LLaVA-MoR: Adaptive Visual Recursion for Multimodal Reasoning

Anonymous authors
Paper under double-blind review

Abstract

Multimodal Large Language Models (MLLMs) such as LLaVA have demonstrated promising capabilities in visual understanding and instruction following. Although allocating reasoning budgets according to problem difficulty has achieved better accuracy-efficiency trade-offs in text reasoning, whether and how to distribute adaptive budgets across visual tokens remains an open question in multimodal scenarios. In this work, we introduce LLaVA-MoR, the first approach to allocate question-conditioned recursion budgets to visual tokens in instruction-tuned MLLMs, optimizing multimodal reasoning by effectively preventing the model from overthinking simple visual regions and under-analyzing complex ones. To achieve this, we first propose the Question-Conditioned Visual Router, which explicitly models question–visual interactions to route visual tokens to different recursion depths. Furthermore, we propose Cosine-Guided Multi-Depth Adaptation, which gradually increases exposure to deeper recursion under answer supervision to adapt the backbone for visual processing while preserving its language capabilities. Extensive experiments demonstrate that LLaVA-MoR achieves a superior accuracy-efficiency trade-off across multiple backbones, even outperforming fixed maximum-depth recursion. These results highlight the critical value of token-level budget allocation in multimodal reasoning tasks. Code is available at `https://anonymous.4open.science/r/LLaVA_MoR/`.

1 Introduction

Multimodal Large Language Models (MLLMs) such as LLaVA have demonstrated promising capabilities in visual understanding and instruction following (Liu et al., 2023; 2024b). While allocating computation according to task difficulty has been shown to improve the accuracy-efficiency trade-off in text reasoning (Xiao et al., 2026), how to dynamically distribute compute budgets across visual tokens in multimodal scenarios remains an open question. Unlike pure text reasoning, answering multimodal questions requires dynamically integrating visual evidence with textual conditions and relevant knowledge (Liu et al., 2023; Lu et al., 2022). Because the required visual evidence varies across questions involving different objects, attributes, and relationships within an image (Anderson et al., 2018; Hudson & Manning, 2019), the corresponding visual tokens naturally demand varying degrees of reasoning. Therefore, we investigate whether and how assigning adaptive reasoning budgets to visual tokens can enhance multimodal reasoning and its accuracy-efficiency trade-off in this work.

To this end, conditional computation methods enable token-level allocation through expert selection or selective layer execution (Lin et al., 2026; Jing et al., 2026; Guo et al., 2025; Luo et al., 2025). Recursive computation methods enable further processing through repeated hidden-state updates using shared parameters (Saunshi et al., 2025). For instance, Mixture-of-Recursions (MoR) uses token-level routing to assign different recursion depths to individual tokens (Bae et al., 2025b). However, neither conditional nor recursive computation mechanisms ensure that the resulting allocation truly benefits multimodal reasoning. Specifically, these methods inherently allocate excessive computation to simple visual patches while providing insufficient processing for patches containing complex details or relationships. To overcome this limitation, we propose LLaVA-MoR, which uses question-conditioned adaptive recursion to focus reasoning budgets on visual patches that require iterative refinement.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

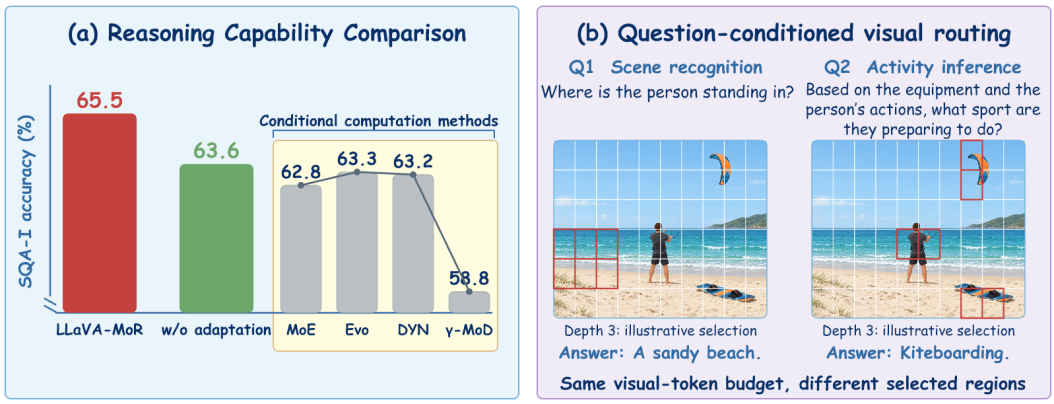


Figure 1: **Motivation for adaptive visual recursion.** LLaVA-MoR demonstrates two key advantages over existing multimodal architectures: **(a) Mitigating capability degradation with parameter-shared recursion:** On SQA-I, LLaVA-MoR achieves superior performance compared to both the unadapted baseline (w/o Stage I adaptation) and existing conditional computation methods (MoE-LLaVA, EvoMoE, DYNMOE, γ-MoD). **(b) Question-dependent visual selection:** Under the exact same visual-token budget, LLaVA-MoR dynamically routes deeper computation to query-relevant regions, selecting the beach for scene recognition (Q1) and the kite, person, and board for activity inference (Q2).

However, implementing such allocation requires addressing two challenges, illustrated in Figure 1. **(1) Capability degradation under parameter-shared recursion.** Reusing the same parameters across recursion depths requires shared blocks to serve multiple stages of processing, which can compromise the model's pretrained capabilities (Bae et al., 2025a). The key to addressing this challenge lies in adaptively allocating shared parameters to recursive computation while avoiding degradation in multimodal task performance. **(2) Question-agnostic visual budget allocation.** A region's visual complexity alone does not determine its processing demands; its role in answering the question also matters. However, vanilla MoR (Bae et al., 2025b) lacks a routing design tailored for multimodal visual scenarios, leaving visual token selection independent of question–visual interactions. Determining which visual tokens should receive deeper recursion thus requires a routing mechanism that explicitly incorporates the question.

LLaVA-MoR addresses these challenges through two training stages. **Cosine-Guided Multi-Depth Adaptation (Stage I)** gradually increases exposure to deeper recursion while updating the shared blocks and multimodal projector under answer supervision. Its cosine schedule changes depth-sampling probabilities slowly near the beginning and end of the transition, providing a gradual adaptation from shallow to multi-depth computation. Building on the adapted backbone, **Question-Conditioned Visual Router (Stage II)** learns question–visual interactions to select tokens for further recursion, with the backbone frozen. Selected visual tokens receive additional shared-block updates, while unselected tokens exit and retain their last hidden states.

Extensive experiments across multiple language backbones demonstrate that LLaVA-MoR achieves a superior reasoning performance and inference-cost trade-off without expanding the language backbone. Crucially, question-driven token selection achieves higher reasoning accuracy with substantially lower computational overhead, outperforming the allocation strategy that assigns all visual tokens to the full depth. Furthermore, qualitative visualizations illustrate that our routing mechanism assigns question-dependent recursion budgets to different visual regions, allowing informative areas to receive deeper reasoning while irrelevant ones exit early to suppress visual overthinking.

2 Preliminaries

2.1 Large Language and Vision Assistant

Large Language and Vision Assistant (LLaVA) is an autoregressive vision-language model that connects a pretrained vision encoder to a language model through a lightweight multimodal projector (Liu et al., 2023). In this paper, we denote a training sample by (X_v, X_q, X_a), comprising an image

X_v, a question X_q, and a reference answer X_a. For simplicity, we omit the sample index. Following LLaVA-1.5 (Liu et al., 2024b), the visual encoder g extracts image features, and a trainable MLP projector f_{proj} maps each feature into the language-model embedding space:

$$Z_v = g(X_v), \qquad H_v = f_{\text{proj}}(Z_v). \tag{1}$$

These projected visual features serve as visual tokens and are concatenated with the text embeddings to form a unified multimodal instruction. The concatenated training sequence is formulated as:

$$H_{\text{mm}} = [E(X_q); H_v; E(X_a)], \tag{2}$$

where E is the text embedding function and $[\cdot;\cdot]$ denotes concatenation. The model is supervised to predict the answer tokens X_a autoregressively.

2.2 Recursive Computation

Recursive Transformers reuse a shared stack of layers across depth. Following the Middle-Cycle sharing strategy of MoR (Bae et al., 2025b), we use an input block F_{in}, a shared intermediate block $F(\cdot;\theta)$ with parameters θ, and an output block F_{out} that includes the language-model prediction head. The input and output blocks are applied once. Let $\mathcal{D} = \{1, \ldots, R\}$ denote the supported recursion depths, with maximum depth R. Given the multimodal embedding sequence H_{mm} defined above, all valid tokens in a sample with depth $D \in \mathcal{D}$ undergo D shared-block applications:

$$\begin{aligned}
\mathbf{H}^{(0)} &= F_{\text{in}}(H_{\text{mm}}), \\
\mathbf{H}^{(d)} &= F\left(\mathbf{H}^{(d-1)};\theta\right) = \mathbf{H}^{(d-1)} + \Delta F\left(\mathbf{H}^{(d-1)};\theta\right), \quad d = 1, \ldots, D, \\
\mathbf{O}_{\text{out}} &= F_{\text{out}}\left(\mathbf{H}^{(D)}\right).
\end{aligned} \tag{3}$$

Here, $\mathbf{H}^{(0)}$ is the input block's output, and $\mathbf{H}^{(d)} \in \mathbb{R}^{L\times h}$ is the hidden-state sequence after recursion d, where L is the valid sequence length and h is the hidden dimension. For a hidden-state sequence $\mathbf{H}$, $\Delta F(\mathbf{H};\theta) = F(\mathbf{H};\theta) - \mathbf{H}$ is the block's net residual update. The parameters θ are shared across recursions, and $\mathbf{O}_{\text{out}}$ denotes the output token logits.

3 Method

3.1 Architecture Design

Our method LLaVA-MoR converts the language backbone of a dense instruction-tuned LLaVA model (Liu et al., 2024b) into a parameter-shared recursive architecture. Figure 2 summarizes its two-stage training procedure; initialization details are provided in Appendix B.1.

Stage I: Cosine-Guided Multi-Depth Adaptation. We jointly train the multimodal projector and language backbone while freezing the vision encoder and visual router. A cosine schedule gradually shifts the depth-sampling distribution from shallow to multi-depth computation. Each sample independently draws a recursion depth, which is shared by all its valid tokens.

Stage II: Question-Conditioned Visual Router. We freeze the adapted model and train only the visual router. At each recursive depth, the router combines question and visual states to select visual tokens for further computation. Unselected visual tokens exit early while keeping their hidden states unchanged.

3.2 Cosine-Guided Multi-Depth Adaptation

Recursive parameter sharing requires the same blocks to serve multiple processing depths, potentially affecting pretrained capabilities (Bae et al., 2025a). Our adaptation stage aims to limit degradation in multimodal task performance while training the shared backbone for recursive computation. We introduce **Cosine-Guided Multi-Depth Adaptation**, which initially assigns a single recursion to most samples and gradually shifts the depth-sampling probabilities toward a predefined target distribution. The router remains frozen, and all valid tokens in each sample share the sampled depth.

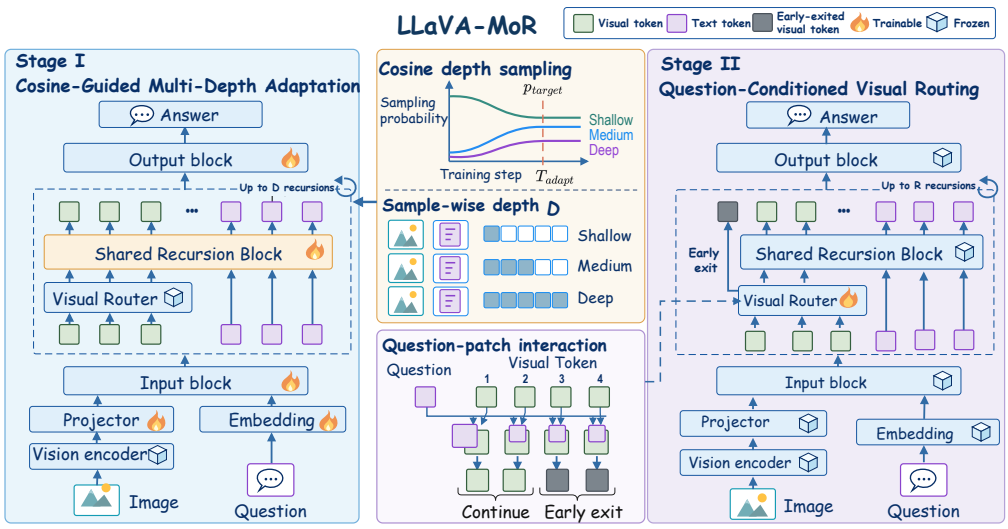


Figure 2: Overview of LLaVA-MoR. Stage I uses cosine-guided sample-wise depth sampling to allocate recursive depths to the shared blocks with the visual router frozen. Stage II freezes the adapted backbone and trains a visual router to select question-relevant visual patches for deeper recursion. Unselected visual tokens exit early while keeping their hidden states unchanged.

Let t denote the training step and $T_{\text{adapt}} > 0$ denote the transition duration:

$$\begin{aligned} \alpha_t &= \min\left(1, \frac{t}{T_{\text{adapt}}}\right), \\ \mathbf{p}_t &= \mathbf{p}_{\text{target}} + \frac{1}{2}\left[1 + \cos(\pi\alpha_t)\right](\mathbf{p}_{\text{start}} - \mathbf{p}_{\text{target}}), \\ D &\sim \text{Categorical}(\mathbf{p}_t). \end{aligned} \tag{4}$$

Here, t is the training step, $T_{\text{adapt}} > 0$ is the transition duration, and α_t is the normalized progress. The depth distribution $\mathbf{p}_t = (p_{t,1}, \ldots, p_{t,R})$ interpolates between the predefined initial and target distributions $\mathbf{p}_{\text{start}}$ and $\mathbf{p}_{\text{target}}$, reaching the latter at $t \geq T_{\text{adapt}}$. For each sample, a recursion depth $D \in \{1, \ldots, R\}$ is drawn independently with probability $P(D = d) = p_{t,d}$, under which all valid visual and text tokens undergo D recursions.

3.3 Question-Conditioned Visual Router

After backbone adaptation, we train the visual router to select visual tokens for further recursion. Let $\mathcal{A}^{(d)}$ denote the indices of visual tokens active at recursion d, with $\mathcal{A}^{(1)}$ containing all N_v visual tokens. Each recursion selects tokens only from the preceding active set:

$$\mathcal{A}^{(R)} \subseteq \cdots \subseteq \mathcal{A}^{(1)}.$$

Unselected visual tokens exit the shared block and retain their last hidden states.

Question-conditioned scoring. To select question-relevant visual tokens for deeper recursion, the router explicitly models question–visual interactions. Let $\mathcal{Q}$ denote the set of valid question-token positions. After recursion d, we mean-pool the normalized question states and take their element-wise product with each normalized active visual-token state. A learned linear projection followed by a sigmoid function produces a score for selecting tokens and weighting their updates at the next recursion:

$$\begin{aligned} \bar{\mathbf{h}}_q^{(d)} &= \frac{1}{|\mathcal{Q}|}\sum_{j\in\mathcal{Q}} \text{LN}\left(\mathbf{h}_{q,j}^{(d)}\right), \\ r_i^{(d)} &= \sigma\left(\mathbf{w}_d^\top\left[\text{LN}\left(\mathbf{h}_{v,i}^{(d)}\right) \odot \bar{\mathbf{h}}_q^{(d)}\right] + b_d\right), \quad i \in \mathcal{A}^{(d)}. \end{aligned} \tag{5}$$

Here, $\mathbf{h}_{q,j}^{(d)}$ and $\mathbf{h}_{v,i}^{(d)}$ are the question and visual states in $\mathbf{H}^{(d)}$, and $\bar{\mathbf{h}}_q^{(d)}$ is the pooled question vector, all in $\mathbb{R}^h$. The operators LN, $\odot$, and σ denote layer normalization, elementwise multiplication, and sigmoid. Each transition has trainable parameters $\mathbf{w}_d \in \mathbb{R}^h$ and $b_d \in \mathbb{R}$. The score $r_i^{(d)} \in (0,1)$ controls both selection and update weighting.

Token selection and update. Following expert-choice routing (Zhou et al., 2022), we select the highest-scoring tokens from the current active set:

$$\mathcal{A}^{(d+1)} = \underset{i \in \mathcal{A}^{(d)}}{\mathrm{TopK}} \left(r_i^{(d)}, K_{d+1} \right), \tag{6}$$

Here, TopK returns the indices of the K_{d+1} highest-scoring tokens in $\mathcal{A}^{(d)}$, where $K_{d+1} \leq |\mathcal{A}^{(d)}|$ is the visual-token budget for the next recursion. Selected tokens receive score-weighted residual updates:

$$\mathbf{h}_{v,i}^{(d+1)} = \begin{cases} \mathbf{h}_{v,i}^{(d)} + r_i^{(d)} \Delta F\big(\mathbf{H}^{(d)}; \theta\big)_i, & i \in \mathcal{A}^{(d+1)}, \\ \mathbf{h}_{v,i}^{(d)}, & \text{otherwise.} \end{cases} \tag{7}$$

The term $\Delta F(\mathbf{H}^{(d)}; \theta)_i$ is the shared-block residual update defined in Eq. 3, evaluated for visual token i using the selected visual tokens and active text tokens. Exited visual tokens retain their states and will not be updated in later recursions.

3.4 Training Objective

Let $X_a = (x_{a,1}, \ldots, x_{a,T_a})$ be the ground-truth answer of length T_a, where $x_{a,k}$ denotes the k-th answer token. The answer-only, length-normalized cross-entropy loss is defined as:

$$\mathcal{L}_{\text{answer}} = -\frac{1}{T_a} \sum_{k=1}^{T_a} \log \Pr(x_{a,k} \mid X_v, X_q, x_{a,<k}), \tag{8}$$

where $x_{a,<k}$ denotes preceding ground-truth answer tokens and Pr is the model's next-token distribution. Stage I trains the multimodal projector and language backbone using this loss.

For router optimization, let $m_i^{(d)} = \mathbb{I}[i \in \mathcal{A}^{(d+1)}]$ indicate whether visual token i is selected for the next recursion, and let $N_{\text{route}} = \sum_{d=1}^{R-1} |\mathcal{A}^{(d)}|$ be the total number of visual tokens scored across routing steps, counting each token once per recursion depth. Following MoR (Bae et al., 2025b), we use these selections as targets for an auxiliary binary cross-entropy:

$$\begin{aligned} \mathcal{L}_{\text{aux}} &= -\frac{1}{N_{\text{route}}} \sum_{d=1}^{R-1} \sum_{i \in \mathcal{A}^{(d)}} \left[m_i^{(d)} \log r_i^{(d)} + \left(1 - m_i^{(d)}\right) \log \left(1 - r_i^{(d)}\right) \right], \\ \mathcal{L}_{\text{route}} &= \mathcal{L}_{\text{answer}} + \eta \mathcal{L}_{\text{aux}}. \end{aligned} \tag{9}$$

Here, $\mathbb{I}[\cdot]$ is the indicator function and η weights the auxiliary loss. Stage II optimizes only the router parameters $\phi = \{(\mathbf{w}_d, b_d)\}_{d=1}^{R-1}$ using $\mathcal{L}_{\text{route}}$.

4 Experiments

4.1 Experiment Settings

Benchmarks. Following the data organization of MoE-LLaVA (Lin et al., 2026), we use LLaVA-PT-558K, Hybrid-FT-964K, and LLaVA-1.5-mix-665K (Liu et al., 2024b) for training. Hybrid-FT-964K combines SViT, LVIS-Instruct4V, LRV-Instruction, and MIMIC-IT (Zhao et al., 2023; Wang et al., 2023; Liu et al., 2024a; Li et al., 2023a). Evaluation covers VQAv2 (Goyal et al., 2017), GQA (Hudson & Manning, 2019), VizWiz (Gurari et al., 2018), ScienceQA-IMG (Lu et al., 2022), and TextVQA (Singh et al., 2019) for visual question answering, alongside POPE (Li et al., 2023b), MME (Fu et al., 2025), and MMBench (Liu et al., 2024c) for multimodal evaluation. We report POPE F1, MME perception scores, and the official metrics for other benchmarks.

Implementation Details. Following the data organization of MoE-LLaVA (Lin et al., 2026), we obtain dense LLaVA models through vision–language alignment on LLaVA-PT-558K and multimodal instruction tuning on Hybrid-FT-964K and LLaVA-1.5-mix-665K (Liu et al., 2024b). Building on the dense LLaVA architectures, we introduce parameter-shared recursion to further train the models in two stages using LLaVA-1.5-mix-665K. In Stage I, Cosine-Guided Multi-Depth Adaptation updates the language backbone and multimodal projector while keeping the visual router frozen. In Stage II, we freeze the adapted backbone and projector and train only the visual router to select visual tokens for deeper recursion. The vision encoder remains frozen throughout training.

Model configuration. We use Qwen-1.8B, StableLM-2-1.6B, Phi-2-2.7B, and Vicuna-7B as the backbone large language models (Bai et al., 2023; Bellagente et al., 2024; Javaheripi et al., 2023; Zheng et al., 2023). All models use CLIP ViT-L/14-336 as the vision encoder (Radford et al., 2021) and a two-layer MLP projector with GELU activation (Hendrycks & Gimpel, 2023). In our default configuration, we set the maximum recursion depth to $R = 3$ and keep 50% and 20% of the preceding active visual tokens at recursion depths 2 and 3, respectively. Both training stages use the AdamW optimizer (Loshchilov & Hutter, 2019) with an effective batch size of 64. Stage I uses a learning rate of 5×10^{-6} for one epoch, while Stage II uses a learning rate of 10^{-5} for half an epoch. The auxiliary routing-loss coefficient is set to $\eta = 0.01$ in Stage II. Additional training, evaluation, and baseline comparison details are provided in Appendices C and D.

4.2 Main Results across Language Backbones

Table 1: **Comparison of different LVLMs across four language backbones.** "Param" denotes language-backbone parameters in billions. SQA-I, VQA-T, and MMB denote ScienceQA-IMG, TextVQA, and MMBench, respectively. POPE reports F1, and MME reports perception scores. Within each backbone group, **bold** and underlined scores indicate the best and second-best results, respectively. Higher is better; "–" denotes unreported results, and * indicates unavailable official code.

Method	Param	SQA-I	MMB	POPE	VizWiz	VQA-T	MME	VQAv2	GQA
Qwen-1.8B									
LLaVA	1.8B	61.5	60.2	86.1	32.2	45.7	1288	75.8	58.5
MoE-LLaVA	2.2B	62.8	59.7	87.0	32.6	48.0	1277	76.2	**61.5**
EvoMoE*	2.0B	63.3	61.6	87.1	–	**49.3**	**1315**	**76.9**	61.2
DYNMOE	2.2B	63.2	61.3	85.8	32.4	47.5	1295	76.4	60.9
γ-MoD	1.8B	58.8	59.9	86.5	32.0	46.0	1261	74.0	57.0
LLaVA-MoR	1.8B	**65.5**	**63.0**	**87.8**	**33.1**	48.9	1290	76.1	58.0
StableLM-2-1.6B									
LLaVA	1.6B	62.9	61.0	86.4	38.8	49.6	1269	75.8	59.2
MoE-LLaVA	2.0B	62.3	60.2	85.7	36.2	50.1	1316	76.7	60.4
EvoMoE*	1.8B	63.5	60.9	86.4	–	**51.5**	**1360**	76.9	61.2
DYNMOE	1.7B	63.4	63.2	85.7	40.6	48.9	1300	**77.4**	**61.4**
γ-MoD	1.6B	61.9	59.4	84.8	37.9	47.5	1301	74.1	56.0
LLaVA-MoR	1.6B	**64.5**	**64.6**	**86.7**	**41.0**	50.4	1312	76.1	58.7
Phi-2-2.7B									
LLaVA	2.7B	66.6	66.1	85.1	41.3	52.1	1344	75.6	58.5
MoE-LLaVA	3.6B	68.7	65.2	86.3	43.9	51.4	1423	77.6	61.4
EvoMoE*	3.0B	69.5	66.8	86.6	–	52.0	1450	77.8	**61.6**
DYNMOE	3.4B	68.0	66.6	86.0	**45.1**	51.8	1430	**77.9**	**61.6**
γ-MoD	2.7B	67.9	65.2	86.1	44.6	50.1	1420	75.5	56.9
LLaVA-MoR	2.7B	**70.0**	**69.7**	**86.9**	44.9	**52.4**	**1466**	77.5	58.1
Vicuna-7B									
LLaVA	6.7B	66.8	64.3	85.9	50.0	58.2	1511	78.5	**62.0**
LLaVA-MoR	6.7B	**67.9**	**65.8**	**87.0**	**50.2**	**58.9**	**1540**	**79.4**	61.6

Table 1 compares LLaVA-MoR with existing LVLMs across four language backbones. Crucially, compared with existing MoE-based and MoD-based conditional computation methods, LLaVA-MoR achieves stronger performance on SQA-I and MMBench through question-conditioned visual recursion without expanding the language backbone. The advantage of question-conditioned adaptive visual recursion is most striking in tasks requiring complex reasoning. On the Qwen-1.8B backbone, LLaVA-MoR improves over the dense LLaVA baseline by a substantial 6.5% relative gain on SQA-I and a 4.7% relative gain on MMBench, while achieving the highest POPE score in its group. More impressively, despite using fewer parameters than MoE-LLaVA, LLaVA-MoR still outperforms it on SQA-I and MMBench. These robust gains extend consistently across model scales. With the Phi-2 backbone, LLaVA-MoR decisively surpasses the parameter-heavy MoE-LLaVA on

both benchmarks. StableLM and Vicuna backbones exhibit similar upward trajectories, with StableLM achieving the highest SQA-I score in its group. Collectively, these results support our central premise: dynamically allocating recursive budgets to specific visual tokens, rather than merely scaling up static backbone parameters, is a highly effective pathway to strengthen multimodal reasoning without expanding language-backbone parameter counts.

4.3 Effectiveness of the Proposed Designs

Table 2 groups three complementary comparisons: the contribution of the two training stages, visual router design, and the depth-sampling schedule. The final row identifies the complete design of our method LLaVA-MoR.

Table 2: **Design ablations on Qwen-1.8B.** A: training stages; B: visual router design ($R = 3$), with routing restricted to visual tokens; C: depth-sampling schedules.

Configuration	SQA-I	VQA-T	POPE	MME	MMB
A. Training stages					
Dense LLaVA	61.5	45.7	86.1	1288	60.2
MoR only	63.6	45.1	85.4	1157	55.4
MoR + Stage I	63.8	46.3	86.9	1279	62.0
MoR + Stage II	64.3	46.4	87.0	1261	62.2
B. Visual Router Design					
Visual only	64.0	45.1	87.0	1281	62.2
Text + Visual	63.9	45.2	87.2	1268	62.8
Text–Visual Cross-Attention	65.1	47.0	87.4	1280	63.0
C. Depth-sampling schedule					
Direct sampling	64.3	45.2	87.1	1209	61.4
Linear depth sampling	64.7	46.2	86.9	1278	62.2
LLaVA-MoR	**65.5**	**48.9**	**87.8**	**1290**	**63.0**

Training stages. Group A isolates the contributions of our two-stage training procedure. While Dense LLaVA provides the non-recursive baseline, a naive MoR implementation lacking both proposed stages yields suboptimal results. Using only Stage I adapts the backbone across multiple depths but relies on vanilla routing, whereas using only Stage II introduces the question-conditioned router but suffers without prior multi-depth adaptation. The full LLaVA-MoR combines both stages, achieving the best performance across all five metrics. Notably, the combined approach improves SQA-I from 64.3 to 65.5 and TextVQA from 46.4 to 48.9 compared to Stage II alone. These results support the complementary benefits of backbone adaptation and visual routing.

Visual router design. Group B investigates router input representations. Although cross-attention between text and visual features surpasses simpler visual-only or naive concatenation baselines, it introduces additional computational overhead. LLaVA-MoR fuses visual states and pooled text queries through a lightweight element-wise product. This interaction achieves the highest scores across all evaluated benchmarks, including a further 0.6% relative gain on SQA-I over the cross-attention variant. These results confirm that explicitly conditioning routing decisions on the question effectively reduces computation allocated to task-irrelevant visual regions without incurring heavy cross-modal computational overhead.

Depth-sampling schedule. Group C isolates the multi-depth adaptation strategies employed in Stage I. Rather than immediately exposing the model to fixed depth sampling via Direct sampling or linearly interpolating between the initial and target depth distributions via Linear sampling, our gradual Cosine schedule (LLaVA-MoR) yields consistent gains across all benchmarks. Highlighted by an impressive 8.2% relative improvement on VQA-T, these empirical results confirm that a smooth cosine transition enables shared parameters to progressively adapt to multi-depth recursion, effectively mitigating capability degradation and preserving pretrained multimodal performance.

Table 3: **Inference efficiency on Qwen-1.8B.** Full visual recursion keeps all visual tokens active at every recursion depth. Reported gains indicate the relative improvement of LLaVA-MoR at $D = 3$. Accuracy is measured on SQA-I.

Method	Inference TFLOPs↓	Inference Latency↓	Inference Throughput↑	SQA-I Acc.↑
Full visual recursion ($D = 3$)	4.47	58.0 ms/img	17.6 samples/s	63.8
LLaVA-MoR ($D = 2$)	2.33	37.7 ms/img	27.4 samples/s	64.1
LLaVA-MoR ($D = 3$)	2.55	45.8 ms/img	22.4 samples/s	65.5
Gains	**-43.0%**	**-21.0%**	**+27.3%**	**+2.7%**

Inference efficiency. Compared with full visual recursion at depth 3, LLaVA-MoR reduces prefill TFLOPs by 43.0% and prefill latency by 21.0%, while improving SQA-I accuracy from 63.8% to 65.5% as shown in Table 3. Relative to dense LLaVA, it achieves higher accuracy at an increased prefill cost. Limiting recursion to $D = 2$ further reduces computation and latency while retaining competitive SQA-I performance. These results confirm that question conditioned visual budget allocation optimizes the tradeoff between accuracy and efficiency in multimodal reasoning.

4.4 Depth and Visual Ratio Sensitivity Analysis

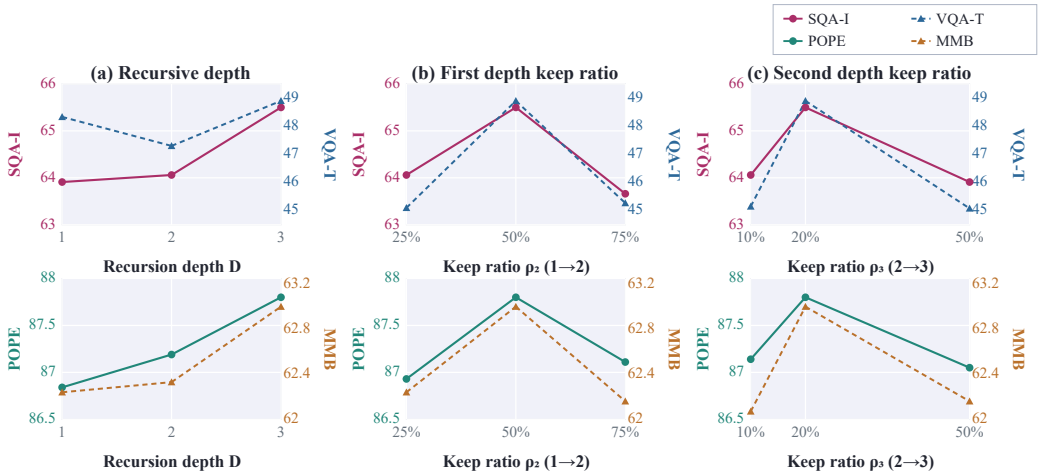


Figure 3: **Sensitivity analysis of depth and token keep ratios.** Upper row shows SQA-I and VQA-T; lower row shows POPE and MMB. **a.** Varying inference depth D. **b.** Varying keep ratio ρ_2 from $D = 1$ to 2 under fixed $\rho_3 = 20\%$. **c.** Varying keep ratio ρ_3 from $D = 2$ to 3 under fixed $\rho_2 = 50\%$. All evaluations use the Qwen-1.8B backbone.

Recursion depth. Figure 3(a) evaluates the same trained backbone across inference recursion depths $D \in \{1, 2, 3\}$ using default keep ratios. Advancing to $D = 3$ achieves peak performance across all metrics, yielding up to a 2.5% relative gain on SQA-I compared to early termination ($D = 1$). This demonstrates that deeper recursive computation systematically unlocks higher-level multimodal reasoning capabilities. Notably, $D = 1$ represents early termination of token processing rather than the dense baseline, confirming that multi-stage recursion progressively refines token representations for complex query answering.

Visual token keep ratios. At maximum depth ($D = 3$), Figures 3(b) and (c) evaluate the impact of visual token keep ratios at the second (ρ_2) and third (ρ_3) depths. The default configuration of $(\rho_2, \rho_3) = (50\%, 20\%)$ consistently yields the best performance trade-off. Deviating toward overly dense or sparse selection causes marked drops, notably a relative decline of over 7.5% on VQA-T. Crucially, retaining all visual tokens across all depths performs strictly worse than our selective routing scheme. This validates our premise in Challenge (2) that unselective recursion leads to the

over-processing of irrelevant visual details, whereas question-guided selection effectively focuses compute on relevant regions.

4.5 Case Study: Question-Conditioned Visual Routing

To illustrate how LLaVA-MoR mitigates visual overthinking, Figure 4 compares its dynamic compute allocation against the dense LLaVA baseline using different questions about the same image. For a shallow scene-recognition query ("Where is this?"), both models succeed. However, the second question requires deeper relational reasoning across distributed clues (the person, kite, and board). By unselectively processing the entire image, dense LLaVA hallucinates "para sailing," whereas LLaVA-MoR correctly deduces "kiteboarding."

The routing maps reveal the underlying mechanism behind this performance gap, directly validating our hypothesis regarding question-dependent visual selection. Rather than processing all regions uniformly, LLaVA-MoR explicitly conditions its compute allocation on the query. For complex questions, it prevents visual overthinking by allowing tokens from irrelevant background regions to exit early, while decisively routing patches containing the kite, person, and board to the deepest recursion depth (D3). This confirms that LLaVA-MoR effectively focuses its recursive refinement capacity on query-relevant regions, enabling accurate multi-hop reasoning. Appendix F.4.2 provides further examples, including cases where the same question is applied to different images.

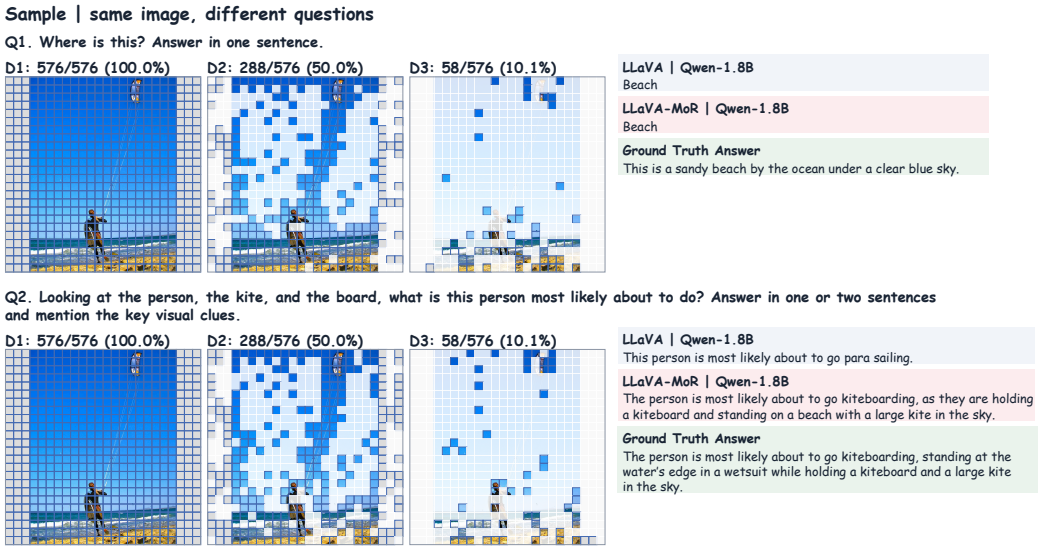


Figure 4: Visual routing for different questions about the same image. Unfaded patches indicate visual tokens that remain active at the indicated recursion depth. Responses from dense LLaVA and LLaVA-MoR, both using the Qwen-1.8B language backbone, are shown with the ground-truth answers.

5 Conclusion

In this work, we introduce LLaVA-MoR, a parameter-efficient framework that transforms a dense multimodal model into a dynamic recursive architecture via cosine-guided multi-depth adaptation and question-conditioned visual routing. By explicitly allocating recursive computation only to query-relevant image regions, LLaVA-MoR consistently improves performance across diverse multimodal reasoning benchmarks without increasing the backbone parameter count. These results confirm our central hypothesis: mitigating visual overthinking through query-guided, coarse-to-fine recursion is a highly effective, parameter-efficient approach to scale multimodal reasoning capabilities. While our current evaluation focuses on static image-level tasks, future work will extend this dynamic routing paradigm to more computationally demanding domains, such as high-resolution video understanding, where recursive visual pruning promises even greater efficiency gains.

AI USE STATEMENT

In this work, we used generative AI tools to assist in the writing of proofs and the implementation of methods. We have not used generative AI tools to generate synthetic datasets, develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, propose or refine hypotheses, design or provide feedback on research methodology or experiments, assist with translation, clean and reformat datasets, support qualitative and thematic data analysis, or interpret results. We have rigorously reviewed all AI-assisted work; specifically, we manually verified the logic of all AI-assisted proofs and tested the generated code for correctness. We take full responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikołaj Bińkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. Flamingo: A visual language model for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 23716–23736, 2022.

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6077–6086, 2018.

Sangmin Bae, Adam Fisch, Hrayr Harutyunyan, Ziwei Ji, Seungyeon Kim, and Tal Schuster. Relaxed recursive transformers: Effective parameter sharing with layer-wise loRA. In *International Conference on Learning Representations*, 2025a.

Sangmin Bae, Yujin Kim, Reza Bayat, Sungnyun Kim, Jiyoun Ha, Tal Schuster, Adam Fisch, Hrayr Harutyunyan, Ziwei Ji, Aaron Courville, and Se-Young Yun. Mixture-of-recursions: Learning dynamic recursive depths for adaptive token-level computation. In *Advances in Neural Information Processing Systems*, volume 38, pp. 96572–96617, 2025b.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

Marco Bellagente, Jonathan Tow, Dakota Mahan, Duy Phung, Maksym Zhuravinskyi, Reshinth Adithyan, James Baicoianu, Ben Brooks, Nathan Cooper, Ashish Datta, Meng Lee, Emad Mostaque, Michael Pieler, Nikhil Pinnaparju, Paulo Rocha, Harry Saini, Hannah Teufel, Niccolo Zanichelli, and Carlos Riquelme. Stable lm 2 1.6b technical report. *arXiv preprint arXiv:2402.17834*, 2024.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. Universal transformers. In *International Conference on Learning Representations*, 2019.

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

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, Rongrong Ji, Caifeng Shan, and Ran He. Mme: A comprehensive evaluation benchmark for multimodal large language models. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Angeliki Giannou, Shashank Rajput, Jy-Yong Sohn, Kangwook Lee, Jason D. Lee, and Dimitris Papailiopoulos. Looped transformers as programmable computers. In *International Conference on Machine Learning*, volume 202, pp. 11398–11442, 2023.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6325–6334, 2017.

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

Danna Gurari, Qing Li, Abigale J. Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P. Bigham. VizWiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3608–3617, 2018.

Yunhong He, Zhengqing Yuan, Weixiang Sun, Yiyang Li, Yixin Liu, Yanfang Ye, and Lichao Sun. Vision-mor: Scaling vision transformer via patch-level mixture-of-recursions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 4699–4707, 2026.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2023.

Drew A. Hudson and Christopher D. Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6693–6702, 2019.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991.

Mojan Javaheripi, Sébastien Bubeck, Marah Abdin, Jyoti Aneja, Sebastien Bubeck, Caio César Teodoro Mendes, Weizhu Chen, Allie Del Giorno, Ronen Eldan, Sivakanth Gopi, et al. Phi-2: The surprising power of small language models. *Microsoft Research Blog*, 1(3):3, 2023.

Linglin Jing, Yuting Gao, Zhigang Wang, Wang Lan, Yiwen Tang, Weiyun Wang, Wenhai Wang, and Qingpei Guo. Evomoe: Expert evolution in mixture of experts for multimodal large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 22463–22471, 2026.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*, 2020.

Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Fanyi Pu, Jingkang Yang, Chunyuan Li, and Ziwei Liu. Mimic-it: Multi-modal in-context instruction tuning. *arXiv preprint arXiv:2306.05425*, 2023a.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 292–305, 2023b.

Bin Lin, Zhenyu Tang, Yang Ye, Jinfa Huang, Junwu Zhang, Yatian Pang, Peng Jin, Munan Ning, Jiebo Luo, and Li Yuan. Moe-llava: Mixture of experts for large vision-language models. *IEEE Transactions on Multimedia*, 28:4408–4419, 2026.

Fuxiao Liu, Kevin Lin, Linjie Li, Jianfeng Wang, Yaser Yacoob, and Lijuan Wang. Mitigating hallucination in large multi-modal models via robust instruction tuning. In *International Conference on Learning Representations*, 2024a.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 34892–34916, 2023.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26286–26296, 2024b.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. Mmbench: Is your multi-modal model an all-around player? In *Computer Vision – ECCV 2024*, pp. 216–233, 2024c.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *Advances in Neural Information Processing Systems*, volume 35, pp. 2507–2521, 2022.

Yaxin Luo, Gen Luo, Jiayi Ji, Yiyi Zhou, Xiaoshuai Sun, Zhiqiang Shen, and Rongrong Ji. γ-MoD: Exploring mixture-of-depth adaptation for multimodal large language models. In *International Conference on Learning Representations*, 2025.

Yongcan Luo, Zhihao Yu, Jiahao Zheng, Zhengjie Yang, Lei Sun, and Dapeng Wu. Causality-aware graph mixture of experts for accurate multi-cloud workload prediction. *IEEE Transactions on Cloud Computing*, 14(3):1569–1584, 2026.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 8748–8763, 2021.

Nikunj Saunshi, Nishanth Dikkala, Zhiyuan Li, Sanjiv Kumar, and Sashank J. Reddi. Reasoning with latent thoughts: On the power of looped transformers. In *International Conference on Learning Representations*, 2025.

Avi Schwarzschild, Eitan Borgnia, Arjun Gupta, Furong Huang, Uzi Vishkin, Micah Goldblum, and Tom Goldstein. Can you learn an algorithm? generalizing from easy to hard problems with recurrent networks. In *Advances in Neural Information Processing Systems*, volume 34, pp. 6695–6706, 2021.

Fangxun Shu, Yue Liao, Lei Zhang, Le Zhuo, Chenning Xu, Guanghao Zhang, Haonan Shi, Long Chan, TaoZhong, Zhelun Yu, Wanggui He, Siming Fu, Haoyuan Li, Si Liu, Hongsheng Li, and Hao Jiang. LLaVA-mod: Making LLaVA tiny via moe-knowledge distillation. In *International Conference on Learning Representations*, 2025.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8309–8318, 2019.

Junke Wang, Lingchen Meng, Zejia Weng, Bo He, Zuxuan Wu, and Yu-Gang Jiang. To see is to believe: Prompting gpt-4v for better visual instruction tuning. *arXiv preprint arXiv:2311.07574*, 2023.

Yang Xiao, Chunpu Xu, Ruifeng Yuan, Jiashuo Wang, Wenjie Li, and Pengfei Liu. Scale: Selective resource allocation for overcoming performance bottlenecks in mathematical test-time scaling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 34034–34042, 2026.

Liu Yang, Kangwook Lee, Robert D Nowak, and Dimitris Papailiopoulos. Looped transformers are better at learning learning algorithms. In *International Conference on Learning Representations*, 2024.

Shaolei Zhang, Qingkai Fang, Zhe Yang, and Yang Feng. LLaVA-mini: Efficient image and video large multimodal models with one vision token. In *International Conference on Learning Representations*, 2025.

Bo Zhao, Boya Wu, Muyang He, and Tiejun Huang. Svit: Scaling up visual instruction tuning. *arXiv preprint arXiv:2307.04087*, 2023.

Jiahao Zheng, Yu Tang, Yongcan Luo, Ning Chen, Dan Zeng, and Dapeng Wu. Cop: Cross-view attention prompt for zero-shot sketch-based image retrieval. *IEEE Transactions on Multimedia*, 28:3880–3892, 2026a.

Jiahao Zheng, Xiaoping Wang, Yongcan Luo, Yun Wang, Yu Tang, and Dapeng Wu. General diffusion transformer for arbitrary medical image translation. *IEEE Transactions on Fuzzy Systems*, 34(5):1430–1444, 2026b.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, pp. 46595–46623, 2023.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M. Dai, Zhifeng Chen, Quoc V. Le, and James Laudon. Mixture-of-experts with expert choice routing. In *Advances in Neural Information Processing Systems*, volume 35, pp. 7103–7114, 2022.

Yichen Zhu, Minjie Zhu, Ning Liu, Zhiyuan Xu, and Yaxin Peng. Llava-phi: Efficient multi-modal assistant with small language model. In *Proceedings of the 1st International Workshop on Efficient Multimedia Computing under Limited*, pp. 18–22, 2024.

CONTENTS OF APPENDIX

A Related Work . . . 15
A.1 Efficient Multimodal Large Language Models . . . 15
A.2 Recursive Transformers and Latent Reasoning . . . 15
B Architecture of Mixture-of-Recursion . . . 16
B.1 Design of Mixture-of-Recursion . . . 16
B.2 Comparison with Other Computation Methods . . . 16
C Experimental Settings . . . 17
C.1 Model Configuration . . . 17
C.2 Training Configuration . . . 17
C.3 Evaluation Benchmark . . . 18
C.4 Experiment Reproduction . . . 18
D Baseline Methods . . . 19
E Detailed Algorithm Framework . . . 19
F Additional Experiments . . . 20
F.1 Ablation Study on Router Design . . . 20
F.2 Sensitivity Analysis on Visual Budget Keep Ratio . . . 20
F.3 Inference Analysis . . . 21
F.4 Visualization . . . 23
F.4.1 Heatmap . . . 23
F.4.2 More Case Studies . . . 24
Same Image, Different Questions . . . 24
Same Question, Different Images . . . 26

A Related Work

A.1 Efficient Multimodal Large Language Models

Multimodal learning aims to integrate complementary information from different modalities, such as vision and language, to enhance understanding and reasoning (Zheng et al., 2026a). Within this paradigm, multimodal large language models combine visual encoders with pretrained language models, as exemplified by Flamingo and LLaVA (Alayrac et al., 2022; Liu et al., 2023). Their large backbones and lengthy visual inputs motivate efforts to reduce model size and computation. LLaVA-Phi adopts a compact language backbone (Zhu et al., 2024), while LLaVA-MoD combines a sparse student architecture with progressive knowledge distillation (Shu et al., 2025). LLaVA-Mini reduces visual context length through vision-language fusion before the language backbone (Zhang et al., 2025). These approaches improve efficiency through compact architectures, knowledge transfer, or visual-token compression.

Another line of work explores conditional computation (Luo et al., 2026; Zheng et al., 2026b). Sparse Mixture-of-Experts (MoE) activates a subset of feed-forward experts for each token (Jacobs et al., 1991; Fedus et al., 2022). MoE-LLaVA introduces dense-to-sparse adaptation for MLLMs (Lin et al., 2026), EvoMoE promotes expert differentiation and modality-aware routing (Jing et al., 2026), and DYNMOE dynamically determines expert counts and per-token activation (Guo et al., 2025). Mixture-of-Depths (MoD) instead allows selected tokens to bypass Transformer blocks. γ-MoD adapts this mechanism to MLLMs through attention-rank-guided layer selection and shared vision-language routing with masked routing learning (Luo et al., 2025). While MoE selects among distinct experts and MoD selectively executes existing layers, LLaVA-MoR repeatedly applies shared Transformer blocks without adding backbone parameters. Its question-conditioned router allocates deeper recursion to relevant image regions to support multimodal reasoning.

A.2 Recursive Transformers and Latent Reasoning

Parameter sharing across depth increases computation without proportionally increasing model size. Early work explored this principle through recurrent refinement in Universal Transformers (Dehghani et al., 2019) and cross-layer parameter sharing in ALBERT (Lan et al., 2020). Subsequent studies connect recurrent computation to algorithmic generalization and reasoning. Schwarzschild et al. (2021) show that recurrent networks can generalize from simpler to harder problem instances by taking additional recurrent steps. Looped Transformers have also been studied as programmable computers (Giannou et al., 2023) and as models for learning iterative algorithms in context (Yang et al., 2024). Beyond algorithmic tasks, Saunshi et al. (2025) provide theoretical and empirical evidence that looping can support reasoning with fewer distinct parameters, including improvements on downstream language-model reasoning tasks.

Recent work further investigates pretrained-model adaptation and adaptive recursive computation. Relaxed Recursive Transformers convert dense language models into recursive architectures using depth-specific low-rank adapters (Bae et al., 2025a). Mixture-of-Recursions (MoR) learns token-specific recursion depths through lightweight routers (Bae et al., 2025b), while Vision-MoR extends adaptive recursion to visual patches (He et al., 2026). Building on these studies, LLaVA-MoR focuses on adapting a dense instruction-tuned vision-language model to recursive computation. It combines cosine-guided multi-depth adaptation with question-conditioned visual routing, preparing the backbone for different recursion depths and directing deeper processing toward question-relevant image regions.

B ARCHITECTURE OF MIXTURE-OF-RECURSION

B.1 DESIGN OF MIXTURE-OF-RECURSION

Figure 5(a) illustrates the Middle-Cycle design of Mixture-of-Recursion (MoR). The first decoder layer forms F_{in}; the intermediate layers form the shared block $F(\cdot;\theta)$; and the final decoder layer, final normalization, and language-model head form F_{out}. The input and output blocks are applied once, while the shared block is repeatedly applied to refine token representations. Intermediate layers retain distinct parameters, with the same parameter set θ reused across recursions (curved arrow). The maximum recursion depth R controls the number of possible shared-block applications, extending the computation path without introducing additional backbone parameters.

The router and bypass path in Figure 5(a) enable token-level adaptive computation. All valid tokens participate in the first recursion. At each subsequent transition, the router selects tokens for further processing through the shared block. Selected tokens (blue) continue updating their hidden states, whereas unselected tokens (gray) exit and preserve their latest representations. Repeating this selection allows different tokens to undergo different numbers of recursive updates within the same sequence. After the final recursion $D \leq R$, the full hidden-state sequence $\mathbf{H}^{(D)}$, including the preserved states of exited tokens, is processed by F_{out} to produce $\mathbf{O}_{\text{out}}$. Early exit therefore skips further shared-block refinement while retaining final output-block processing.

B.2 COMPARISON WITH OTHER COMPUTATION METHODS

Figure 5 contrasts expert selection (MoE), layer skipping (MoD), and shared-stack recursion (MoR). Unlike MoE, which routes tokens across different experts, and MoD, which selectively skips layers, MoR introduces a multi-depth recursive architecture. Specifically, MoR employs a router at each recursion depth to dynamically determine whether a token requires further iterative processing. By repeatedly routing tokens through the parameter-shared block, this mechanism enables progressive hidden-state refinement. This effectively extends the reasoning path for challenging regions, achieving deeper multimodal reasoning capabilities.

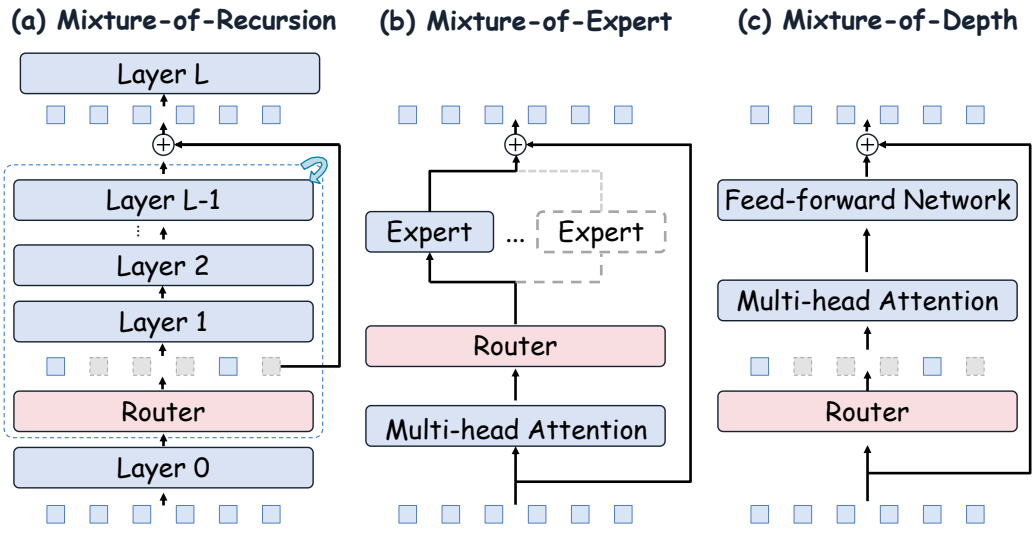


Figure 5: **Comparison of MoR, MoE, and MoD.** (a) MoR dynamically allocates token computation through parameter-shared recursion. (b) MoE expands parameter count via expert selection, and (c) MoD executes layer skipping across tokens. MoR achieves flexible depth adaptation without adding backbone parameters. Crucially, while MoE and MoD act primarily as computational methods that often lead to visual overthinking of simple background patches, MoR leverages multi-depth recursion to focus adaptive reasoning on selected tokens, thereby achieving superior multimodal reasoning capabilities. Blue and gray tokens indicate active and bypassed tokens.

C Experimental Settings

This section provides the model, training, and evaluation settings accompanying Section 4.

C.1 Model Configuration

LLaVA-MoR follows the LLaVA-1.5 architecture (Liu et al., 2024b), combining a CLIP ViT-L/14-336 vision encoder (Radford et al., 2021), a two-layer MLP multimodal projector with GELU activation (Hendrycks & Gimpel, 2023), and a pretrained language backbone. The projector maps visual features into the embedding space of the corresponding language model. We use Qwen-1.8B (Bai et al., 2023), StableLM-2-1.6B (Bellagente et al., 2024), Phi-2-2.7B (Javaheripi et al., 2023), and Vicuna-7B (Zheng et al., 2023) to evaluate performance across language-model families.

C.2 Training Configuration

Training data. Following the data grouping of MoE-LLaVA (Lin et al., 2026), we use LLaVA-PT, Hybrid-FT, and LLaVA-FT to construct the dense initialization, as summarized in Table 4. LLaVA-PT contains 558K image–text samples from LLaVA-1.5 (Liu et al., 2024b) for multimodal alignment. Hybrid-FT provides 964K instruction samples by combining SViT (Zhao et al., 2023), LVIS-Instruct4V (Wang et al., 2023), LRV-Instruction (Liu et al., 2024a), and MIMIC-IT (Li et al., 2023a). We use the SViT core split and the MIMIC-IT LA split, following MoE-LLaVA. LLaVA-FT is the LLaVA-1.5-mix-665K instruction mixture (Liu et al., 2024b). After dense initialization, both Stage I backbone adaptation and Stage II visual-router training reuse this LLaVA-1.5-mix-665K data. All

Table 4: Composition and usage of the training data groups, following MoE-LLaVA (Lin et al., 2026). Stage I and Stage II refer to our MoR training stages after dense initialization.

Data group	Usage	Source	#Samples
LLaVA-PT	Dense pretraining	LLaVA-1.5-558K	558K
Hybrid-FT	Dense instruction tuning	SViT-157K, LVIS-Instruct4V-220K LRV-331K, MIMIC-IT-256K	964K
LLaVA-FT	Dense fine-tuning; MoR Stage I / II	LLaVA-1.5-mix-665K	665K

LLaVA-MoR variants use the two-stage training procedure described in the Method. Stage I updates the language backbone and multimodal projector through Cosine-Guided Multi-Depth Adaptation, while Stage II freezes these components and trains only the Question-Conditioned Visual Router. The vision encoder remains frozen throughout both stages. Table 5 summarizes the hyperparameters for Stage I and Stage II, both of which use LLaVA-1.5-mix-665K.

Table 5: Training hyperparameters for the two training stages.

Config	Stage I	Stage II
Training data	LLaVA-1.5-mix-665K	
Vision encoder	CLIP ViT-L/14-336	
Image resolution	336×336	
Projector	Two-layer MLP with GELU	
Maximum depth R	3	
$\mathbf{p}_{\text{start}}$	$(1, 0, 0)$	–
$\mathbf{p}_{\text{target}}$	$(0.7, 0.2, 0.1)$	–
T_{adapt} (steps)	1000	–
Auxiliary coefficient η	–	0.01
Learning rate	5×10^{-6}	10^{-5}
Epochs	1	0.5
Batch size per GPU	2	
Gradient accumulation	8	
Effective batch size	64	
GPUs	$4\times$ NVIDIA A100-80GB	
Optimizer	AdamW	
DeepSpeed	ZeRO-2	
Learning-rate schedule	Cosine	
Weight decay	0.0	
Text max length	2048	
Precision	BF16	

C.3 Evaluation Benchmarks

We evaluate LLaVA-MoR on eight benchmarks also used by LLaVA-1.5 and MoE-LLaVA (Liu et al., 2024b; Lin et al., 2026). VQAv2 (Goyal et al., 2017) and GQA (Hudson & Manning, 2019) assess visual question answering and compositional reasoning; VizWiz (Gurari et al., 2018) evaluates questions paired with images captured by blind users; ScienceQA-IMG (Lu et al., 2022) measures science question answering on image-containing examples; and TextVQA (Singh et al., 2019) evaluates reasoning over text in images. We additionally use POPE (Li et al., 2023b) to assess object hallucination, MME (Fu et al., 2025) to evaluate visual perception, and MMBench (Liu et al., 2024c) to assess multimodal understanding.

We follow the official benchmark metrics, reporting VQA accuracy for VQAv2, VizWiz, and TextVQA; answer accuracy for GQA, ScienceQA-IMG, and MMBench; F1 for POPE; and the perception score for MME. For models evaluated in our experiments, benchmark splits, OCR inputs, answer extraction, and decoding settings are kept consistent within each comparison. Scores are reported separately to preserve the interpretation of each benchmark's metric.

C.4 Experiment Reproduction

We train and evaluate LLaVA, MoE-LLaVA, DYNMOE, and γ-MoD using their publicly available implementations. The evaluation scores of EvoMoE in Table 1 are directly taken from its original paper (Jing et al., 2026), as its official code is unavailable. For the baselines we reproduce, we use the same vision encoder, image resolution, and multimodal projector as LLaVA-MoR, with consistent evaluation splits, TextVQA OCR inputs, POPE subsets, MMBench versions, and decoding settings. The dense LLaVA baseline also receives additional fine-tuning on LLaVA-1.5-mix-665K. Missing results are marked as "–".

D BASELINE METHODS

We compare dense, MoE-based, and MoD-based architectures to examine how different conditional computation methods affect multimodal performance.

LLaVA. LLaVA (Liu et al., 2024b) connects a visual encoder to a dense language model through a multimodal projector. It provides the reference for evaluating the effect of recursive conversion.

MoE-LLaVA. MoE-LLaVA (Lin et al., 2026) converts a dense vision-language model into a sparse expert architecture through MoE tuning. Tokens activate a subset of feed-forward experts, providing a comparison with computation allocated across model width.

EvoMoE. EvoMoE (Jing et al., 2026) introduces expert evolution and a dynamic token-aware router. Expert evolution constructs differentiated experts, while the router uses modality-specific transformations to select experts for visual and textual tokens.

DYNMOE. DYNMOE (Guo et al., 2025) adapts the expert pool and the number of experts activated by each token. Its vision-language experiments provide a comparison with adaptive expert allocation.

γ-MoD. γ-MoD (Luo et al., 2025) uses attention-map rank to guide the placement of sparse depth layers, together with shared vision-language routing and masked routing learning. It provides a comparison with selective token processing across distinct backbone layers.

E DETAILED ALGORITHM FRAMEWORK

Algorithm 1 follows Eqs. 1–9. Forward operations and routing sets are defined per sample; losses in Eqs. 8–9 are averaged over each minibatch.

Algorithm 1 Training LLaVA-MoR

Require: Dense LLaVA; samples (X_v, X_q, X_a); R; $\mathbf{p}_{\text{start}}, \mathbf{p}_{\text{target}}, T_{\text{adapt}}$; budgets $K_2, \ldots, K_R$; η.
Ensure: Adapted LLaVA-MoR with learned visual-router parameters ϕ.

Initialize the recursive backbone (Appendix B.1) and $\phi = \{(\mathbf{w}_d, b_d)\}_{d=1}^{R-1}$.
Stage I: Cosine-Guided Multi-Depth Adaptation
Freeze g and ϕ; train f_{proj} and the language backbone.
for each training minibatch at step t **do**
  Compute $\mathbf{p}_t$ using the cosine schedule in Eq. 4.
  for each sample, with independent $D \sim \text{Categorical}(\mathbf{p}_t)$ **do**
    Construct H_{mm} using Eqs. 1–2; set $\mathbf{H}^{(0)} = F_{\text{in}}(H_{\text{mm}})$.
    Apply $F(\cdot;\theta)$ to all valid tokens D times to obtain $\mathbf{H}^{(D)}$ (Eq. 3).
    $\mathbf{O}_{\text{out}} \leftarrow F_{\text{out}}(\mathbf{H}^{(D)})$; compute $\mathcal{L}_{\text{answer}}$ using Eq. 8.
  Update f_{proj} and the language backbone using the minibatch mean of $\mathcal{L}_{\text{answer}}$.
Stage II: Question-Conditioned Visual Router
Freeze g, f_{proj}, and the adapted language backbone; train only ϕ.
for each training minibatch **do**
  for each sample in the minibatch **do**
    Construct H_{mm} using Eqs. 1–2; set $\mathbf{H}^{(0)} = F_{\text{in}}(H_{\text{mm}})$.
    $\mathbf{H}^{(1)} \leftarrow F(\mathbf{H}^{(0)};\theta)$ for all valid tokens; $\mathcal{A}^{(1)} \leftarrow$ all visual-token indices.
    for $d = 1, \ldots, R-1$ **do**
      Pool $\bar{\mathbf{h}}_q^{(d)}$ over $\mathcal{Q}$; compute $r_i^{(d)}$ for $i \in \mathcal{A}^{(d)}$ using Eq. 5.
      Select $\mathcal{A}^{(d+1)} \subseteq \mathcal{A}^{(d)}$ by Top-K with budget K_{d+1} (Eq. 6).
      Evaluate $\Delta F(\mathbf{H}^{(d)};\theta)$ on selected visual and all valid text tokens.
      $\mathbf{h}_{v,i}^{(d+1)} \leftarrow \mathbf{h}_{v,i}^{(d)} + r_i^{(d)} \Delta F(\mathbf{H}^{(d)};\theta)_i$ for $i \in \mathcal{A}^{(d+1)}$ (Eq. 7).
      Apply full residuals to all valid text tokens; retain exited visual states.
    $\mathbf{O}_{\text{out}} \leftarrow F_{\text{out}}(\mathbf{H}^{(R)})$; compute $\mathcal{L}_{\text{answer}}$ using Eq. 8.
    Compute $m_i^{(d)}$, N_{route}, and $\mathcal{L}_{\text{aux}}$ as defined for Eq. 9.
  Update only ϕ using the minibatch mean of $\mathcal{L}_{\text{route}} = \mathcal{L}_{\text{answer}} + \eta \mathcal{L}_{\text{aux}}$.

F Additional Experiments

This section provides supplementary details for the main paper, extending the ablation study on router configurations and presenting sensitivity analyses on visual budget keep ratios, inference analysis, and visualization experiments.

F.1 Ablation Study on Router Design

This section extends the ablation study of router designs presented in Group B of Table 2. Table 6 evaluates six distinct configurations on the Qwen-1.8B backbone with a maximum recursion depth of 3. The first category includes two global baselines operating without modality-specific routing. The Static full recursion acts as a dense baseline forcing all visual and text tokens to undergo all three recursions without router score weighting. The Vanilla router applies a standard routing mechanism indiscriminately across both visual and text tokens.

The second category comprises four configurations routing exclusively on visual tokens while keeping all text tokens active. These variants progressively enhance the routing computation through Visual only scoring, Text + Visual interaction, Text and Visual Cross Attention, and our final LLaVA-MoR setup featuring question-conditioned elementwise interaction. As shown in Table 6, LLaVA-MoR achieves the highest average score, yielding a 2.4 percent relative improvement over the Static full recursion baseline and a 1.0 percent relative improvement over the Text and Visual Cross Attention variant. The relative underperformance of the Static full recursion demonstrates that blindly executing every token through maximum depth does not yield the strongest capability, validating our premise that dynamic multi-depth routing effectively mitigates redundant visual processing.

Table 6: Router design ablation on Qwen-1.8B with maximum recursion depth of 3. Static full recursion uses full recursion for all tokens. Vanilla router reproduces the MoR only reference, while visual router results reproduce Group B. POPE reports F1 and MME reports perception scores. Avg is the mean of the five reported scores after dividing MME by 20 to match the 0 to 100 scale. Best scores are bolded.

Router design	SQA-I	VQA-T	POPE	MME	MMB	Avg
Static full recursion	63.8	45.2	86.9	1278.8	62.0	64.4
Vanilla router	63.6	45.1	85.4	1157.0	55.4	61.5
Visual only	64.0	45.1	87.0	1281.0	62.2	64.5
Text + Visual	63.9	45.2	87.2	1268.0	62.8	64.5
Text and Visual Cross Attention	65.1	47.0	87.4	1280.0	**63.0**	65.3
LLaVA-MoR	**65.5**	**48.9**	**87.8**	**1290.0**	**63.0**	**65.9**

F.2 Sensitivity Analysis on Visual Budget Keep Ratio

This section supplements Section 4.4, *Depth and visual ratio sensitivity analysis*, with the complete results for visual token keep ratios. We perform a sensitivity analysis on the per transition visual token keep ratios ρ_2 and ρ_3 using the same trained Qwen-1.8B backbone at maximum recursion depth 3. The first recursion processes all visual tokens. Starting from the default keep ratios of 50 percent and 20 percent, we vary the visual-token keep ratio at one depth at a time and include a baseline condition keeping all visual tokens. Each ratio is relative to the preceding active visual set.

Table 7 demonstrates that the default configuration achieves the highest scores across all metrics. Compared to the brute-force approach of keeping all visual tokens, the default setup yields a 7.9 percent relative improvement on TextVQA and a 2.7 percent relative improvement on SQA-I. Increasing either keep ratio diminishes overall performance, indicating that excessive visual token retention introduces noise and over-processing of irrelevant visual features during inference. These results firmly support the default setting for this checkpoint and highlight the inference-time sensitivity of recursive budget allocation.

Table 7: Sensitivity analysis of visual budget keep ratios at maximum recursion depth 3. Ratios are relative to the preceding active visual set. All token routing retains score-weighted updates. POPE reports F1 and MME reports perception scores. Best scores are bolded.

Configuration	Keep ratio ρ_2	Keep ratio ρ_3	SQA-I	VQA-T	POPE	MME	MMB
Lower ρ_2	25%	20%	64.1	45.1	86.9	1273.0	62.2
Higher ρ_2	75%	20%	63.7	45.3	87.1	1277.5	62.2
Lower ρ_3	50%	10%	64.1	45.2	87.1	1279.1	62.1
Higher ρ_3	50%	50%	63.9	45.1	87.1	1267.0	62.2
All visual tokens kept	100%	100%	63.8	45.3	86.9	1279.7	62.2
LLaVA-MoR (Default)	50%	20%	**65.5**	**48.9**	**87.8**	**1290.0**	**63.0**

F.3 Inference Analysis

Table 8: Inference efficiency of LLaVA-MoR on Qwen-1.8B. Extended configurations for Table 3, measured on one NVIDIA A100-80GB with bf16, batch size 1. TFLOPs and latency measure per-image prefill; throughput uses a separate continuous run of 100 images. SQA-I accuracy follows Tables 2 and 3. Best scores are bolded.

Method	Visual tokens at each depth	Inference TFLOPs↓	Inference Latency↓ (ms/img)	Inference Throughput↑ (samples/s)	SQA-I Acc.↑
Dense LLaVA	576	**1.58**	**26.4**	**39.6**	61.5
Full visual recursion ($D = 1$)	576	**1.58**	27.1	38.4	63.9
Full visual recursion ($D = 2$)	$576 \to 576$	3.02	42.8	24.0	64.1
Full visual recursion ($D = 3$)	$576 \to 576 \to 576$	4.47	58.0	17.6	63.8
LLaVA-MoR ($D = 2$)	$576 \to 288$	2.33	37.7	27.4	64.1
LLaVA-MoR ($D = 3$)	$576 \to 288 \to 58$	2.55	45.8	22.4	**65.5**

Computational efficiency. Table 8 extends Table 3 by evaluating all six configurations under a unified inference protocol while retaining the benchmark accuracy reported in the main text. In the baseline full recursion setup, processing all 576 visual tokens at every depth causes the computational cost to scale sharply from 1.58 TFLOPs at depth 1 to 4.47 TFLOPs at depth 3. In contrast, LLaVA-MoR dynamically prunes visual tokens across depths down to 58 tokens at depth 3, effectively reducing inference computation by 43.0 percent and latency by 21.0 percent while increasing throughput by 27.3 percent relative to full recursion at the same depth. At depth 2, LLaVA-MoR similarly achieves an 11.9 percent reduction in latency and a 14.2 percent increase in throughput compared to its full recursion counterpart. Throughput and latency are measured in separate continuous benchmark runs, which explains why throughput is not the exact mathematical reciprocal of per-image latency.

Memory and parameter efficiency. Complementing the inference TFLOPs, latency, and throughput in Table 3, Figure 6 compares peak memory and language-backbone parameters alongside SQA-I accuracy and POPE F1, extending the accuracy–efficiency analysis to the comparison across models.

Figure 6(a) compares peak memory and SQA-I accuracy on the Qwen-1.8B backbone. LLaVA-MoR achieves 65.5 accuracy with 5.6 GB peak memory, compared with 61.5 accuracy and 5.5 GB for dense LLaVA. This corresponds to a 6.5% relative gain on SQA-I with only 0.1 GB additional memory. Compared with MoE-LLaVA, LLaVA-MoR improves SQA-I accuracy from 62.8 to 65.5, a 4.3% relative gain, while reducing peak memory from 8.0 to 5.6 GB, a 30.0% reduction.

Figure 6(b) compares POPE F1 against language-backbone parameter counts. LLaVA-MoR achieves 87.8 F1 with 1.84B parameters, compared with 86.1 F1 and 1.83B parameters for dense LLaVA with Qwen-1.8B. It also achieves higher POPE F1 than the displayed MoE variants while using fewer parameters. Consistent with Table 1, these results support the parameter efficiency of LLaVA-MoR: question-conditioned visual recursion improves multimodal performance with a language backbone comparable in size to dense LLaVA.

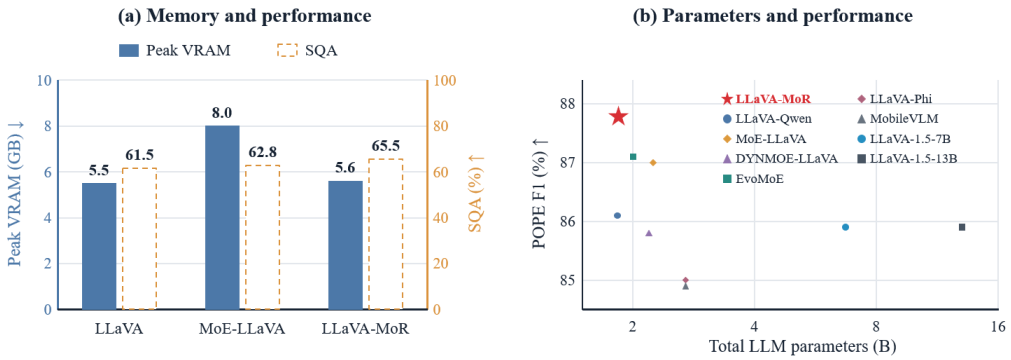


Figure 6: **Memory and parameter efficiency of LLaVA-MoR.** (a) Peak memory (GB) and SQA-I accuracy of LLaVA, MoE-LLaVA, and LLaVA-MoR with the Qwen-1.8B backbone. (b) POPE F1 versus language-backbone parameters in billions, with LLaVA-MoR marked by a red star; the horizontal axis is logarithmic. Panel (a) reports the comparison across models, separately from the recursion-depth configurations in Table 8.

F.4 VISUALIZATION

F.4.1 HEATMAP

For the displayed question, the router retains 576, 288, and 58 visual tokens at depths 1–3. Gray cells indicate early exit; colors show relative scores among active tokens within each depth.

In one or two sentences, describe how the cricket bat is positioned relative to the batter and any other important people in the scene.

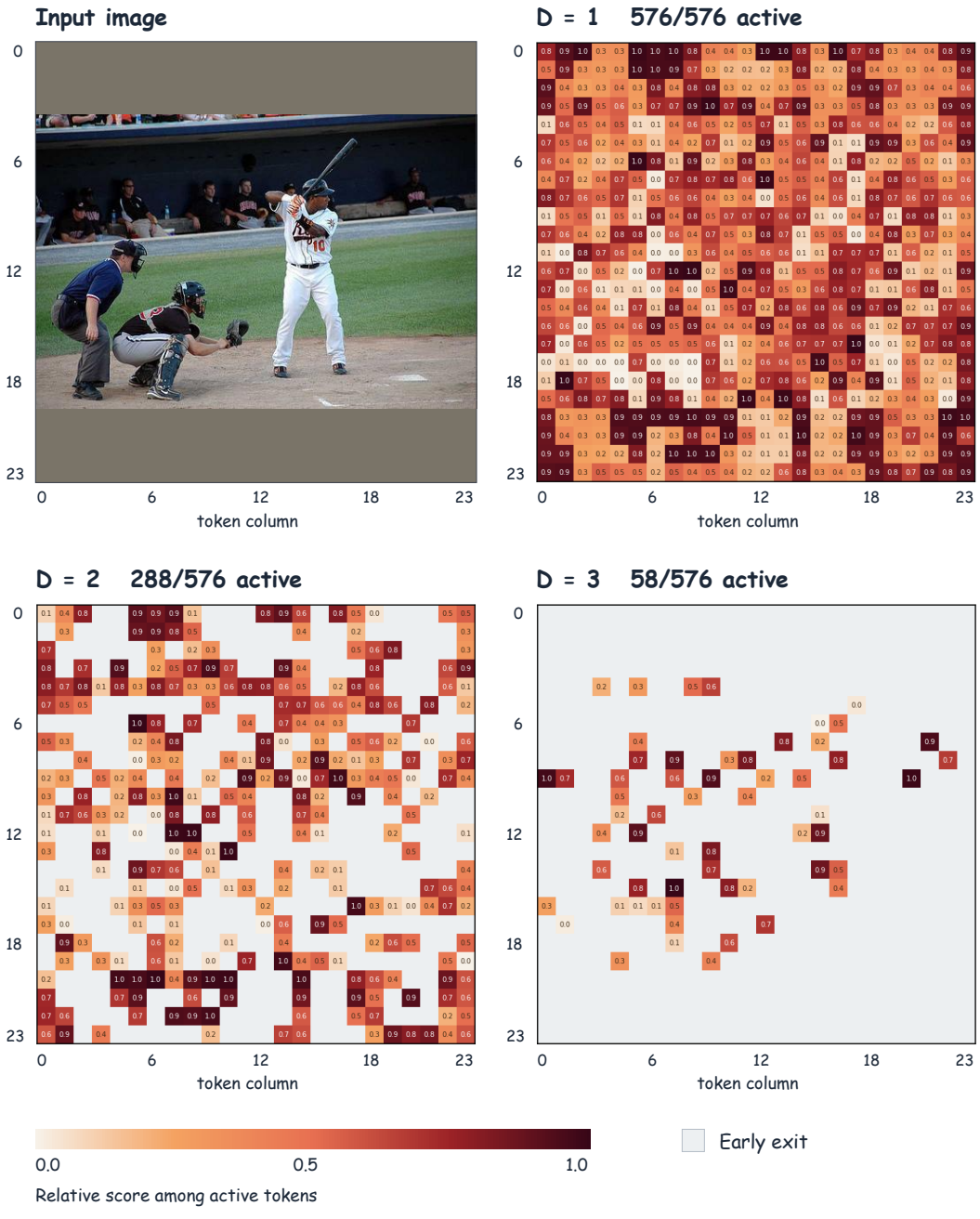


Figure 7: Question-conditioned visual-token routing on a padded 24×24 grid.

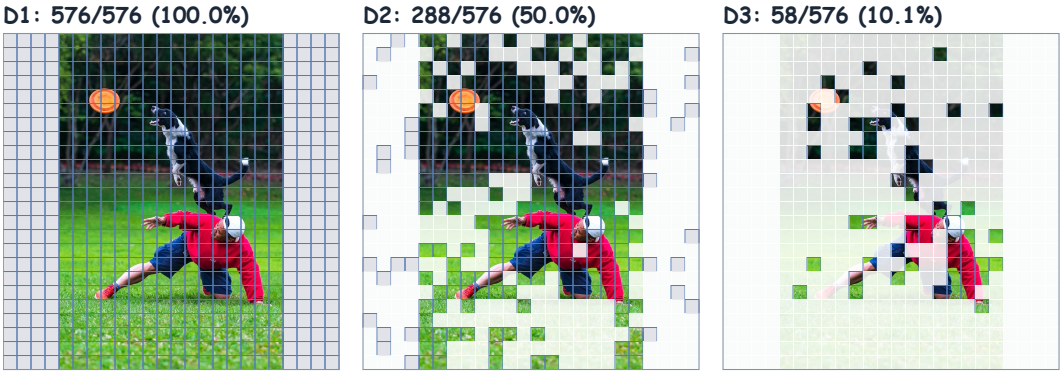
F.4.2 More Case Studies

We extend Section 4.5 with comparisons across questions and images. Responses are shown alongside ground-truth answers; the same-image cases also show visual routing at successive depths.

Same Image, Different Questions. Following Figure 4, the park case compares scene recognition and activity reasoning under the same visual budget.

Case 1 | Same image, different questions

Q1. Where is this? Answer in one sentence.



LLaVA | Qwen-1.8B	LLaVA-MoR | Qwen-1.8B	Ground Truth Answer
Park	Park	This is a grassy park.

Q2. Looking at the person, the dog, and the flying disc, what are they most likely doing? Answer in one or two sentences and mention the key visual clues.



LLaVA | Qwen-1.8B	LLaVA-MoR | Qwen-1.8B	Ground Truth Answer
The man and the dog are playing with a flying disc, and the dog is jumping on the man's head.	The man and the dog are most likely playing frisbee in the park.	The person and the dog are playing with a flying disc; the dog is jumping up to catch the disc.

Unfaded patches: active tokens. Gray margins: square padding.

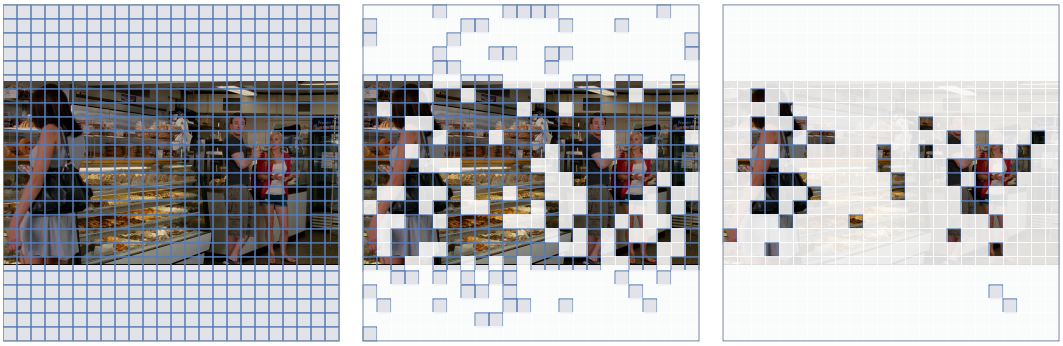
Figure 8: Visual routing for different questions about the same park image. Unfaded patches denote active visual tokens on the padded 24×24 grid. Responses from dense LLaVA and LLaVA-MoR are shown with the ground-truth answers. The two questions share 26 of their 58 active D3 tokens.

Same Image, Different Questions (continued). The bakery case contrasts scene recognition with reasoning about the man's activity. The two questions use the same recursion budget but retain different visual tokens.

Case 2 | Same image, different questions

Q1. Where is this? Answer in one sentence.

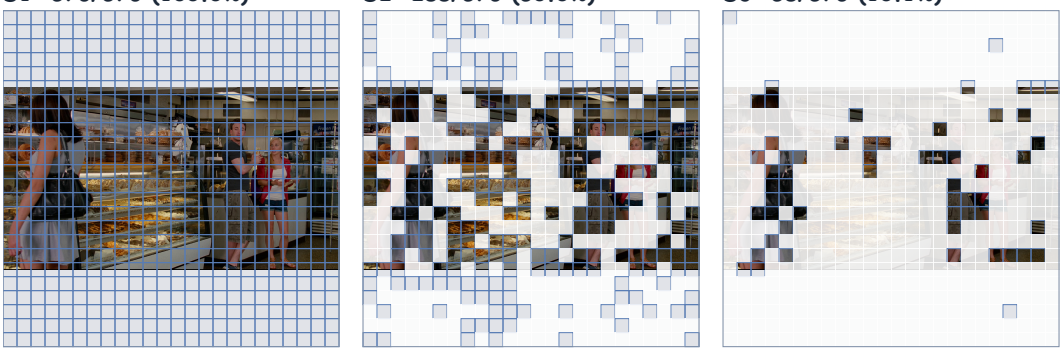
D1: 576/576 (100.0%) | D2: 288/576 (50.0%) | D3: 58/576 (10.1%)



LLaVA | Qwen-1.8B	LLaVA-MoR | Qwen-1.8B	Ground Truth Answer
Bakery	Bakery	This is a bakery.

Q2. Looking at the man, the food in his hand, and the display case, what is he most likely doing? Answer in one or two sentences and mention the key visual clues.

D1: 576/576 (100.0%) | D2: 288/576 (50.0%) | D3: 58/576 (10.1%)



LLaVA | Qwen-1.8B	LLaVA-MoR | Qwen-1.8B	Ground Truth Answer
The man is most likely ordering food from the display case, as he is holding a plate with food and is standing near the counter.	The man is most likely about to eat something from the display case.	The man is holding a pastry and eating it at the bakery counter, beside the display case.

Unfaded patches: active tokens. Gray margins: square padding.

Figure 9: Visual routing for different questions about the same bakery image. Unfaded patches denote active visual tokens on the padded 24×24 grid. Responses from dense LLaVA and LLaVA-MoR are shown with the ground-truth answers. The two questions share 24 of their 58 active D3 tokens.

Same Question, Different Images. We keep the question fixed and change only the image, comparing a birthday-cake scene with a person handling clothes beside a suitcase.

Case 3 | Same question, different images

Looking at the person and the important objects around them, what is this person most likely about to do? Answer in one or two sentences and mention the key visual clues.

Image A | Birthday cake



LLaVA | Qwen-1.8B

The person is most likely about to blow out the candles on the chocolate birthday cake.

LLaVA-MoR | Qwen-1.8B

The person is most likely about to blow out the candles on their birthday cake, as there are candles on the cake and the person is blowing them out.

Ground Truth Answer

The person is about to blow out the candles on the birthday cake.

Image B | Preparing for a trip



LLaVA | Qwen-1.8B

The person is most likely about to go shopping, as they are holding a suitcase and looking at clothes in a store window.

LLaVA-MoR | Qwen-1.8B

The person is most likely getting ready to go on a trip, as they are standing next to a suitcase and holding cloths, which are common items associated with travel.

Ground Truth Answer

The man is handling clothes in garment bags beside a suitcase and is most likely preparing for a trip.

Figure 10: Responses to the same question on different images. Both models identify candle blowing in image A. In image B, LLaVA-MoR infers travel preparation rather than shopping, consistent with the ground truth.