic routing-based MoLT (D-MoLT) replaces the fixed k with a learnable per-token threshold. To align routing with recognition difficulty, we further add a token-difficulty-aware lower-upper preference that biases, rather than overrides, the learned routing: Difficulty-adaptive D-MoLT (DD-MoLT) derives the difficulty signal from one modality, while Cross-modal DD-MoLT (CDD-MoLT) uses the audio-video reliability gap. On LRS3, our models consistently improve WER over a strong MMS-LLaMA baseline across SNR conditions, with the gain widening under severe noise (e.g., 13.85 vs. 17.10% at -15 dB), at negligible overhead.

1 Introduction

Automatic Speech Recognition (ASR) systems that rely solely on acoustic signals are inherently limited in their robustness under noisy conditions. Audio-Visual Speech Recognition (AVSR) addresses this by jointly leveraging visual information, such as lip movements, alongside acoustic signals (Afouras et al., 2018a; Ma et al., 2022) to achieve robust recognition in both clean and noisy environments.

Recent advances in large-scale pre-training have produced high-quality modality-specific encoders such as the self-supervised AV-HuBERT (Shi et al., 2022) and the weakly-supervised Whisper (Radford et al., 2023), and connecting these to Large Language Model (LLM) decoders via projection layers or Q-Formers (Li et al., 2023) has further boosted AVSR accuracy (Fathullah et al., 2024; Cappellazzo et al., 2025a; Yeo et al., 2025). Despite this progress, significant performance degradation persists under strong acoustic noise, necessitating further improvements to ensure robustness.

Several prior works have found that intermediate layers of SSL speech encoders contain information beneficial to speech recognition (Pasad et al., 2021, 2023; wen Yang et al., 2021). However, most existing AVSR approaches rely exclusively on the final-layer output of each encoder, leaving complementary intermediate-layer information unexploited. We hypothesize that leveraging these intermediate representations can provide richer audio-visual features and enable more robust recognition under adverse acoustic conditions.

Motivated by this, we propose the **Mixture of Layer-wise Tokens (MoLT)** module, which selectively aggregates intermediate layer outputs of the AV-HuBERT encoder on a per-token basis, inspired by Mixture of Experts (MoE) architectures (Shazeer et al., 2017; Fedus et al., 2022). The Top-k Sparse Mixture of Layer-wise Tokens (S-MoLT) selects only the top-k most beneficial layers per token, reducing the influence of task-irrelevant representations.

Since the optimal layer combination may differ across tokens, we further propose **Dynamic routing-based MoLT (D-MoLT)**, which replaces the fixed top-k selection with a learnable threshold-based mechanism that adaptively determines which layers to activate per token.

To guide routing in alignment with AVSR task characteristics, we additionally introduce a **lower-upper preference mechanism** based on token difficulty, which biases the dynamic routing toward task-appropriate layers without enforcing uniform layer utilization.

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 39954–39969
October 24-29, 2026 ©2026 Association for Computational Linguistics

2 Related Work

2.1 Audio-Visual Speech Recognition Models

Early AVSR systems demonstrated that jointly modeling lip movements and acoustic signals substantially improves robustness over audio-only ASR (Afouras et al., 2018a; Petridis et al., 2018; Ma et al., 2021, 2022). Self-supervised pre-training approaches such as AV-HuBERT (Shi et al., 2022) and RAVEn (Haliassos et al., 2023) further reduced reliance on labeled data by learning unified audio-visual representations from large unlabeled corpora. Building on these encoders, recent work couples them with LLM decoders via projection layers or Q-Formers (Li et al., 2023; Fathullah et al., 2024; Cappellazzo et al., 2025a; Yeo et al., 2025), yet AVSR performance still degrades sharply under low SNR conditions, motivating richer use of encoder representations beyond the final layer.

2.2 Mixture of Experts and Layer-wise Token Mixture

Sparsely activated Mixture of Experts (MoE) models route each input token to a subset of expert sub-networks, scaling model capacity without a proportional increase in compute (Shazeer et al., 2017; Lepikhin et al., 2021; Fedus et al., 2022). Flexible routing variants include dynamic, input-adaptive selection of the number of activated experts (Huang et al., 2024; Guo et al., 2025) and layer-wise token mixing that routes tokens to a variable number of layers or mixes representations across depth (Raposo et al., 2024; Antoniak et al., 2024). To prevent routing collapse, most MoE variants rely on auxiliary load-balancing or diversity losses that encourage uniform expert utilization (Shazeer et al., 2017; Fedus et al., 2022). In contrast, our work treats per-layer SSL encoder outputs as token-level candidate representations and introduces a token-difficulty-aware preference that biases routing toward task-appropriate layers, without enforcing uniform layer utilization.

3 Method

We adopt MMS-LLaMA as our baseline, which couples pre-trained audio and visual encoders with an LLM decoder. We augment it with the Mixture of Layer-wise Tokens (MoLT) module that exploits intermediate layer outputs of the AV-HuBERT encoder; the resulting visual features are concatenated with audio features before being passed to the AV Q-Former. Figure 1 illustrates the overall architecture.

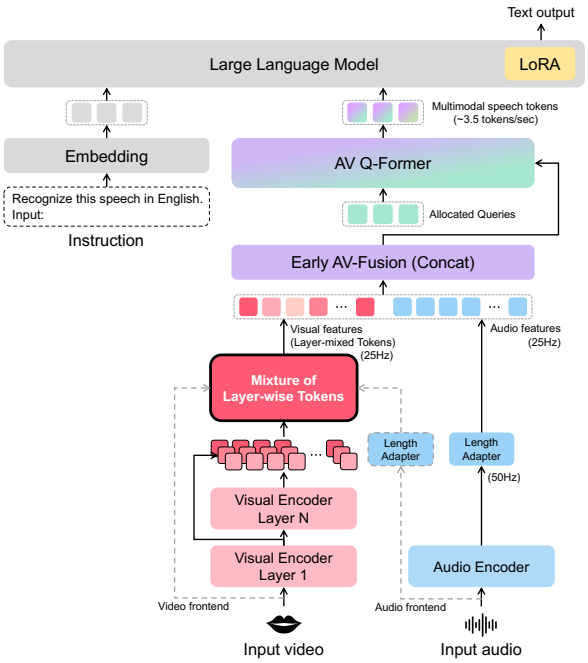


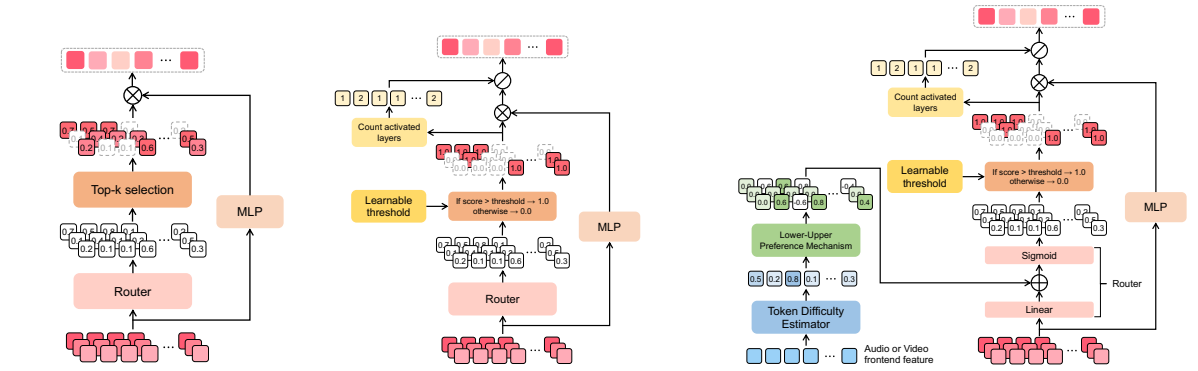
Figure 1: Overall architecture of the proposed LLM-based AVSR system. The Mixture of Layer-wise Tokens module aggregates intermediate layer outputs of the visual SSL encoder before concatenation with audio features for the AV Q-Former. The audio- and video-frontend features are used only in the token-difficulty-based variants.

Figure 2 presents each variant of the proposed Mixture of Layer-wise Tokens, and in this section we describe in detail how each method selects and aggregates intermediate layers on a per-token basis from the encoder.

3.1 Top-k based Sparse Mixture of Layer-wise Tokens (S-MoLT)

As a foundational approach to verify the effectiveness of leveraging intermediate layer information from the SSL encoder, we propose Top-k based Sparse Mixture of Layer-wise Tokens, inspired by the Mixture of Experts (MoE) framework and Raposo et al. (2024).

Conventional MoE selects the top-k experts according to scores produced by a router and constructs the final representation by computing a weighted sum of their outputs using the corresponding scores. Unlike Shazeer et al. (2017), we treat the intermediate layers of the encoder as experts and aim to construct a representation by aggregating the layer outputs that are beneficial for recognition performance.



(a) S-MoLT (b) D-MoLT (c) DD-MoLT

Figure 2: Illustrations of the three proposed Mixture of Layer-wise Tokens (MoLT) variants. (a) S-MoLT uses a fixed top-k selection; (b) D-MoLT replaces top-k with a learnable threshold; (c) DD-MoLT additionally incorporates a Token Difficulty Estimator to bias routing toward lower or upper layers depending on token difficulty. Here, $\oplus$, $\otimes$, and $\oslash$ denote element-wise addition, multiplication, and division: $\oplus$ adds the lower-upper preference bias to the base routing logits, $\otimes$ applies the per-token routing weights (or the binary gate) to the per-layer features, and $\oslash$ normalizes the aggregated representation by the number of activated layers.

Let the t-th token at the l-th layer of the encoder be denoted as $\mathbf{h}_t^l \in \mathbb{R}^D$, $l \in \{1, \ldots, L\}$. Each $\mathbf{h}_t^l$ is mapped to a scalar score s_t^l via a shared linear projection applied uniformly across all layers:

$$s_t^l = \mathbf{w}^\top \mathbf{h}_t^l + b$$

where $\mathbf{w} \in \mathbb{R}^D$ is a shared weight vector and $b \in \mathbb{R}$ is a bias term. The score vector over all layers is defined as $\mathbf{s}_t = [s_t^1, \ldots, s_t^L] \in \mathbb{R}^L$.

Each score s_t^l is normalized via softmax to obtain the routing probability p_t^l:

$$p_t^l = \frac{\exp(s_t^l)}{\sum_{j=1}^{L} \exp(s_t^j)}$$

The value of p_t^l lies in the range $(0, 1)$, where a higher value indicates a greater contribution of the corresponding layer output to the final representation. To retain only the top-k layers with the highest routing probabilities, we introduce a top-k selection module as follows.

Letting $\mathcal{K}_t = \text{Top-K}(p_t^{1:L}, k)$ denote the set of top-k indices, the re-normalized routing weight $\tilde{p}_t^l$ is defined as:

$$\tilde{p}_t^l = \begin{cases} \dfrac{p_t^l}{\sum_{j \in \mathcal{K}_t} p_t^j}, & l \in \mathcal{K}_t, \\ 0, & \text{otherwise.} \end{cases}$$

The top-k selection is performed independently for each token position t.

Each $\mathbf{h}_t^l$ is then transformed through an MLP as follows:

$$\mathbf{f}_t^l = \text{MLP}(\mathbf{h}_t^l) \in \mathbb{R}^D$$

The final encoder representation $\mathbf{z}_t$ is then obtained by computing a weighted sum of the selected layer representations using $\tilde{p}_t^l$ as weights:

$$\mathbf{z}_t = \sum_{l \in \mathcal{K}_t} \tilde{p}_t^l \, \mathbf{f}_t^l, \quad \mathbf{z}_t \in \mathbb{R}^D$$

3.2 Dynamic routing-based Mixture of Layer-wise Tokens (D-MoLT)

Since Sparse Mixture of Layer-wise Tokens relies on the top-k module, it applies a fixed k uniformly to all tokens despite the fact that the appropriate number of layers may vary across tokens. To address this limitation, inspired by Guo et al. (2025), we propose a dynamic routing-based Mixture of Layer-wise Tokens framework that determines k in an adaptive manner.

We replace the conventional top-k selection scheme with a learnable threshold-based selection rule and introduce a top-any mechanism that dynamically determines the number of activated layers. By selectively activating only the layers necessary for each token, this approach aims to produce richer encoder-derived representations compared to the fixed top-k counterpart.

Threshold-based Dynamic routing. For the t-th token representation $\mathbf{h}_t^l \in \mathbb{R}^D$ at layer l, the routing score is computed via cosine similarity with a learnable router vector $\mathbf{r} \in \mathbb{R}^D$:

$$s_t^l = \cos(\mathbf{h}_t^l, \mathbf{r}) = \frac{\mathbf{h}_t^{l\top}\mathbf{r}}{\|\mathbf{h}_t^l\|\,\|\mathbf{r}\|}$$

The routing score s_t^l is then mapped to the range $(0, 1)$ via the sigmoid function:

$$p_t^l = \sigma(s_t^l), \quad \sigma(x) = \frac{1}{1+e^{-x}}$$

A single learnable threshold parameter $G \in \mathbb{R}$, shared across all layers, is defined such that $\theta = \sigma(G)$. The activation of each layer is determined by a binary gate mask as follows:

$$m_t^l = \mathbb{I}\left[p_t^l \geq \theta\right], \quad m_t^l \in \{0, 1\}$$

where $\mathbb{I}(\cdot)$ is an indicator function that activates only those layers whose routing probability meets or exceeds the threshold.

During training, to mitigate the non-differentiability introduced by the binary gate, backpropagation is performed using the straight-through estimator (STE) (Bengio et al., 2013; Jang et al., 2017). This allows the binary gate to be maintained in the forward pass while enabling gradient flow through the continuous probability values in the backward pass.

Minimum Activation Constraint. In the threshold-based selection scheme, it is possible that no layer is activated for certain tokens. To prevent this, we enforce a minimum number of activated layers $k_{\min}$, set to $k_{\min} = 1$ in this work.

Defining the number of activated layers as

$$k_t = \sum_{l=1}^{L} m_t^l,$$

if $k_t < k_{\min}$, the top-$k_{\min}$ layers with the highest routing scores are forcibly selected as:

$$\mathcal{L}^* = \underset{l}{\text{Top-K}}\left(s_t^l,\ k_{\min}\right), \quad m_t^l = 1 \text{ for } l \in \mathcal{L}^*$$

This constraint ensures representational stability and prevents potential information loss during the routing process.

Masked Layer Aggregation. The representation at each layer is projected into a common representation space via an MLP:

$$\mathbf{g}_t^l = \text{MLP}(\mathbf{h}_t^l)$$

The representations of the selected layers are then aggregated using the binary gate mask and normalized by the number of activated layers to obtain the final encoder representation $\mathbf{z}_t$:

$$\mathbf{z}_t = \frac{\sum_{l=1}^{L} m_t^l\, \mathbf{g}_t^l}{\sum_{l=1}^{L} m_t^l}$$

Since the Minimum Activation Constraint defined above guarantees $k_t \geq k_{\min} = 1$ at all times, the denominator is never zero. This formulation enables balanced aggregation of information from the selected layers while allowing a flexible, token-specific number of layers to be utilized.

3.3 Token Difficulty based Lower-Upper Preference Mechanism for D-MoLT

3.3.1 Token Difficulty Estimation

Motivated by curriculum learning, which has shown that adapting model behavior to per-example difficulty improves generalization (Bengio et al., 2009), we introduce a lightweight Token Difficulty Estimator that predicts a scalar difficulty score from the token representation. Specifically, given an input representation $\mathbf{h}_t \in \mathbb{R}^D$ (the audio-frontend feature or the video-frontend feature, depending on the variant defined below), the difficulty score $\hat{s}_t \in [0, 1]$ is estimated as:

$$\hat{s}_t = \sigma(\mathbf{W}_2\,\text{GELU}(\mathbf{W}_1\,\text{LN}(\mathbf{h}_t) + \mathbf{b}_1) + b_2)$$

where $\text{LN}(\cdot)$ denotes Layer Normalization, $\mathbf{W}_1 \in \mathbb{R}^{D_h \times D}$ and $\mathbf{W}_2 \in \mathbb{R}^{1 \times D_h}$ are learnable parameters, and D_h is the hidden dimension set to $D/4$ by default. A value of $\hat{s}_t$ close to 0 indicates that the token is clean or easy to recognize, while a value close to 1 indicates that the token is heavily corrupted by noise or difficult to recognize. The estimator is trained end-to-end without external supervision, acquiring difficulty signals indirectly from the ASR task loss.

3.3.2 Lower-Upper Preference Mechanism

Based on the estimated difficulty score, we propose a method for correcting the base routing scores of D-MoLT. We first define the layer rank r_l, which represents the relative position of each layer:

$$r_l = \frac{2l - L - 1}{L - 1} \in [-1, 1], \quad l \in \{1, \ldots, L\}$$

r_l takes values close to -1 for lower layers and close to 1 for upper layers. A noise bias computed from the difficulty score is added to the base logit to obtain a corrected routing logit, thereby guiding the routing such that layers appropriate to the difficulty of each token are naturally selected. The corrected logit is converted to a routing probability via the sigmoid function, as in the threshold-based dynamic routing described above, and compared against the learnable threshold θ to determine the binary gate m_t^l.

Depending on which features are used to estimate the difficulty score, we propose two variants.

Difficulty-adaptive D-MoLT. This variant estimates the difficulty score from a single-modality feature (either the audio-frontend feature or the video-frontend feature). The estimated difficulty score $\hat{s}_t$ is transformed into a directional signal in the range $[-1, 1]$:

$$d_t = 2\hat{s}_t - 1 \in [-1, 1]$$

and the noise bias is computed as:

$$b_t^l = \alpha \cdot d_t \cdot r_l$$

The corrected routing logit is defined as $\tilde{s}_t^l = s_t^l + b_t^l$. For tokens with low difficulty ($d_t < 0$), the routing logits of lower layers ($r_l < 0$) are increased, promoting the selection of lower layers that preserve fine-grained acoustic-visual cues. Conversely, for tokens with high difficulty ($d_t > 0$), the logits of upper layers are relatively increased, leading to greater activation of upper layers that are rich in contextual information.

Cross-modal Difficulty-adaptive D-MoLT. This variant employs two modality-specific Token Difficulty Estimators to independently estimate difficulty scores $\hat{s}_t^a$ and $\hat{s}_t^v$ from the audio-frontend feature and the video-frontend feature, respectively, and incorporates the relative reliability difference between the two modalities into the routing. The difference between the two difficulty scores is defined as:

$$\delta_t = \hat{s}_t^a - \hat{s}_t^v \in [-1, 1]$$

and the noise bias is computed as:

$$b_t^l = \alpha \cdot \delta_t \cdot r_l$$

A positive δ_t indicates that audio is noisier than video; $\delta_t = 0$ preserves the base routing. In both variants, the scale parameter α is learnable.

4 Experiments

4.1 Experimental Setup

Datasets. We utilize Lip Reading Sentences 3 (LRS3) (Afouras et al., 2018b) and VoxCeleb2 (Chung et al., 2018) for our main experiments. LRS3 is an audio-visual corpus collected from TED and TEDx talks, comprising approximately 433 hours of transcribed data. VoxCeleb2 is a large-scale speaker recognition corpus extracted from YouTube videos, consisting of approximately 2,442 hours of multilingual audio-visual data. Following Shi et al. (2022) and Yeo et al. (2025), we use only the English portion (1,326 hours) and obtain pseudo-transcriptions for the unlabeled audio with the Whisper (Radford et al., 2023) ASR model. Both corpora are used for training, while all evaluations are reported on the LRS3 test set.

We additionally assess cross-corpus generalization on LRS2 (Afouras et al., 2018a), a second audio-visual benchmark of approximately 225 hours of transcribed video collected from BBC broadcasts, in Appendix C.

Implementation Details. Since we build upon MMS-LLaMA (Yeo et al., 2025) as our backbone, all components other than the proposed MoLT module variants (S-MoLT, D-MoLT) are kept identical to it, including the pre-processing pipeline and training hyperparameters (detailed in Appendix A).

We adopt AV-HuBERT Large (L=24 layers) as the visual encoder, Whisper (Radford et al., 2023) medium-en as the audio encoder, and LLaMA-3.2 3B (Grattafiori et al., 2024) as the LLM decoder. Babble noise from the NOISEX dataset (Varga and Steeneken, 1993) is randomly mixed into the audio at a 75% probability. The AV Q-Former and MoLT module are trained from scratch, while the encoders and the LLM are kept frozen except for QLoRA (Dettmers et al., 2023) adapters on the LLM.

Unless otherwise specified, all MoLT variants route over the layers of the visual encoder (AV-HuBERT). For D-MoLT the threshold logit G is initialized so that $\sigma(G)$=0.5, the minimum activation is $k_{\min}$=1, and the difficulty scale α is learnable and initialized to 1.0. For DD-MoLT, the audio-frontend feature denotes the output of the Whisper convolutional frontend (i.e., the feature immediately prior to the Whisper Transformer layers), and the video-frontend feature denotes the output of the AV-HuBERT ResNet visual frontend (i.e., the fea-

Method	WER (%) at SNR (dB)					
	clean	5	0	-5	-10	-15
Baseline	1.00	1.37	2.83	7.85	17.41	26.67
Top-1 layer	0.92	1.37	2.69	8.38	18.94	27.57
Top-2 layers	0.92	**1.29**	**2.35**	**6.97**	14.49	21.40
Top-3 layers	**0.82**	1.34	2.53	7.32	16.66	23.97
Top-4 layers	0.91	1.42	2.53	7.62	15.86	23.17
Top-5 layers	0.89	2.29	2.52	7.20	15.88	22.26
Top-6 layers	0.89	1.37	2.50	7.11	**14.35**	**21.39**

Table 1: WER (%) on LRS3 across Top-k settings of S-MoLT, which leverages intermediate layer representations of the AV-HuBERT encoder. All experiments are reproduced in our environment.

ture immediately prior to the AV-HuBERT Transformer layers). All experiments are conducted on 8 NVIDIA L40 48 GB GPUs.

4.2 Results of Sparse Mixture of Layer-wise Tokens (S-MoLT)

Table 1 reports WER on the LRS3 test set for each Top-k setting of S-MoLT. Unlike the baseline, which relies solely on the final-layer representation, S-MoLT generally achieves lower WER across most SNR conditions, with only a few exceptions. Notably, the Top-2 layers and Top-6 layers configurations attain the lowest WER values, particularly under noisy conditions (SNR ≤ -5 dB), demonstrating that exploiting intermediate layer information is especially effective for improving AVSR robustness in adverse acoustic environments. Nevertheless, some Top-k settings underperform the baseline, indicating that a fixed, uniform number of layers is suboptimal and motivating per-token routing.

We further compare against a SUPERB-style aggregation (wen Yang et al., 2021), which combines the per-layer outputs of the visual (AV-HuBERT) encoder through a learned weighted summation of per-layer scalar weights. As shown in Table 2, this approach is beneficial under relatively clean conditions (SNR ≥ 0 dB) but falls below the baseline as noise increases. This indicates that adaptively selecting layers per token, as in S-MoLT, is more effective under noisy conditions than applying a single set of layer-wise weights uniformly to all tokens.

Applying S-MoLT to the Whisper audio encoder, by contrast, generally degrades performance across all conditions (Table 2). We discuss this further in Section 4.6.1.

Method	WER (%) at SNR (dB)					
	clean	5	0	-5	-10	-15
Baseline (Yeo et al., 2025)	1.00	1.37	2.83	7.85	17.41	26.67
SUPERB (wen Yang et al., 2021) (U-24)	0.90	1.29	2.61	8.32	19.07	28.53
SUPERB (wen Yang et al., 2021) (U-6)	0.95	1.33	2.74	8.26	18.33	26.45
S-MoLT (Audio Top-3, U-6)	1.06	1.54	2.86	8.11	18.28	25.45
S-MoLT (Video Top-24, U-24)	0.91	1.18	2.43	7.13	15.24	21.24
S-MoLT (Video Top-2, U-6)	0.91	1.37	2.35	6.78	14.72	21.68
S-MoLT (Video Top-3, U-6)	0.96	1.50	2.55	7.41	16.32	23.53
D-MoLT (U-24)	0.93	1.25	2.48	7.53	16.38	23.27
D-MoLT (U-6)	0.95	1.47	2.37	6.94	14.92	21.18
DD-MoLT (Audio-frontend, U-24)	0.96	1.25	2.36	7.52	17.27	24.98
DD-MoLT (Audio-frontend, U-6)	0.90	1.32	2.55	**6.45**	**13.93**	**19.82**
DD-MoLT (Video-frontend, U-24)	1.04	1.49	2.60	7.08	15.48	22.17
DD-MoLT (Video-frontend, U-6)	0.90	**1.16**	**2.28**	7.05	15.56	22.67
CDD-MoLT (U-24)	0.88	1.23	2.50	7.37	15.99	24.07
CDD-MoLT (U-6)	**0.77**	**1.16**	**2.28**	6.78	15.06	20.92

Table 2: WER (%) on LRS3 across SNR conditions. U-n: only the upper n layers of AV-HuBERT are used as layer candidates. Audio/Video Top-n: Top-n selection is applied to the audio or video encoder (Whisper or AV-HuBERT). All D-MoLT variants (D-MoLT, DD-MoLT, and CDD-MoLT) apply MoLT only to the visual (AV-HuBERT) encoder. Audio-/Video-frontend: difficulty score estimated from the Whisper or AV-HuBERT frontend feature, respectively.

4.3 Results of Dynamic routing-based Mixture of Layer-wise Tokens (D-MoLT)

Table 2 reports WER for D-MoLT under two configurations: all 24 AV-HuBERT layers (U-24) and the top six layers by routing score (U-6) (motivated by the layer analysis in Section 4.6.1).

D-MoLT consistently outperforms the baseline across all SNR conditions, with particularly pronounced gains in the low-SNR regime. Compared with the best fixed-k S-MoLT configuration (Top-6, U-24) reported in Table 1, D-MoLT (U-6) achieves comparable performance on most SNR levels while further reducing WER at -15 dB from 21.39 to 21.18. Restricting the layer candidates to the upper six layers (U-6) yields better WER than using all 24 layers (U-24) under heavy noise (e.g., 21.18 vs. 23.27 at -15 dB), confirming that the intermediate layers with the highest routing scores carry information most beneficial for AVSR while pruning task-irrelevant lower layers. However, the gains of D-MoLT (U-6) over the best S-MoLT configuration (Top-2, U-24: 21.40 at -15 dB) remain modest, suggesting that dynamic routing alone, without explicit guidance, does not fully exploit per-token characteristics.

4.4 Results of Difficulty-adaptive D-MoLT Variants

Difficulty-adaptive D-MoLT (DD-MoLT). We evaluate two DD-MoLT variants differing in the

source of the difficulty signal: the audio- and video-frontend features (Audio and Video, respectively).

Table 2 shows that DD-MoLT (Audio, U-6) achieves the best WER under heavy noise, attaining 6.45, 13.93, and 19.82 % at −5, −10, and −15 dB, respectively, outperforming D-MoLT (U-6) by 0.49, 0.99, and 1.36 absolute WER points at those conditions. This demonstrates that audio-based difficulty estimation effectively captures per-token noise severity, enabling the router to activate more upper-layer representations for heavily corrupted tokens. DD-MoLT (Video, U-6), on the other hand, achieves the best WER under relatively mild conditions, attaining 1.16 and 2.28 % at SNR 5 and SNR 0 dB, respectively. Since the video modality is inherently unaffected by acoustic noise, the video-based difficulty estimator provides more stable guidance under mild conditions, while it becomes less discriminative as noise severity increases. These complementary behaviors suggest that audio- and video-based difficulty signals capture different aspects of token-level difficulty, motivating a cross-modal approach.

Cross-modal DD-MoLT (CDD-MoLT). CDD-MoLT uses $\delta_t = \hat{s}_t^a - \hat{s}_t^v$ as the routing bias, directly encoding the relative reliability difference between the two modalities.

As shown in Table 2, CDD-MoLT (U-6) achieves the best overall WER under clean and mild noise conditions, attaining 0.77, 1.16, and 2.28 % at clean, SNR 5, and SNR 0 dB, respectively. Compared with DD-MoLT (Video, U-6), which ties on SNR 5 and SNR 0, the cross-modal variant additionally reduces clean WER from 0.90 to 0.77. Under heavy noise, however, DD-MoLT (Audio, U-6) remains superior (e.g., 19.82 vs. 20.92 at −15 dB), suggesting that the single-modal audio difficulty signal provides a stronger and more direct routing push when the audio is severely corrupted. Overall, these results indicate that the cross-modal approach effectively leverages the relative reliability difference between modalities to achieve superior performance across a wide range of conditions, while the single-modal audio variant retains an advantage under the most adverse noise.

4.5 Comparison with the state-of-the-art models

Performance comparison under clean conditions. Table 3 compares CDD-MoLT (U-6) with prior AVSR systems on the LRS3 test set under the clean condition. Trained only on LRS3 (433 h), our model attains a WER of 0.77%, already lower than all methods trained on comparable data, including the strong MMS-LLaMA (0.90%) and Llama-AVSR (0.95%) baselines. Adding VoxCeleb2 (1,759 h in total) further lowers the WER to 0.69%, the best among all compared systems, surpassing even those trained on orders of magnitude more data such as LP Conformer (0.9% with 100K h). These results indicate that leveraging the intermediate layers of the AV-HuBERT encoder benefits recognition performance.

Method	Audio Encoder	Visual Encoder	Training Data (h)	WER (%)
CM-seq2seq (Ma et al., 2021)	Conformer		433	2.3
AV-data2vec (Lian et al., 2023)	Transformer		433	2.5
ViT3D-CM (Serdyuk et al., 2022)	Conformer		90K	1.6
CMA (Kim et al., 2024)	Transformer		433	1.5
Auto-AVSR (Ma et al., 2023)	Conformer		1,902/3,448	1.0/0.9
LP Conformer (Chang et al., 2024)	Conformer		100K	0.9
Whisper-Flamingo (Rouditchenko et al., 2024)	Whisper	AV-HuBERT	433/1,759	1.1/0.76
Llama-AVSR (Cappellazzo et al., 2025a)	Whisper	AV-HuBERT	433/1,759	0.95/0.72
MMS-LLaMA (Yeo et al., 2025)	Whisper	AV-HuBERT	433/1,759	0.90/0.72
Ours (CDD-MoLT, U-6)	Whisper	AV-HuBERT	433/1,759	**0.77/0.69**

Table 3: Comparison with state-of-the-art AVSR methods on the LRS3 test set (clean condition). Training Data (h) denotes the total number of audio-visual hours used for training.

Method	WER (%) at SNR (dB)							
	clean	15	10	5	0	-5	-10	-15
Baseline	0.72	0.75	**0.76**	1.00	1.90	7.00	13.80	17.10
Baseline + Ours	**0.69**	**0.73**	0.77	**0.85**	**1.87**	**5.35**	**10.31**	**13.85**

Table 4: WER (%) on LRS3 comparing MMS-LLaMA (Baseline) and CDD-MoLT (U-6) across SNR conditions from +15 to −15 dB. Both models are trained on LRS3 and VoxCeleb2.

Comparison with Baseline and CDD-MoLT. Table 4 compares MMS-LLaMA (Yeo et al., 2025) and our CDD-MoLT (U-6) trained on LRS3 and VoxCeleb2 across SNR +15 to −15 dB, confirming consistent gains that widen under heavier noise.

4.6 Analysis

4.6.1 Activated layers

Activated layer distribution of S-MoLT. To identify which intermediate layers are useful for AVSR, we visualize the per-layer routing score distribution of Top-3 S-MoLT applied to AV-HuBERT (left panel of Figure 3) and Whisper (right panel).

Lower layers of AV-HuBERT are rarely selected, with layers in the index range 18–23 being predominantly chosen. Notably, layers 20–22 exhibit higher routing scores than the final layer. Although the routing score distribution varies somewhat across

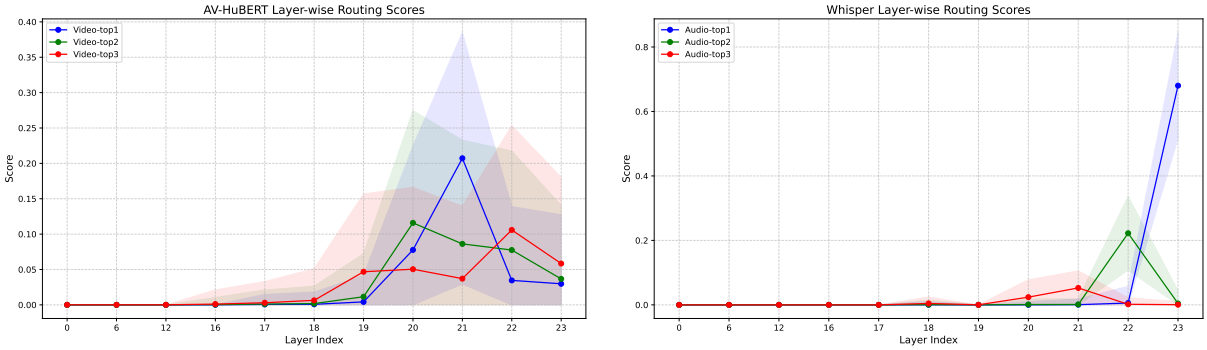


Figure 3: Per-layer routing score distribution of Top-3 S-MoLT applied to AV-HuBERT (left) and Whisper (right).

tokens, this suggests that certain intermediate layers of AV-HuBERT may contain information more useful for speech recognition than the final-layer representation alone, motivating the use of the upper-six-layer candidate set (U-6) in subsequent experiments.

In contrast, applying S-MoLT to the Whisper audio encoder (S-MoLT (ATop-3, U-6) in Table 2) consistently degrades WER across all conditions. As shown in the right panel of Figure 3, the routing scores of Whisper are heavily concentrated on the final layer, with intermediate layers receiving negligible scores. This suggests that, for the Whisper encoder, mixing intermediate layers may act as noise rather than provide useful information, and that the information relevant to speech recognition is most heavily concentrated in the final layer. This tendency is consistent with the analyses of (Yang et al., 2023; Peng et al., 2026). Pasad et al. (2023) show that the distribution of information across layers depends on the training objective; since Whisper is trained to perform speech recognition solely from its final-layer output, the information best suited for speech recognition may be concentrated in the final layer.

Number of activated layers of D-MoLT variants. Figure 4 shows the distribution of the average number of activated layers per token for D-MoLT and its variants. All variants activate between 4 and 6 out of the 6 upper-layer candidates on average. Plain D-MoLT and DD-MoLT (Video) exhibit the highest mean activation counts (5.38 and 5.34, respectively), while DD-MoLT (Audio) activates the fewest layers on average (4.04) with the smallest variance (0.36), indicating more selective and consistent routing. CDD-MoLT falls in between with a mean of 4.63, yet exhibits the largest variance (1.21), suggesting that the cross-modal difficulty

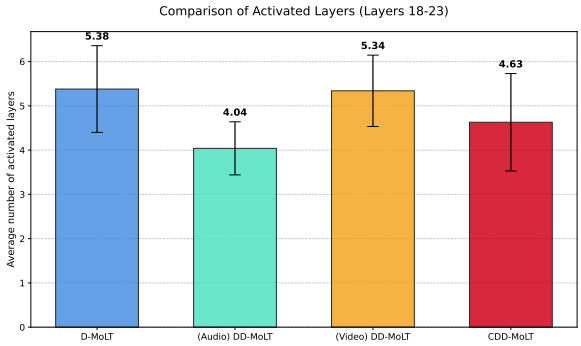


Figure 4: Distribution of the average number of activated layers per token for D-MoLT and its variants (U-6).

signal induces more diverse, input-adaptive routing behavior across tokens. Note that these statistics are measured under the U-6 setting; the U-24 configuration analyzed in Appendix F exhibits substantially sparser routing (close to one layer per token on average), as the larger candidate pool allows the threshold-based router to suppress most layers. Notably, despite activating fewer layers on average than plain D-MoLT, DD-MoLT (Audio) achieves substantially better WER under heavy noise conditions (Table 2). This reveals that the benefit of the difficulty-aware mechanism stems not from the total number of activated layers, but from routing each token to the task-appropriate layers, directing low-difficulty tokens toward lower layers and high-difficulty tokens toward upper layers; Appendix E visualizes how this per-layer activation shifts with SNR.

4.6.2 Distribution of Token Difficulty Scores

Figure 5 shows the distribution of the estimated difficulty scores across SNR conditions for the difficulty-adaptive variants (DD-MoLT and CDD-MoLT), together with their relationship to the LLM decoder's prediction confidence. As the input be-

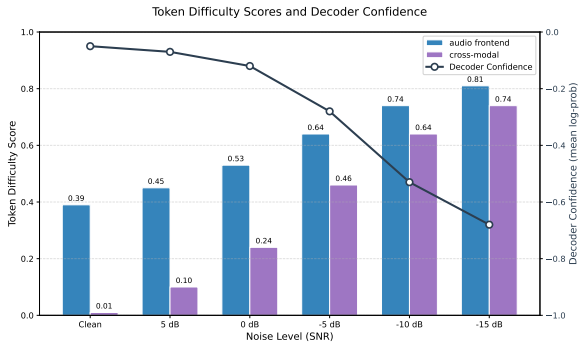


Figure 5: Distribution of token difficulty scores across SNR conditions and their relationship with the decoder's prediction confidence, for the difficulty-adaptive variants (DD-MoLT and CDD-MoLT). Values are averaged over tokens within each utterance, then across utterances.

comes noisier, the estimated difficulty score increases while the decoder's prediction confidence decreases. This indicates that, despite being trained without explicit score supervision, our Token Difficulty Estimator effectively captures the difficulty of the input speech.

To further verify that the estimated difficulty score reflects a meaningful recognition-related signal rather than a mere byproduct of the input SNR, we conduct two complementary analyses. First, as an internal-consistency check, the difficulty score $\hat{s}_t$ is negatively correlated with the decoder's own prediction confidence (pooled Spearman $\rho = -0.58$). Second, $\hat{s}_t$ also tracks recognition errors: pooled across all SNR levels, it correlates with utterance-level WER (Spearman $\rho = 0.43$–0.45) and separates error-containing from error-free utterances (AUROC 0.78–0.79). Crucially, this association persists even when the SNR is held fixed (within-level $\rho \approx 0.04$–0.10), indicating that the difficulty score is not a pure SNR proxy but carries a genuine per-utterance difficulty signal.

4.6.3 Efficiency Comparison: Baseline vs. MoLT variants

Table 5 reports the additional parameters and prefill FLOPs overhead of each MoLT variant on the LRS3 test set. Our MoLT is lightweight: even its largest variant, CDD-MoLT, introduces only about 6.8M additional parameters (0.16% of the baseline) and increases prefill FLOPs by at most 0.9%. Moreover, since the MoLT module runs only once before the AV Q-Former, the number of tokens passed to the LLM and the dominant LLM decoding cost remain unchanged; a wall-clock latency breakdown across routing strategies is provided in Appendix B.

Method	Added Params (% of baseline)	Prefill FLOPs (T)
Baseline (Yeo et al., 2025)	–	1.339
S-MoLT (Top-6, U-24)	4.2M (0.10%)	1.350 (+0.9%)
D-MoLT (U-6)	4.2M (0.10%)	1.341 (+0.2%)
DD-MoLT (Audio, U-6)	6.6M (0.15%)	1.344 (+0.5%)
DD-MoLT (U-6)	6.8M (0.16%)	1.345 (+0.5%)

Table 5: Parameter and prefill FLOPs overhead of the MoLT variants relative to the MMS-LLaMA baseline. Prefill FLOPs count all computation before the LLM decoder input and are averaged over all utterances; all percentages are relative to the baseline.

5 Conclusion

We proposed a Mixture of Layer-wise Tokens (MoLT) framework that exploits intermediate layer representations of the AV-HuBERT SSL encoder for LLM-based AVSR. Starting from a Top-k Sparse variant (S-MoLT), we introduced Dynamic routing-based MoLT (D-MoLT) to adaptively determine the number and combination of activated layers per token, and further proposed a token-difficulty-aware lower-upper preference mechanism. Experiments on LRS3 show that our best model CDD-MoLT outperforms the MMS-LLaMA baseline across nearly all SNR conditions, with the gain widening under severe noise (e.g., 13.85 vs. 17.10 % WER at -15 dB). Overall, these results demonstrate that selectively leveraging intermediate encoder layers is highly effective for robust LLM-based AVSR.

6 Limitations

Our proposed approach was not effective for the Whisper audio encoder. Since Whisper concentrates the information suited for speech recognition in its final layer, better strategies for effectively exploiting its intermediate-layer information remain to be explored. In addition, our experiments are confined to the AV-HuBERT and Whisper encoders, so the effectiveness of MoLT with other speech or visual encoders remains unverified. Nevertheless, our method outperforms the baseline across nearly all conditions, with the improvement being particularly pronounced under severely degraded acoustic conditions.

Acknowledgments

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT)(RS-2022-II220989(2022-0-00989), Development of Artificial Intelligence Technology for Multi-speaker Dialog Modeling)

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*.

Szymon Antoniak, Michał Krutul, Maciej Pióro, Jakub Krajewski, Jan Ludziejewski, Kamil Ciebiera, Krystian Król, Tomasz Odrzygóźdź, Marek Cygan, and Sebastian Jaszczur. 2024. Mixture of tokens: Continuous moe through cross-example aggregation. *Advances in Neural Information Processing Systems*, 37:103873–103896.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.

Umberto Cappellazzo, Minsu Kim, Honglie Chen, Pingchuan Ma, Stavros Petridis, Daniele Falavigna, Alessio Brutti, and Maja Pantic. 2025a. Large language models are strong audio-visual speech recognition learners. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Umberto Cappellazzo, Minsu Kim, Pingchuan Ma, Honglie Chen, Xubo Liu, Stavros Petridis, and Maja Pantic. 2026. Mome: Mixture of matryoshka experts for audio-visual speech recognition. *Advances in neural information processing systems*, 38:40013–40040.

Umberto Cappellazzo, Minsu Kim, Stavros Petridis, Daniele Falavigna, and Alessio Brutti. 2025b. Scaling and enhancing llm-based avsr: A sparse mixture of projectors approach. In *Interspeech 2025*, pages 1823–1827.

Oscar Chang, Hank Liao, Dmitriy Serdyuk, Ankit Shah, and Olivier Siohan. 2024. Conformer is all you need for visual speech recognition. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 10136–10140. IEEE.

Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. 2018. Voxceleb2: Deep speaker recognition. In *Interspeech 2018*, pages 1086–1090.

Jiankang Deng, Jia Guo, Evangelos Ververas, Irene Kotsia, and Stefanos Zafeiriou. 2020. Retinaface: Single-shot multi-level face localisation in the wild. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5202–5211. IEEE.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36:10088–10115.

Yassir Fathullah, Chunyang Wu, Egor Lakomkin, Junteng Jia, Yuan Shangguan, Ke Li, Jinxi Guo, Wenhan Xiong, Jay Mahadeokar, Ozlem Kalinli, and 1 others. 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.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Yongxin Guo, Zhenglin Cheng, Xiaoying Tang, Zhaopeng Tu, and Tao Lin. 2025. Dynamic mixture of experts: An auto-tuning approach for efficient transformer models. In *International Conference on Learning Representations*, volume 2025, pages 79643–79672.

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*.

Quzhe Huang, Zhenwei An, Nan Zhuang, Mingxu Tao, Chen Zhang, Yang Jin, Kun Xu, Liwei Chen, Songfang Huang, and Yansong Feng. 2024. Harder task needs more experts: Dynamic routing in moe models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12883–12895.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*.

Sungnyun Kim, Kangwook Jang, Sangmin Bae, Sungwoo Cho, and Se-Young Yun. 2025. MoHAVE: Mixture of hierarchical audio-visual experts for robust speech recognition. In *International Conference on Machine Learning*.

Sungnyun Kim, Kangwook Jang, Sangmin Bae, Hoirin Kim, and Se-Young Yun. 2024. Learning video temporal dynamics with cross-modal attention for robust audio-visual speech recognition. In *2024 IEEE Spoken Language Technology Workshop (SLT)*, pages 447–454. IEEE.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2021. Gshard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations*.

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.

Jiachen Lian, Alexei Baevski, Wei-Ning Hsu, and Michael Auli. 2023. Av-data2vec: Self-supervised learning of audio-visual speech representations with contextualized target representations. In *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1–8. IEEE.

Ilya Loshchilov and Frank Hutter. 2017. Sgdr: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*.

Pingchuan Ma, Alexandros Haliassos, Adriana Fernandez-Lopez, Honglie Chen, Stavros Petridis, and Maja Pantic. 2023. 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, 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.

Pingchuan Ma, Stavros Petridis, and Maja Pantic. 2022. Visual speech recognition for multiple languages in the wild. *Nature Machine Intelligence*, 4(11):930–939.

Ankita Pasad, Ju-Chieh Chou, and Karen Livescu. 2021. Layer-wise analysis of a self-supervised speech representation model. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 914–921. IEEE.

Ankita Pasad, Bowen Shi, and Karen Livescu. 2023. Comparative layer-wise analysis of self-supervised speech models. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Jian Peng, Lixu Sun, Yongchao Li, Yineng Cai, Nurmemet Yolwas, and Wushour Silamu. 2026. Hierarchical analysis and efficient fine-tuning of weakly supervised speech models. *Expert Systems with Applications*, page 132262.

Stavros Petridis, Themos Stafylakis, Pingehuan Ma, Feipeng Cai, Georgios Tzimiropoulos, and Maja Pantic. 2018. End-to-end audiovisual speech recognition. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 6548–6552. IEEE.

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.

David Raposo, Sam Ritter, Blake Richards, Timothy Lillicrap, Peter Conway Humphreys, and Adam Santoro. 2024. Mixture-of-depths: Dynamically allocating compute in transformer-based language models. *arXiv preprint arXiv:2404.02258*.

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 2024*, pages 2420–2424.

Dmitriy Serdyuk, Otavio Braga, and Olivier Siohan. 2022. Transformer-based video front-ends for audio-visual speech recognition for single and muti-person video. In *Interspeech 2022*, pages 2833–2837.

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. In *International Conference on Learning Representations*.

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*.

David Snyder, Guoguo Chen, and Daniel Povey. 2015. Musan: A music, speech, and noise corpus. *arXiv preprint arXiv:1510.08484*.

Andrew Varga and Herman JM Steeneken. 1993. Assessment for automatic speech recognition: Ii. noisex-92: A database and an experiment to study the effect of additive noise on speech recognition systems. *Speech communication*, 12(3):247–251.

Shu wen Yang, Po-Han Chi, Yung-Sung Chuang, Cheng-I Jeff Lai, Kushal Lakhotia, Yist Y. Lin, Andy T. Liu, Jiatong Shi, Xuankai Chang, Guan-Ting Lin, Tzu-Hsien Huang, Wei-Cheng Tseng, Ko tik Lee, Da-Rong Liu, Zili Huang, Shuyan Dong, Shang-Wen Li, Shinji Watanabe, Abdelrahman Mohamed, and Hung yi Lee. 2021. Superb: Speech processing universal

performance benchmark. In *Interspeech 2021*, pages 1194–1198.

Hao Yang, Jinming Zhao, Gholamreza Haffari, and Ehsan Shareghi. 2023. Investigating pre-trained audio encoders in the low-resource condition. In *Interspeech 2023*, pages 1498–1502.

Jeong Hun Yeo, Hyeongseop Rha, Se Jin Park, and Yong Man Ro. 2025. Mms-llama: Efficient llm-based audio-visual speech recognition with minimal multimodal speech tokens. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20724–20735.

Appendix

A Implementation Details

Pre-processing. Following Yeo et al. (2025), we apply an identical pre-processing pipeline to the LRS3 and VoxCeleb2 data. Audio and video are resampled to 16 kHz and 25 fps, respectively. For each frame, the face is localized with RetinaFace (Deng et al., 2020) and the mouth region is cropped to 96×96 pixels, converted to grayscale, and flipped horizontally for data augmentation during training. On the audio side, the NOISEX babble noise is mixed in before feature extraction, after which each utterance is zero-padded to 30 s and converted into an 80-dimensional log-Mel spectrogram before being fed to the Whisper encoder.

Training and evaluation. We use the same training and evaluation setup as Yeo et al. (2025). The LLM decoder is fine-tuned with QLoRA (Dettmers et al., 2023) (rank r=16, α=32, dropout 0.05) applied to the query, key, value, and output projection layers. Models are optimized with Adam (β_1=0.9, β_2=0.98, lr 1×10^{-4}) and a cosine learning rate schedule (Loshchilov and Hutter, 2017) with 0.5k warm-up steps over 30,000 total training steps, decaying to a minimum learning rate of 1×10^{-5} (final learning-rate scale 0.05). At inference, we decode with beam search (beam size 5) and a temperature of 0.3.

B Inference Latency

To complement the parameter and FLOPs overhead in Table 5, we measure the wall-clock inference latency of each routing strategy. Table 6 reports the median per-utterance latency over the full LRS3 test set (NVIDIA L40, batch size 1, beam size 5), with the isolated routing cost timed via CUDA events. The routing itself costs less than 1 ms per utterance ($\leq$ 0.22% of the end-to-end latency) across all variants, confirming that the different routing strategies are comparably cheap. The end-to-end latency is instead dominated by LLM decoding: since the number of generated tokens is identical (9.9 on average) and no additional tokens are passed to the LLM, the small spread in the end-to-end medians (421–444 ms) reflects run-to-run variation rather than a systematic difference between models, and the 421 ms of CDD-MoLT should not be read as a genuine speed-up over the baseline.

Method	End-to-end (ms)	Routing (ms)	Gen. tokens
Baseline (Yeo et al., 2025)	428	–	9.9
S-MoLT (Top-6, U-24)	444	0.47 (0.11%)	9.9
D-MoLT (U-6)	444	0.55 (0.12%)	9.9
DD-MoLT (Audio, U-6)	443	0.85 (0.19%)	9.9
CDD-MoLT (U-6)	421	0.91 (0.22%)	9.9

Table 6: Median per-utterance inference latency on the LRS3 test set (NVIDIA L40, batch size 1, beam size 5). **End-to-end** is the full recognition time; **Routing** is the isolated MoLT routing cost timed via CUDA events, with its percentage of the end-to-end latency in parentheses; **Gen. tokens** is the average number of generated tokens. End-to-end medians are dominated by LLM decoding, so their spread is run-to-run noise rather than a real difference.

C Generalization to New Corpora and Unseen Noise

To verify that the robustness of our MoLT variants is not confined to the LRS3 (Afouras et al., 2018b) corpus and the NOISEX (Varga and Steeneken, 1993) noise used for training, we conduct two generalization experiments: generalization to a different corpus by retraining on LRS2 (Afouras et al., 2018a), and robustness to noise types unseen during training.

Cross-corpus generalization on LRS2. We retrain and evaluate the full pipeline on LRS2, using the same NOISEX babble noise setup to simulate the SNR conditions. As shown in Table 7, our best difficulty-adaptive variants, DD-MoLT (U-6) and CDD-MoLT (U-6), outperform the baseline across all SNR levels, and CDD-MoLT (U-6) achieves the largest gains under heavy noise. At -15 dB, CDD-MoLT reduces WER from 49.35 to 34.35, a 30.4% relative reduction that is even larger than in the in-domain LRS3 setting, indicating that the layer-wise routing mechanism generalizes across corpora.

Zero-shot robustness to unseen noise on LRS3. The baseline and our models are trained on LRS3 with NOISEX babble noise. Without any retraining, we evaluate them on two MUSAN (Snyder et al., 2015) noise categories, music and natural, that are absent from training. As shown in Table 8, each variant's WER on the unseen music and natural noise closely tracks its in-domain babble curve, indicating that the improvements reflect genuine noise robustness rather than overfitting to the training noise.

D Comparison with Existing MoE-based Approaches

Positioning. Existing MoE-based AVSR systems apply MoE within the decoder or the LLM and operate on the encoder's final-layer output alone, whereas MoLT routes over the intermediate layers of the encoder. The two directions are therefore architecturally orthogonal and, in principle, complementary.

Quantitative comparison. Table 9 compares our variants against published LRS3 numbers from representative MoE-based and recent AVSR systems. Under the shared NOISEX-babble protocol, our variants attain the lowest WER at every SNR level shown. Against MoHAVE-Large, the MoE-AVSR system with the most complete per-SNR results, our models are substantially more robust under heavy noise (6.45 vs. 9.8 at -5 dB; 13.93 vs. 21.0 at -10 dB). We caution that differences in noise source and decoder architecture preclude a strictly controlled comparison.

E Per-Layer Activation across SNR

Figure 6 reports, for each upper-layer candidate (L18–L23) under the U-6 setting, the fraction of tokens that activate it as a function of SNR, for D-MoLT, DD-MoLT (Audio), and CDD-MoLT. For plain D-MoLT, the per-layer activation ratios are essentially flat across SNR: without difficulty guidance, the router applies the same layer preference regardless of the acoustic condition. In contrast, both difficulty-adaptive variants shift their layer usage with noise: as the SNR decreases, activation of the uppermost layers (L22, L23) rises toward 100% while that of the lower candidates (L18, L19) drops. This directly visualizes the intended behavior of the lower-upper preference, where clean, easy tokens rely more on lower layers that preserve fine-grained cues, whereas noisy, hard tokens are routed toward upper layers rich in contextual information.

Method	WER (%) at SNR (dB)					
	clean	5	0	-5	-10	-15
Baseline (Yeo et al., 2025)	2.10	3.06	6.89	16.35	34.47	49.35
SUPERB (wen Yang et al., 2021) (U-24)	2.04	2.82	5.93	17.45	35.99	48.33
SUPERB (wen Yang et al., 2021) (U-6)	2.25	2.97	6.52	18.06	37.36	49.22
S-MoLT (Video Top-24, U-24)	2.16	2.69	5.60	16.16	33.63	44.44
S-MoLT (Video Top-3, U-24)	2.09	2.91	5.69	17.21	36.02	49.38
D-MoLT (U-24)	2.13	2.75	5.18	13.78	27.03	37.24
D-MoLT (U-6)	**1.95**	2.76	5.69	15.80	29.64	40.08
DD-MoLT (Audio-frontend, U-24)	2.28	3.00	5.75	17.25	35.95	48.35
DD-MoLT (Audio-frontend, U-6)	2.04	**2.64**	5.39	15.08	31.10	40.24
DD-MoLT (Video-frontend, U-24)	2.15	2.96	5.69	15.15	28.84	41.29
DD-MoLT (Video-frontend, U-6)	2.18	2.84	5.72	15.17	30.77	41.26
CDD-MoLT (U-24)	2.00	2.84	5.72	15.95	31.19	44.23
CDD-MoLT (U-6)	2.01	2.67	**4.86**	**13.26**	**25.72**	**34.35**

Table 7: Cross-corpus generalization: WER (%) on the LRS2 test set after retraining the full pipeline on LRS2.

Noise	Method	WER (%) at SNR (dB)				
		5	0	-5	-10	-15
NOISEX babble (in-domain)	Baseline	1.37	2.83	7.85	17.41	26.67
	DD-MoLT (Audio, U-6)	1.32	2.55	**6.45**	**13.93**	**19.82**
	CDD-MoLT (U-6)	**1.16**	**2.28**	6.78	15.06	20.92
MUSAN music (unseen)	Baseline	1.35	3.01	8.36	18.87	26.74
	DD-MoLT (Audio, U-6)	1.35	2.98	**6.96**	**14.26**	**19.93**
	CDD-MoLT (U-6)	**1.18**	**2.73**	7.24	15.29	21.42
MUSAN natural (unseen)	Baseline	1.39	2.97	8.13	17.97	26.39
	DD-MoLT (Audio, U-6)	1.37	2.73	7.14	**13.80**	**19.80**
	CDD-MoLT (U-6)	**1.20**	**2.65**	**7.08**	14.79	21.14

Table 8: Zero-shot robustness to unseen MUSAN noise. Models trained on LRS3 with NOISEX babble noise are evaluated, without retraining, on unseen music and natural noise. Clean is omitted, as MUSAN has no noise-free condition.

Method	MoE position	WER (%) at SNR (dB)					
		clean	5	0	-5	-10	-15
MoHAVE-Large† (Kim et al., 2025)	decoder FFN	1.5	2.2	4.1	9.8	21.0	–
MoME (Cappellazzo et al., 2026)	in frozen LLM	0.9	–	–	–	–	–
Llama-SMoP (Cappellazzo et al., 2025b)	projector	2.29	6.1	9.5	–	–	–
Llama-AVSR (Cappellazzo et al., 2025a)	–	0.9	2.2	4.2	16.9	–	–
MMS-LLaMA (Yeo et al., 2025)	–	0.95	1.30	2.66	7.44	–	–
DD-MoLT (Audio, U-6)	encoder layers	0.90	1.32	2.55	**6.45**	**13.93**	**19.82**
CDD-MoLT (U-6)	encoder layers	**0.77**	**1.16**	**2.28**	6.78	15.06	20.92

Table 9: WER (%) on the LRS3 test set; all models are trained on 433 h. Best result in each column is in **bold**. †MoHAVE uses MUSAN babble and a non-LLM decoder (1.0B/553M active parameters); all other systems use NOISEX babble. MoME reports noise results only on LRS2, hence its blank cells.

F Auxiliary Losses for D-MoLT

In the absence of explicit constraints, the threshold-based router of D-MoLT may in principle collapse onto a small subset of layers or, conversely, activate nearly all layers indiscriminately. Prior MoE works address such pathologies by introducing auxiliary objectives that regularize the routing distribution: load balancing (Shazeer et al., 2017; Fedus et al., 2022) discourages routing collapse by promoting uniform expert utilization, while DynMoE (Guo et al., 2025) employs diversity and simplicity losses to encourage diverse selection patterns and to suppress unnecessary activations. As discussed in the main text, these objectives are not directly aligned with our goal of guiding each token to select the layers genuinely necessary for AVSR; uniform layer usage in particular conflicts with the layer-wise specialization observed in AV-HuBERT, where only a subset of upper layers carries information useful for recognition. Nevertheless, to empirically verify

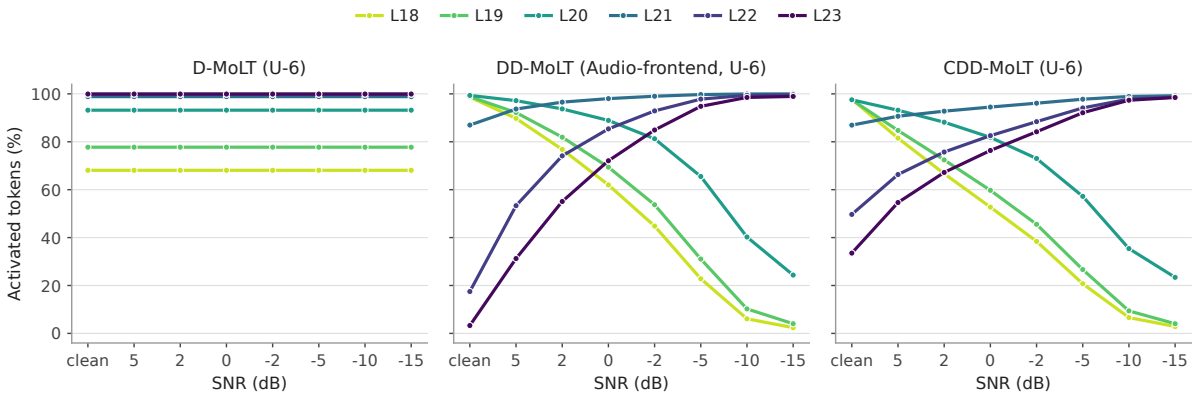


Figure 6: Per-layer activation ratio of the upper-six candidate layers (L18–L23) as a function of SNR, under the U-6 setting. Left: D-MoLT; middle: DD-MoLT (Audio); right: CDD-MoLT. D-MoLT is noise-agnostic (flat curves), whereas the difficulty-adaptive variants shift activation toward the uppermost layers as noise increases.

whether such regularizers help or hurt D-MoLT in the AVSR setting, we consider the following three losses and report their formulations and effects below.

F.1 Formulations

Load Balancing Loss. This loss encourages uniform distribution of per-layer activation rates across all tokens in a batch, and is defined as:

$$\mathcal{L}_{\text{balance}} = L \cdot \sum_{l=1}^{L} f_l \cdot p_l$$

where f_l denotes the fraction of tokens for which layer l is activated, and p_l denotes the mean routing probability for layer l, defined respectively as:

$$f_l = \frac{1}{BT} \sum_{b=1}^{B} \sum_{t=1}^{T} m_{b,t}^{l}, \quad p_l = \frac{1}{BT} \sum_{b=1}^{B} \sum_{t=1}^{T} \sigma(s_{b,t}^{l})$$

Diversity Loss. This loss encourages diverse selection patterns across layers by promoting different layers to be activated by different subsets of tokens. Defining the soft gate matrix $G \in \mathbb{R}^{BT \times L}$ as $G_{(b,t),l} = \sigma(s_{b,t}^{l})$ and its column-normalized counterpart $\tilde{G}$, the loss is computed as:

$$\mathcal{L}_{\text{div}} = \left\| \tilde{G}^{\top} \tilde{G} - I \right\|_F$$

As the column-wise cosine similarities approach the identity matrix, each layer exhibits an increasingly independent selection pattern.

Simplicity Loss. This loss suppresses overall activation by reducing the number of tokens that activate each layer, and is defined as the mean ℓ_2 norm of the column vectors $\mathbf{g}_l \in \mathbb{R}^{BT}$ of the soft gate matrix G:

$$\mathcal{L}_{\text{simple}} = \frac{1}{L} \sum_{l=1}^{L} \|\mathbf{g}_l\|_2$$

For $\mathcal{L}_{\text{balance}}$, f_l is computed by detaching the binary gate $m_{b,t}^{l}$ to block gradient flow, such that gradients propagate exclusively through p_l. For $\mathcal{L}_{\text{div}}$ and $\mathcal{L}_{\text{simple}}$, since the binary gate m_t^l does not carry valid column-wise gradients due to the STE, both losses are computed based on the continuous soft gate values $\sigma(s_{b,t}^{l})$.

Method	WER (%) at SNR (dB)					
	clean	5	0	-5	-10	-15
Baseline	1.00	1.37	2.83	7.85	17.41	26.67
D-MoLT	**0.93**	**1.25**	**2.48**	7.53	16.38	23.27
D-MoLT + lb	0.95	1.39	2.55	7.90	18.14	26.72
D-MoLT + div	1.09	1.50	2.62	7.32	16.32	24.42
D-MoLT + simp	**0.93**	1.30	2.50	**7.28**	**15.19**	**22.00**
D-MoLT + div + simp	0.98	1.48	2.52	9.17	18.35	25.18

Table 10: WER (%) on LRS3 for D-MoLT (U-24) with and without auxiliary losses. lb: load balancing, div: diversity, simp: simplicity.

F.2 Empirical Results on D-MoLT

We attach each of the three auxiliary losses to D-MoLT (U-24) and evaluate WER on the LRS3 test set; results are summarized in Table 10.

Load balancing loss causes consistent performance degradation relative to plain D-MoLT, particularly under noisy conditions. This is likely because encouraging every layer to be activated uniformly across tokens is fundamentally misaligned with the AVSR task, in which only a small sub-

set of upper layers carries information useful for recognition.

Diversity loss plays a similar role to load balancing by promoting independent per-layer selection patterns, and accordingly shows a comparable trend of performance degradation across most SNR conditions.

Simplicity loss yields results comparable to, or occasionally marginally better than, plain D-MoLT at mid-to-low SNR levels (e.g., -5, -10, -15 dB in Table 10). This can be attributed to the fact that D-MoLT (U-24) already converges to a very sparse routing, activating on average close to one layer per token; thus, the additional suppression imposed by the simplicity loss does not substantially alter the routing behavior. Crucially, however, the best overall results in Table 2 are achieved by DD-MoLT variants, which adaptively activate *more* layers depending on input difficulty. This observation suggests that simply reducing the number of activated layers is not always optimal; rather, activating more layers when the input conditions demand it, as enabled by the difficulty-aware preference mechanism, is key to improving AVSR robustness under adverse acoustic conditions.

G Use of AI Assistants

The authors employed Claude exclusively as a language-editing tool to polish grammar and writing style when preparing the final (camera-ready) version of this manuscript. It was not used in the initial submission and played no role in generating the scientific hypotheses, conducting the analyses, or developing the intellectual content, all of which originate from the human authors.