MoLE: Mixture of Latent Experts for Complementary Visual Reasoning

Anonymous authors
Paper under double-blind review

Abstract

Latent visual reasoning equips vision–language models with continuous intermediate states that can process visual evidence without explicit textual reasoning traces or repeated image operations. However, existing methods often allow multiple latent tokens to access the same visual evidence through shared value projections, providing no mechanism for them to extract complementary visual information; simply increasing the latent budget can therefore yield redundant latent representations. We argue that effective latent reasoning should encourage different latent tokens to extract complementary visual information, and thereby act as specialized visual experts. Based on this insight, we propose MoLE, a Mixture of Latent Experts framework that controls both what visual evidence each latent visual expert observes and how it transforms that evidence. MoLE isolates latent visual experts during evidence extraction and uses dedicated latent summary experts to aggregate the complementary representations of latent visual experts. A two-stage training pipeline first forces visual evidence through this latent pathway and then restores direct visual access, requiring neither predefined expert roles nor intermediate visual targets. Across five visual reasoning benchmarks, MoLE achieves an average score of 78.6, outperforming data-matched supervised fine-tuning by 4.9 and the strongest evaluated latent visual reasoning baseline at the same latent budget by 3.6. Representation analyses show lower latent-state similarity and more diverse visual attention, while masking the latent pathway reduces average performance by 9.2. These results demonstrate that specializing latent computation is more effective than merely increasing the number of latent tokens.

1 Introduction

Visual reasoning requires a vision–language model (VLM) to generate an answer through intermediate computation over visual inputs. The model must identify visual evidence, preserve fine-grained details, and combine cues that may be distributed across multiple objects, regions, or images. Existing approaches instantiate this intermediate process in different ways. Text-based chain-of-thought expresses it in discrete language (Wei et al., 2022; Xu et al., 2025; Huang et al., 2026). Interleaved approaches acquire additional visual observations during reasoning, but often require cropping, re-encoding, or external tools at inference (Shao et al., 2024; Su et al., 2025; Wu et al., 2026). Latent visual reasoning instead performs the intermediate computation in continuous hidden states (Li et al., 2026a; Qin et al., 2026; Li et al., 2026b). It therefore keeps the reasoning process internal without explicit textual traces or repeated image operations.

However, this advantage does not by itself make multiple latent tokens useful. In a common design, all latents can access the same visual tokens and process them through a shared value projection (Li et al., 2026a; Qin et al., 2026; Li et al., 2026b). These latents may therefore converge to redundant representations. Merely increasing the latent count can add more redundant states rather than new reasoning capacity.

This limitation motivates us to explore explicit specialization as a way to make better use of a fixed latent budget. A latent representation is determined by both *what* visual evidence it receives and *how* it transforms that evidence. When these two factors are shared across all latents, increasing the latent count can largely replicate the same computation and produce redundant representations. Differentiating both the available evidence and its transformation encourages latents to capture com-

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.

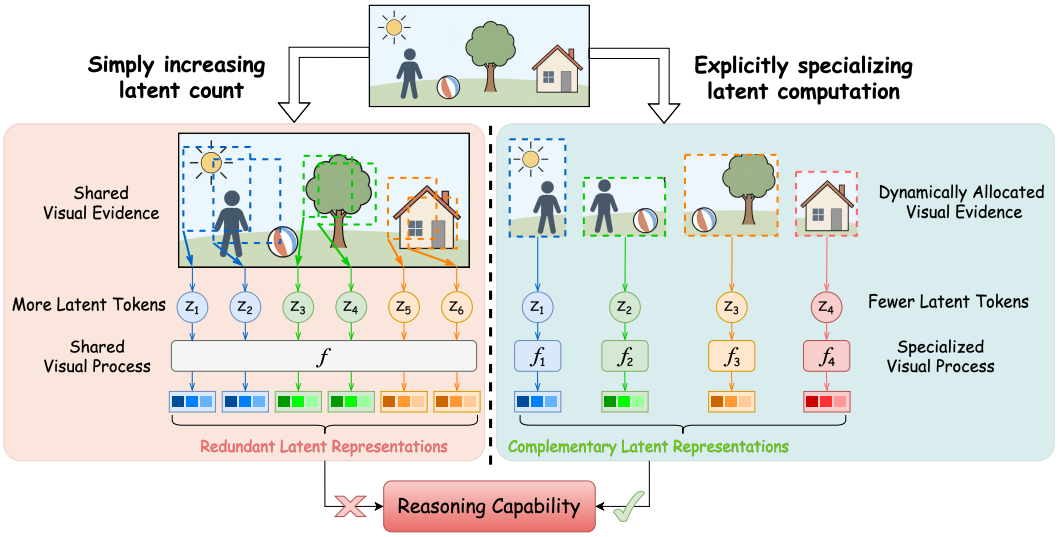


Figure 1: Two paths to enhancing latent visual reasoning. Simply increasing latent count while sharing visual evidence and transformations can produce redundant representations. Explicitly specializing what each latent observes and how it transforms that evidence encourages complementary representations and stronger reasoning without increasing the latent budget.

plementary visual information within a fixed budget. Figure 1 contrasts this explicit specialization with simply increasing the latent count under shared evidence and transformations.

Motivated by this insight, we introduce **MoLE** (**M**ixture of **L**atent **E**xperts) for complementary visual reasoning. At every decoder layer, sparse visual routing assigns each visual token to a subset of latent visual experts, differentiating the evidence available to them. Each latent visual expert further uses an Expert-Specific Value Adapter (ESVA) to transform its routed visual evidence differently. We isolate latent visual experts from one another to preserve their complementarity, then introduce latent summary experts that aggregate only the representations produced by the latent visual experts. Answer tokens attend to both latent expert sets, allowing complementary visual information and its aggregated summary to jointly support answer prediction.

MoLE is trained without predefined expert roles or intermediate visual targets. Because direct visual access can allow answer tokens to bypass the latent states, we follow LIVR (Li et al., 2026b) and adopt the two-stage training design to ensure that the latent pathway learns visual information useful for answer generation. Stage 1 imposes a visual attention bottleneck that forces visual evidence through the latent pathway. Stage 2 restores direct visual access, allowing answer generation to combine the learned latent representations with the original visual tokens. Training uses supervision only on answer tokens, together with the MoE loss for router optimization and a diversity objective for the ESVAs. Both stages use this same joint training objective. At inference, the latent states are computed during prefilling and require no additional image processing.

Across five visual reasoning benchmarks, MoLE achieves an average score of 78.6, improving data-matched SFT by 4.9 and the strongest evaluated latent visual reasoning baseline at the same latent budget by 3.6. It produces the lowest latent-state similarity and the highest KL diversity of latent attention-pattern among the compared latent visual reasoning methods. Moreover, masking the learned latent pathway lowers the average by 9.2 despite preserving direct visual access, showing that the answer generation uses information carried by the latent tokens. Doubling the latent budget does not improve the best configuration, further supporting the importance of specializing latent computation rather than merely increasing its size.

Our contributions are as follows:

- We identify the lack of complementary latent computation as a central limitation of latent visual reasoning and motivate encouraging latents to extract complementary visual information.

- We propose MoLE, a new latent visual reasoning method that combines dynamic visual allocation, expert-specific visual transformations, and topological latent aggregation without predefined expert roles or intermediate visual targets.
- We demonstrate consistent gains across visual reasoning benchmarks and provide ablation, representation, and intervention evidence that the learned latents are complementary and functionally used for answer generation.

2 Related Work

Text-based visual reasoning. Chain-of-thought (CoT) prompting improves reasoning in large language models (LLMs) by eliciting textual intermediate steps before an answer (Wei et al., 2022). This paradigm extends to vision–language models (VLMs): LLaVA-CoT organizes generation into summarization, visual interpretation, logical reasoning, and conclusion stages (Xu et al., 2025), whereas Vision-R1 combines a multimodal CoT cold start with reinforcement learning (Huang et al., 2026). These methods provide readable task decompositions, but expressing visual reasoning entirely through discrete text may omit fine-grained evidence, motivating methods that operate on visual information more directly.

Interleaved multimodal reasoning. Interleaved multimodal reasoning incorporates processed visual observations into textual trajectories. Visual CoT predicts a question-relevant bounding box, then combines re-encoded crop tokens with full-image tokens for answer generation (Shao et al., 2024). Tool-based methods acquire evidence more flexibly: Pixel Reasoner invokes zoom-in and frame-selection operations (Su et al., 2025); VTool-R1 learns to edit charts and tables with Python tools (Wu et al., 2026); and VISTA-R1 coordinates visual and symbolic tools over multi-turn interactions (Lu et al., 2026). These methods adapt evidence acquisition to the question but incur additional tool execution and visual processing at inference.

Latent visual reasoning. Latent reasoning instead represents intermediate computation as continuous states. COCONUT, CODI, and SeLaR explore this paradigm in LLMs (Hao et al., 2025; Shen et al., 2025; Fu & Luo, 2026), motivating its extension to visual reasoning without explicit image operations at inference. Latent visual reasoning methods differ mainly in how they supervise latent states. LVR reconstructs query-relevant visual tokens, while LANTERN and VaLR align latents with vision-encoder features (Li et al., 2026a; Viveiros et al., 2026; Jeon et al., 2026). Mirage distills latent tokens from ground-truth image embeddings, ILVR selectively distills relevant features from helper images, and Monet aligns observation-token representations under controlled access to auxiliary images (Yang et al., 2026; Dong et al., 2026; Wang et al., 2026). UniVLR compresses a shared canvas containing rendered reasoning traces and auxiliary images, whereas CoVT supervises distinct token groups with segmentation, depth, edge, and DINO targets (Jiang et al., 2026; Qin et al., 2026). In contrast, LIVR trains learnable latent tokens without intermediate targets, using an attention bottleneck and answer-token loss (Li et al., 2026b).

However, these methods generally let multiple latents access the same visual evidence through shared value projections, without explicitly encouraging complementary extraction. Increasing the latent budget can therefore yield redundant representations. MoLE instead sparsely routes visual evidence to isolated latent visual experts with expert-specific value transformations, then aggregates their complementary representations through dedicated latent summary experts.

3 Method

3.1 Preliminaries

Given a set of images $\mathcal{X} = \{X_1, \ldots, X_M\}$, a question prompt Q, and a target answer $Y = (y_1, \ldots, y_T)$, the task is to generate Y conditioned on $\mathcal{X}$ and Q. A vision–language model uses a vision encoder to extract image features and a visual projector to map them into the language-model hidden space, yielding N_v visual tokens $\mathbf{V}^{(0)} \in \mathbb{R}^{N_v \times d}$. Separately, a text tokenizer converts Q and Y into token ids, which the language model's input embedding layer then maps to prompt tokens $\mathbf{Q}^{(0)} \in \mathbb{R}^{N_q \times d}$ and answer tokens $\mathbf{Y}^{(0)} \in \mathbb{R}^{T \times d}$, respectively. A latent visual reasoning

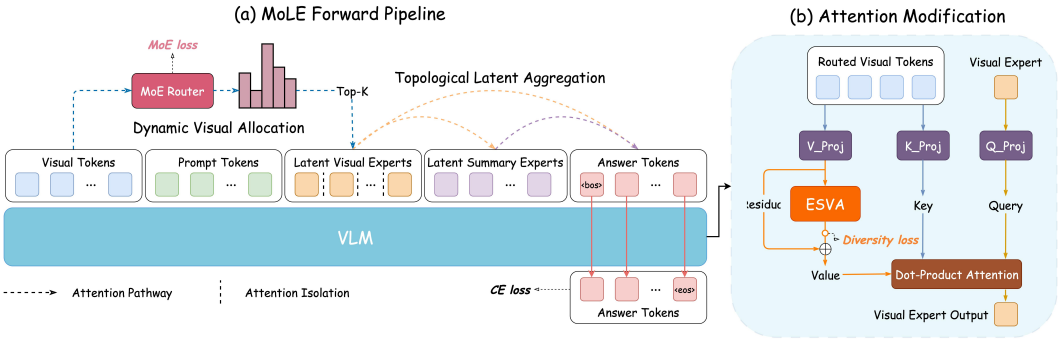


Figure 2: Overview of MoLE. (a) A layer-wise router sparsely assigns visual tokens to isolated latent visual experts; latent summary experts aggregate their representations, and answer tokens attend to both latent sets. (b) An Expert-Specific Value Adapter (ESVA) is used within the attention block when a latent visual expert attends to the routed visual tokens. Importantly, the ESVA is activated only for expert–visual-token connections selected by the router; all other attention interactions use the standard attention computation without an ESVA.

model then introduces N_z latent tokens between the prompt and answer:

$$\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_{N_z}] \in \mathbb{R}^{N_z \times d}. \tag{1}$$

The resulting training sequence concatenates these token embeddings in the following order:

$$\mathbf{H}^{(0)} = [\mathbf{V}^{(0)}; \mathbf{Q}^{(0)}; \mathbf{Z}; \mathbf{Y}^{(0)}]. \tag{2}$$

During the prefix forward pass, the latent tokens gather visual and textual context through the decoder attention layers. The model then generates Y autoregressively while attending to the resulting latent representations. This paradigm provides continuous intermediate computation. However, it does not specify which visual evidence each latent token should observe or how different latent tokens should transform that evidence, which can lead to redundant latent representations.

3.2 Mixture of Latent Experts

We introduce the components of MoLE by tracing the path of visual information through the model. We first describe how visual evidence is dynamically allocated and transformed, and then explain how the resulting latent representations are aggregated. Subsequently we present the two-stage training pipeline and the training objectives. Figure 2 provides an overview of the framework.

Dynamic visual allocation. To distribute visual evidence across latent computations, MoLE introduces E latent visual experts:

$$\mathbf{Z}^{\text{vis}} = [\mathbf{z}_1^{\text{vis}}, \ldots, \mathbf{z}_E^{\text{vis}}] \in \mathbb{R}^{E \times d}. \tag{3}$$

Unlike a standalone feed-forward expert in conventional MoE models (Shazeer et al., 2017; Fedus et al., 2022), each latent visual expert acts as a latent query in the decoder sequence. It attends to a routed subset of visual tokens and applies an expert-specific transformation to the corresponding visual evidence. To determine this subset, each decoder layer $\ell \in \{1, \ldots, L\}$ uses an independent router $\mathbf{W}_\ell^R \in \mathbb{R}^{E \times d}$. Let $\mathbf{h}_v^{(\ell-1)}$ denote the hidden state of visual token v entering layer ℓ. The router computes expert logits and a distribution over all latent visual experts:

$$r_{\ell,v,e} = (\mathbf{w}_{\ell,e}^R)^\top \mathbf{h}_v^{(\ell-1)}, \qquad \pi_{\ell,v,e} = \frac{\exp(r_{\ell,v,e})}{\sum_{j=1}^{E} \exp(r_{\ell,v,j})}. \tag{4}$$

Each visual token selects its K highest-scoring latent visual experts, $\mathcal{S}_{\ell,v} = \text{TopK}(\{r_{\ell,v,e}\}_{e=1}^{E}, K)$. If token v is assigned to latent visual expert e, the latent visual expert's latent query can attend to that token. All unselected expert–token edges are masked. Each latent visual expert consequently receives a dynamically allocated subset of the visual evidence at each decoder layer.

Router prior steering. Top-K selection determines whether a latent visual expert can access a visual token. Within the selected expert set for a visual token, we first renormalize the router logits:

$$\rho_{\ell,v,e} = \frac{\exp(r_{\ell,v,e})}{\sum_{j\in\mathcal{S}_{\ell,v}} \exp(r_{\ell,v,j})}, \qquad e \in \mathcal{S}_{\ell,v}, \tag{5}$$

We then convert the normalized score into an additive prior, $b_{\ell,v,e} = \alpha \log(K\rho_{\ell,v,e} + \varepsilon)$, where $\varepsilon > 0$ is a small constant that prevents taking the logarithm of zero and ensures numerical stability. We add this prior to the attention logit when the corresponding latent visual expert attends to that visual token. For attention head h,

$$\widetilde{a}^{(\ell,h)}_{e,v} = \frac{(\mathbf{q}^{(\ell,h)}_{z^{\text{vis}}_e})^\top \mathbf{k}^{(\ell,h)}_v}{\sqrt{d_h}} + b_{\ell,v,e}, \qquad e \in \mathcal{S}_{\ell,v}. \tag{6}$$

Here, $\mathbf{q}^{(\ell,h)}_{z^{\text{vis}}_e}$ is the query vector of latent visual expert e, and $\mathbf{k}^{(\ell,h)}_v$ is the key vector of visual token v, both at decoder layer ℓ and attention head h. The head dimension is denoted by d_h. The modified logits are then normalized together with the other accessible attention logits through softmax, producing the attention weights used to aggregate the corresponding value representations. The factor K centers the prior near zero when routing is uniform within the selected expert set. The router can therefore learn which visual experts are better suited to which visual tokens.

Expert-specific value adapter. Routing differentiates which visual evidence is available to each latent visual expert. Without expert-specific processing, however, all experts would still apply the same value projection and may produce redundant latent representations. We therefore equip each latent visual expert with an Expert-Specific Value Adapter (ESVA), implemented as an expert-specific two-layer MLP. For visual token v at layer ℓ, we write the transformation compactly as

$$\mathbf{u}_{\ell,v} = \text{flatten}(\mathbf{W}^V_\ell \mathbf{h}^{(\ell-1)}_v), \qquad \boldsymbol{\delta}_{\ell,v,e} = \text{ESVA}_e(\mathbf{u}_{\ell,v}), \qquad \widetilde{\mathbf{u}}_{\ell,v,e} = \boldsymbol{\delta}_{\ell,v,e} + \gamma_{\ell,e}\mathbf{u}_{\ell,v}. \tag{7}$$

Here, $\mathbf{W}^V_\ell$ is the value-projection matrix in the attention block of decoder layer ℓ, and $\mathbf{h}^{(\ell-1)}_v$ is the hidden state of visual token v entering that layer. The projected value is flattened across attention heads to form $\mathbf{u}_{\ell,v}$, while $\gamma_{\ell,e}$ is a learned residual scale. The resulting $\widetilde{\mathbf{u}}_{\ell,v,e}$ serves as the new value representation for the routed expert–token connection and is used in the subsequent attention-weighted value aggregation. ESVAs bring different visual information transformations to various latent visual experts. The detailed ESVA architecture is provided in Appendix A.1.1.

Topological latent aggregation. After the latent visual experts extract complementary visual information, MoLE introduces S latent summary experts to aggregate their representations:

$$\mathbf{Z}^{\text{sum}} = [\mathbf{z}^{\text{sum}}_1, \ldots, \mathbf{z}^{\text{sum}}_S] \in \mathbb{R}^{S\times d}. \tag{8}$$

The complete decoder input sequence is therefore $[\mathbf{V}^{(0)}; \mathbf{Q}^{(0)}; \mathbf{Z}^{\text{vis}}; \mathbf{Z}^{\text{sum}}; \mathbf{Y}^{(0)}]$. Direct interaction among latent visual experts during evidence extraction could prematurely mix their representations and weaken their complementarity. We therefore block attention between latent visual experts and use the latent summary experts to aggregate their complementary representations into a more complete visual representation. Answer tokens can attend to both visual and summary experts, allowing answer generation to use both the complementary evidence and its aggregated representation.

Two-stage training pipeline. Under conventional end-to-end training, answer tokens can directly attend to the original visual tokens, creating a shortcut that allows the model to minimize the answer loss while ignoring the latent pathway. The resulting latent tokens may therefore carry little visual information that is useful for answer generation. Following the two-stage training strategy of LIVR (Li et al., 2026b), we first make the latent pathway informative and then integrate it with direct visual access. In Stage 1, we block direct attention from ordinary prompt and answer tokens to visual tokens, so only latent visual experts can access routed visual evidence. This visual attention bottleneck trains the latent visual experts to extract and preserve task-relevant information. Stage 2 restores direct visual access for prompt and answer tokens, allowing answer generation to rely jointly on the learned latent representations and the original visual tokens. We provide the detailed attention-bottleneck configurations for both stages in Appendix A.1.2.

Training objectives. We first apply language-modeling supervision only to answer tokens:

$$\mathcal{L}_{\text{CE}} = -\frac{1}{T}\sum_{t=1}^{T} \log p_\Theta(y_t \mid \mathcal{X}, Q, \mathbf{Z}^{\text{vis}}, \mathbf{Z}^{\text{sum}}, y_{<t}). \tag{9}$$

Furthermore, we regularize the router with the load-balancing loss used in conventional MoE models (Shazeer et al., 2017; Fedus et al., 2022) and the router z-loss introduced by ST-MoE (Zoph et al., 2022). To distinguish the experts selected by Top-K routing from their soft routing probabilities, we define the binary selection variable

$$g_{\ell,v,e} = \begin{cases} 1, & \text{if expert } e \text{ is selected for visual token } v \text{ at layer } \ell, \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

The selection frequency and mean routing probability of expert e are

$$f_e = \frac{1}{KLN_v} \sum_{\ell=1}^{L} \sum_{v=1}^{N_v} g_{\ell,v,e}, \qquad \overline{\pi}_e = \frac{1}{LN_v} \sum_{\ell=1}^{L} \sum_{v=1}^{N_v} \pi_{\ell,v,e}. \tag{11}$$

We define

$$\mathcal{L}_{\text{balance}} = E \sum_{e=1}^{E} \text{sg}(f_e)\overline{\pi}_e, \qquad \mathcal{L}_z = \frac{1}{LN_v} \sum_{\ell=1}^{L} \sum_{v=1}^{N_v} \left(\log \sum_{e=1}^{E} \exp(r_{\ell,v,e}) \right)^2, \tag{12}$$

where sg stops gradients through the discrete selection frequency. The load-balancing loss prevents visual experts from being persistently bypassed and becoming inactive, while the z-loss controls the scale of the router logits, thereby stabilizing gradients during training. We refer to their weighted combination, $\mathcal{L}_{\text{MoE}} = \lambda_{\text{balance}}\mathcal{L}_{\text{balance}} + \lambda_z\mathcal{L}_z$, as the *MoE loss*.

In addition, to discourage different ESVAs from producing redundant representations for the same visual token, we minimize their pairwise squared cosine similarity. With $\overline{\boldsymbol{\delta}}_{\ell,v,e} = \boldsymbol{\delta}_{\ell,v,e}/\|\boldsymbol{\delta}_{\ell,v,e}\|_2$, the diversity loss is

$$\mathcal{L}_{\text{div}} = \frac{1}{LN_v} \sum_{\ell=1}^{L} \sum_{v=1}^{N_v} \frac{2}{K(K-1)} \sum_{\substack{i<j \\ i,j \in \mathcal{S}_{\ell,v}}} \left(\overline{\boldsymbol{\delta}}_{\ell,v,i}^{\top} \overline{\boldsymbol{\delta}}_{\ell,v,j} \right)^2. \tag{13}$$

For $K = 1$, no selected expert pair exists and we define $\mathcal{L}_{\text{div}} = 0$. The complete objective is $\mathcal{L} = \mathcal{L}_{\text{CE}} + \mathcal{L}_{\text{MoE}} + \lambda_{\text{div}}\mathcal{L}_{\text{div}}$. Both training stages use the same joint objective.

4 Experiments

We adopt Qwen2.5-VL-7B (Bai et al., 2025b) as the backbone VLM. In both training stages, we fine-tune the language backbone with LoRA (Hu et al., 2022) on its attention and MLP blocks, while keeping the vision encoder and projector frozen. In addition to the LoRA parameters, we unfreeze only the embedding rows corresponding to the $E + S$ newly introduced latent experts.

We construct a 63K-example training set by sampling from CoVT-Dataset (Qin et al., 2026), which combines vision-centric data from LLaVA-OneVision (Li et al., 2025) with counting and relative-depth samples derived from TallyQA (Acharya et al., 2019) and ADE20K (Zhou et al., 2017). We do not sample directly from any of the evaluation benchmarks described below, thereby avoiding direct reuse of their evaluation splits. Detailed data sampling, processing, and decontamination against the evaluation benchmarks are described in Appendix A.2.

4.1 Evaluation Benchmarks

We evaluate across five benchmarks with distinct perceptual demands. CVBench (Tong et al., 2024) separates 2D spatial relations and object counting from 3D depth ordering and relative-distance understanding. BLINK (Fu et al., 2024) broadens this coverage by recasting 14 classical computer-vision tasks as single- and multi-image multiple-choice questions, including visual correspondence, relative depth, and multi-view reasoning. RealWorldQA (RWQA) (xAI, 2024) shifts the evaluation to natural physical scenes, where questions involve properties such as relative size, direction, and navigability. V* (Wu & Xie, 2024) focuses specifically on visual search, requiring models to locate and reason about details in high-resolution, visually crowded images. Finally, POPE (Li et al., 2023) complements these capability-oriented benchmarks by probing resistance to object hallucination through binary questions about object presence. For CVBench, we report accuracy separately on its 2D and 3D subsets. We report F1 for POPE and accuracy for the remaining benchmarks. Avg. is the unweighted mean of these six metrics.

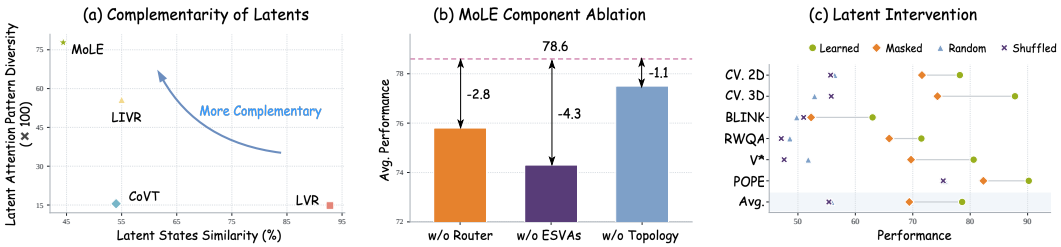


Figure 3: Complementarity, component ablations, and latent-state interventions. (a) Lower latent-state similarity and higher attention-pattern diversity indicate more complementary latent representations. (b) Performance after independently removing each component; the dashed line marks the full MoLE result. (c) Performance under inference-time interventions on the learned latent states.

4.2 Main Results

Baselines. We compare MoLE with several categories of reasoning models. GPT-4o and Claude-4-Sonnet serve as proprietary VLM references. We also report the Qwen2.5-VL-7B base model and Qwen2.5-VL-7B-SFT, obtained by directly fine-tuning the same backbone on the same training set. The former establishes the pretrained starting point, while the latter provides a data-matched control for isolating gains attributable to MoLE rather than exposure to the training data. Vision-R1 (Huang et al., 2026) represents text-based visual reasoning, while PixelReasoner (Su et al., 2025) interleaves textual reasoning with image operations to actively acquire additional visual evidence during inference. We further compare with CoVT, LVR, and LIVR (Qin et al., 2026; Li et al., 2026a;b) as representative latent visual reasoning methods. For local evaluation, we use the official publicly released checkpoints of Vision-R1, PixelReasoner, CoVT, and LVR. We train LIVR on the same training set as MoLE. Detailed training configurations for our default MoLE model, Qwen2.5-VL-7B-SFT, and LIVR are provided in Appendix A.3.

Evaluation configurations. We evaluate CoVT, LVR, and LIVR with different latent budgets of 8 and 16. For LVR, we can simply set the number of reasoning steps directly. For CoVT, we use two publicly released checkpoints: CoVT-Seg with 8 steps and CoVT-Seg-Depth-DINO with 16 steps. Because LIVR fixes its latent budget during training, we train separate models with 8 and 16 latent tokens on the same training set. Our main MoLE configuration uses a total latent budget of $E + S = 8$. We evaluate all open models with our evaluation scripts, which follow the official evaluation configurations of the respective benchmarks. For a fair comparison, every evaluated baseline and MoLE use the same scripts and benchmark-specific configurations. Detailed evaluation configurations are provided in Appendix A.4. For the proprietary models, results on CVBench, BLINK, RWQA, and V* are taken from CoVT (Qin et al., 2026); their POPE scores follow Sarkar et al. (2025) for GPT-4o and the Moondream benchmark (M87 Labs, 2026) for Claude-4-Sonnet. Since CoVT does not report CVBench results separately for its 2D and 3D subsets, we use its reported GPT-4o and Claude-4-Sonnet results to calculate separate 2D and 3D scores for each model. The calculation is provided in Appendix A.5.

Results. Table 1 shows that MoLE achieves the highest average score of 78.6 and the best result on all six reported metrics among the evaluated models. Compared with data-matched SFT, it improves every metric and raises the average by 4.9, showing that the gain cannot be explained solely by exposure to CoVT-Dataset. At the same latent budget of 8, MoLE outperforms the strongest latent visual reasoning baseline, LIVR, by 3.6 on average. Simply doubling the latent budget does not yield consistent gains: LVR degrades, while CoVT and LIVR improve only modestly. Even the strongest 16-step result remains 3.1 below MoLE with 8 latents. These results suggest that organizing a fixed latent budget around information complementarity is more effective than increasing the latent count alone. The largest improvements occur on CVBench-3D and BLINK, where predictions depend heavily on spatial structure and fine-grained perceptual cues. Improvements on V* and POPE further suggest that the benefits extend to visual search and reliable object grounding.

Table 1: Overall performance across five visual reasoning benchmarks.

Method	CVBench 2D	CVBench 3D	BLINK	RWQA	V*	POPE	Avg.
Proprietary models							
GPT-4o†	74.6	83.9	63.0	69.7	42.9	85.6	70.0
Claude-4-Sonnet†	73.5	79.1	39.6	63.7	15.2	84.6	59.3
Base and data-matched models							
Qwen2.5-VL-7B	75.4	73.3	55.6	68.9	76.6	85.0	72.5
Qwen2.5-VL-7B-SFT	74.2	76.9	56.3	69.5	77.5	87.6	73.7
Explicit/interleaved reasoning							
Vision-R1	74.2	77.2	52.6	68.8	78.5	88.6	73.3
PixelReasoner	<u>75.8</u>	78.6	55.5	67.1	76.4	86.8	73.4
Latent visual reasoning							
LVR (steps=8)	75.3	81.9	55.6	64.7	78.4	84.3	73.4
LVR (steps=16)	74.8	81.4	52.8	62.5	<u>80.1</u>	80.1	72.0
CoVT (steps=8)	74.5	81.6	56.3	68.4	77.5	88.6	74.5
CoVT (steps=16)	74.3	<u>83.9</u>	56.8	<u>70.1</u>	78.4	88.5	75.3
LIVR (steps=8)	73.8	83.2	56.7	69.1	79.4	87.7	75.0
LIVR (steps=16)	74.6	83.7	<u>57.4</u>	69.7	78.6	<u>88.9</u>	<u>75.5</u>
Ours							
MoLE	**78.2**	**87.8**	**63.0**	**71.5**	**80.6**	**90.2**	**78.6**
Δ *vs. best competitor*	+2.4	+3.9	+5.6	+1.4	+0.5	+1.3	+3.1

Bold/underline: best/second best among open models. Δ denotes the absolute difference from the second-best open model in each column. Proprietary-model results were obtained under different evaluation configurations and are included for reference only. †: externally reported.

4.3 Quantitative Analysis

Where do the performance gains come from? Starting from the default configuration, we independently retrain three variants that remove sparse visual routing, ESVAs, or topological latent aggregation (Figure 3(b)). Without the router, every visual token is made equally accessible to all visual experts. Removing ESVAs retains the shared value projection but eliminates the subsequent expert-specific transformations. Removing topological latent aggregation instead restores conventional causal attention among visual experts, allowing their representations to mix directly. ESVA produces the largest performance drop when removed, indicating that expert-specific visual transformations and the resulting complementary latent representations are particularly important for visual reasoning. The router also contributes substantially by assigning different visual evidence to different visual experts and strengthening their complementarity. Finally, the degradation under causal expert mixing indicates that premature interaction can blur complementary information before it reaches the summary experts, ultimately reducing model performance.

Are the latents used for answer generation? We intervene on the learned latent states at inference in three ways: replacing them with random values, shuffling the values within each latent vector, or masking the latent key–value states from answer tokens (Figure 3(c)). All three interventions substantially degrade performance. Randomization and shuffling cause the largest drops, but they also introduce strong distribution shifts: meaningless latent states may actively disrupt the decoder's normal language modeling. Masking provides a cleaner test. Even though answer tokens retain direct access to the visual tokens, removing access to the latent pathway lowers the average by 9.2, showing that the learned latents carry information that is used for answer generation.

Do the latents become more complementary? We assess latent redundancy from two perspectives. First, pairwise squared cosine similarity between latent-token hidden states measures their numerical similarity. Second, pairwise KL diversity between the visual-token attention patterns of latents measures how differently they attend to visual information. Both metrics are aggregated over the evaluation data, scaled by 100. As shown in Figure 3(a), MoLE achieves the lowest latent similarity

Table 2: Latent-allocation, training-stage, and training-objective ablations. (a) Allocation of visual (E) and summary (S) experts. (b) Training stages and objectives.

(a) Latent allocation

Budget	E	S	Avg.
8	4	4	**78.6**
	6	2	77.2
	7	1	<u>77.4</u>
	8	0	75.7
16	8	8	**77.8**
	12	4	76.8
	14	2	77.0
	15	1	<u>77.4</u>
	16	0	75.4

(b) Stages and objectives

Ablation	Configuration	Avg.
Training stage	Stage 1 only	69.0
	Stage 2 only	<u>75.7</u>
	Stage 1 + Stage 2	**78.6**
Training objective	CE only	76.0
	CE + MoE	76.9
	CE + Diversity	<u>77.1</u>
	CE + MoE + Diversity	**78.6**

and the highest attention-pattern diversity among the compared latent visual reasoning methods. Detailed formulations of both metrics are provided in Appendix A.6.

4.4 Ablation Studies

Latent allocation. We vary the allocation of visual and summary experts under fixed latent budgets of 8 and 16 (Table 2(a)). The balanced configurations, $E = 4, S = 4$ and $E = 8, S = 8$, perform best within their respective budgets. However, the best configuration with 16 latents does not outperform its 8-latent counterpart, indicating that simply increasing the latent budget does not necessarily improve visual reasoning performance. We also set $S = 0$ at both budgets to test whether summary experts are necessary. Although these variants devote the entire latent budget to visual experts, their average performance drops by 2.9 and 2.4, respectively. This result shows that allocating capacity to summary experts, which integrate the complementary representations produced by visual experts, is needed.

Training stages and objectives. Table 2(b) examines both the training pipeline and its auxiliary objectives. All stage variants retain the other settings of the default MoLE model and are trained for three epochs in total: Stage 1 only uses Stage 1 for all three epochs, Stage 2 only uses Stage 2 for all three epochs, and Stage 1 + Stage 2 follows the default schedule of two Stage 1 epochs and one Stage 2 epoch. The complete two-stage pipeline outperforms either stage alone. Stage 1 shows that the latent pathway can preserve substantial visual information under the visual attention bottleneck, while the gain over Stage 2-only training supports first making this pathway informative before restoring direct visual access. Regarding the objectives, the MoE loss improves expert utilization and router stability, while the diversity loss encourages distinct ESVA transformations; combining both yields the best performance.

More ablations. More ablations and results across random seeds are provided in Appendix A.7.

5 Conclusion

We introduced MoLE, a framework that organizes latent visual reasoning around complementary latent computation rather than latent count. MoLE routes visual evidence to isolated latent visual experts, applies expert-specific visual transformations to routed visual evidence through ESVAs, and aggregates the resulting complementary latent representations with latent summary experts. Across five benchmarks, MoLE improves data-matched SFT by 4.9 points on average and outperforms the evaluated latent reasoning methods. Ablations validate the contributions of the architecture and training objectives. Representation analyses show lower latent-state similarity and more diverse visual-attention patterns of latents than other latent visual reasoning methods. Inference-time interventions further confirm the usefulness of our latents.

AI USE STATEMENT

In our work, AI-assisted tools were only used for minor language editing and polishing. We reviewed the resulting text and take full responsibility for all the content of the paper.

REPRODUCIBILITY STATEMENT

We train MoLE from the open-source Qwen2.5-VL-7B checkpoint (Bai et al., 2025b) using a 63K-example subset constructed from the open-source CoVT-Dataset (Qin et al., 2026). To ensure full reproducibility, we provide the complete anonymous code base in supplementary material.

REFERENCES

Manoj Acharya, Kushal Kafle, and Christopher Kanan. TallyQA: Answering complex counting questions. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(1):8076–8084, 2019. doi: 10.1609/aaai.v33i01.33018076.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-VL technical report. arXiv preprint arXiv:2511.21631, 2025a.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report. arXiv preprint arXiv:2502.13923, 2025b.

Ali Furkan Biten, Ruben Tito, Andres Mafla, Lluis Gomez, Marcal Rusinol, Ernest Valveny, C. V. Jawahar, and Dimosthenis Karatzas. Scene text visual question answering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4291–4301, 2019.

Guiming Hardy Chen, Shunian Chen, Ruifei Zhang, Junying Chen, Xiangbo Wu, Zhiyi Zhang, Zhihong Chen, Jianquan Li, Xiang Wan, and Benyou Wang. ALLaVA: Harnessing GPT-4V-synthesized data for lite vision-language models. arXiv preprint arXiv:2402.11684, 2024.

Erfei Cui, Yinan He, Zheng Ma, Zhe Chen, Hao Tian, Weiyun Wang, Kunchang Li, Yi Wang, Wenhai Wang, Xizhou Zhu, Lewei Lu, Tong Lu, Yali Wang, Limin Wang, Yu Qiao, and Jifeng Dai. ShareGPT-4o: Comprehensive multimodal annotations with GPT-4o. Project page, 2024. URL `https://sharegpt4o.github.io/`. Accessed September 26, 2026.

Shuai Dong, Siyuan Wang, Xingyu Liu, Chenglin Li, Haowen Hou, and Zhongyu Wei. Interleaved latent visual reasoning with selective perceptual modeling. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 29316–29335. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.1351.

Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 107:3–11, 2018. doi: 10.1016/j.neunet.2017.12.012.

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.

Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24 (6):381–395, 1981. doi: 10.1145/358669.358692.

Renyu Fu and Guibo Luo. SeLaR: Selective latent reasoning in large language models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7073–7087. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.320.

Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A. Smith, Wei-Chiu Ma, and Ranjay Krishna. BLINK: Multimodal large language models can see but not perceive. In *Computer Vision – ECCV 2024*, pp. 148–166, 2024. doi: 10.1007/978-3-031-73337-6_9.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason E. Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. In *Second Conference on Language Modeling*, 2025.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Wenxuan Huang, Bohan Jia, Shaosheng Cao, Zheyu Ye, Fei Zhao, Zhe Xu, Yao Hu, and Shaohui Lin. Vision-R1: Incentivizing reasoning capability in multimodal large language models. In *International Conference on Learning Representations*, pp. 63794–63812, 2026.

Byungwoo Jeon, Yoonwoo Jeong, Hyunseok Lee, Minsu Cho, and Jinwoo Shin. Vision-aligned latent reasoning for multi-modal large language model. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Houcheng Jiang, Jiajun Fu, Junfeng Fang, Chen Gao, Xiang Wang, Xiangnan He, and Yong Li. UniVLR: Unifying text and vision in visual latent reasoning for multimodal LLMs. arXiv preprint arXiv:2605.11856, 2026.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2901–2910, 2017.

Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. The hateful memes challenge: Detecting hate speech in multimodal memes. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Bangzheng Li, Ximeng Sun, Jiang Liu, Ze Wang, Jialian Wu, Xiaodong Yu, Hao Chen, Emad Barsoum, Muhao Chen, and Zicheng Liu. Latent visual reasoning. In *International Conference on Learning Representations*, pp. 148076–148090, 2026a.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-OneVision: Easy visual task transfer. *Transactions on Machine Learning Research*, 2025.

Kelvin Li, Chuyi Shang, Leonid Karlinsky, Rogerio Feris, Trevor Darrell, and Roei Herzig. Latent implicit visual reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 33457–33466, 2026b.

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. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.20.

Fangyu Liu, Guy Emerson, and Nigel Collier. Visual spatial reasoning. *Transactions of the Association for Computational Linguistics*, 11:635–651, 2023. doi: 10.1162/tacl_a_00566.

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

David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. doi: 10.1023/B:VISI.0000029664.99615.94.

Meng Lu, Ran Xu, Yi Fang, Wenxuan Zhang, Yue Yu, Gaurav Srivastava, Yuchen Zhuang, Mohamed Elhoseiny, Charles Fleming, Carl Yang, Zhengzhong Tu, Yang Xie, Guanghua Xiao, Di Jin, Wenqi Shi, and Xuan Wang. Scaling agentic reinforcement learning for tool-integrated reasoning in VLMs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26518–26529, 2026.

Pan Lu, Liang Qiu, Jiaqi Chen, Tony Xia, Yizhou Zhao, Wei Zhang, Zhou Yu, Xiaodan Liang, and Song-Chun Zhu. IconQA: A new benchmark for abstract diagram understanding and visual language reasoning. In *The 35th Conference on Neural Information Processing Systems (NeurIPS) Track on Datasets and Benchmarks*, 2021.

M87 Labs. Moondream documentation: Performance benchmarks. Moondream Documentation, 2026. URL `https://docs.moondream.ai/`. Accessed September 15, 2026.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Yiming Qin, Bomin Wei, Jiaxin Ge, Konstantinos Kallidromitis, Stephanie Fu, Trevor Darrell, and Xudong Wang. Chain-of-Visual-Thought: Teaching VLMs to see and think better with continuous visual tokens. In *European Conference on Computer Vision*, 2026. Oral presentation.

Pritam Sarkar, Sayna Ebrahimi, Ali Etemad, Ahmad Beirami, Sercan Ö. Arık, and Tomas Pfister. Mitigating object hallucination in MLLMs via data-augmented phrase-level alignment. In *International Conference on Learning Representations*, 2025.

Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-OKVQA: A benchmark for visual question answering using world knowledge. In *Computer Vision – ECCV 2022*, pp. 146–162, 2022. doi: 10.1007/978-3-031-20074-8_9.

Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. Visual CoT: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning. In *Advances in Neural Information Processing Systems*, volume 37, pp. 8612–8642, 2024. doi: 10.52202/079017-0275.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017.

Zhenyi Shen, Hanqi Yan, Linhai Zhang, Zhanghao Hu, Yali Du, and Yulan He. CODI: Compressing chain-of-thought into continuous space via self-distillation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 677–693. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.36.

Wenhao Shi, Zhiqiang Hu, Yi Bin, Junhua Liu, Yang Yang, See-Kiong Ng, Lidong Bing, and Roy Ka-Wei Lee. Math-LLaVA: Bootstrapping mathematical reasoning for multimodal large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 4663–4680. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-emnlp.268.

Alex Su, Haozhe Wang, Weiming Ren, Fangzhen Lin, and Wenhu Chen. Pixel Reasoner: Incentivizing pixel space reasoning via curiosity-driven reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 38, pp. 8222–8251, 2025. doi: 10.52202/085713-0280.

Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, Austin Wang, Rob Fergus, Yann LeCun, and Saining Xie. Cambrian-1: A fully open, vision-centric exploration of multimodal LLMs. In *Advances in Neural Information Processing Systems*, volume 37, pp. 87310–87356. Curran Associates, Inc., 2024. doi: 10.52202/079017-2771.

André G. Viveiros, Nuno Gonçalves, Matthias Lindemann, and André F. T. Martins. LANTERN: Latent visual structured reasoning. In *ICLR 2026 Workshop on Multimodal Intelligence*, 2026.

Qixun Wang, Yang Shi, Yifei Wang, Yuanxing Zhang, Pengfei Wan, Kun Gai, Xianghua Ying, and Yisen Wang. Monet: Reasoning in latent visual space beyond image and language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12030–12040, 2026.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837, 2022. doi: 10.52202/068431-1800.

Mingyuan Wu, Jingcheng Yang, Jize Jiang, Meitang Li, Kaizhuo Yan, Hanchao Yu, Minjia Zhang, ChengXiang Zhai, and Klara Nahrstedt. VTool-R1: VLMs learn to think with images via reinforcement learning on multimodal tool use. In *International Conference on Learning Representations*, pp. 78298–78319, 2026.

Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal LLMs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13084–13094, 2024.

xAI. RealWorldQA. Hugging Face dataset, 2024. URL `https://huggingface.co/datasets/xai-org/RealworldQA`. Accessed September 25, 2026.

Guowei Xu, Peng Jin, Ziang Wu, Hao Li, Yibing Song, Lichao Sun, and Li Yuan. LLaVA-CoT: Let vision language models reason step-by-step. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2087–2098, 2025.

Zeyuan Yang, Xueyang Yu, Delin Chen, Maohao Shen, and Chuang Gan. Machine mental imagery: Empower multimodal reasoning with latent visual tokens. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 33510–33520, 2026.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. In *Advances in Neural Information Processing Systems*, volume 32, pp. 12360–12371, 2019.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ADE20K dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 633–641, 2017.

Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. Visual7W: Grounded question answering in images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4995–5004, 2016.

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

A Appendix

A.1 Method Details

This section provides further implementation details for the ESVA and the two-stage attention bottleneck.

A.1.1 EXPERT-SPECIFIC VALUE ADAPTER

ESVA applies a layer-specific normalization followed by an expert-specific two-layer MLP to the output of the shared value projection. At decoder layer ℓ, the hidden state of visual token v is first processed by the backbone's standard value projection:

$$\mathbf{u}_{\ell,v} = \mathrm{flatten}\Big(\mathbf{W}_\ell^V \mathbf{h}_v^{(\ell-1)}\Big) \in \mathbb{R}^{d_V}, \qquad d_V = H_{\mathrm{KV}} d_h, \tag{14}$$

where H_{KV} and d_h denote the number and dimension of the key–value heads, respectively. This shared projection is retained for every expert. We normalize its output using a layer-specific RMSNorm (Zhang & Sennrich, 2019):

$$\widehat{\mathbf{u}}_{\ell,v} = \boldsymbol{\omega}_\ell \odot \frac{\mathbf{u}_{\ell,v}}{\sqrt{d_V^{-1}\|\mathbf{u}_{\ell,v}\|_2^2 + \varepsilon_{\mathrm{norm}}}}, \tag{15}$$

where $\boldsymbol{\omega}_\ell \in \mathbb{R}^{d_V}$ is learned independently for each decoder layer and $\varepsilon_{\mathrm{norm}}$ ensures numerical stability.

Each latent visual expert e then applies a bias-free two-layer MLP with hidden dimension d_b:

$$\boldsymbol{\delta}_{\ell,v,e} = \mathbf{U}_e\ \mathrm{SiLU}(\mathbf{D}_e \widehat{\mathbf{u}}_{\ell,v}), \qquad \mathbf{D}_e \in \mathbb{R}^{d_b \times d_V}, \quad \mathbf{U}_e \in \mathbb{R}^{d_V \times d_b}. \tag{16}$$

Here, SiLU denotes the sigmoid-weighted linear unit activation (Elfwing et al., 2018). The matrices $\mathbf{D}_e$ and $\mathbf{U}_e$ are specific to expert e but shared across decoder layers. In the default configuration, $d_b = 32$. For an expert–visual-token connection selected by the router, the MLP output is combined with the original projected value using a learned layer- and expert-specific scale $\gamma_{\ell,e}$:

$$\widetilde{\mathbf{u}}_{\ell,v,e} = \boldsymbol{\delta}_{\ell,v,e} + \gamma_{\ell,e}\mathbf{u}_{\ell,v}, \qquad e \in \mathcal{S}_{\ell,v}. \tag{17}$$

The resulting $\widetilde{\mathbf{u}}_{\ell,v,e}$ replaces the standard value on that routed connection during attention-weighted value aggregation. Thus, ESVA does not replace the backbone value projection; it performs an additional expert-specific transformation of its output. The key projection remains unchanged, and all other attention interactions use the standard value representation.

A.1.2 TWO-STAGE ATTENTION BOTTLENECK

Figure 4 shows the complete information pathways used in the two training stages. In both stages, latent visual experts access visual tokens through router-selected connections and remain isolated from one another, while latent summary experts aggregate only the visual experts. In Stage 1, prompt and answer tokens cannot directly access visual tokens, so visual evidence must reach the answer through the latent pathway. This bottleneck encourages the latent experts to carry information useful for answer generation. Stage 2 starts from the Stage 1 checkpoint and restores direct visual access for prompt and answer tokens, allowing the answer to combine the learned latent representations with fine-grained visual evidence. The two stages optimize the same parameters with the same objective, and all MoLE-specific pathways remain unchanged.

A.2 DATASET CONSTRUCTION

We construct the training set from CoVT-Dataset (Qin et al., 2026). The following sections describe the source data, our task-balanced sampling strategy, and the online decontamination procedure used to produce the final 63K examples.

A.2.1 SOURCE DATASET

CoVT-Dataset aggregates vision-centric subsets of LLaVA-OneVision (Li et al., 2025) with a re-filtered TallyQA subset (Acharya et al., 2019) and relative-depth questions derived from ADE20K (Zhou et al., 2017). It covers a broad range of tasks, including counting, compositional and spatial reasoning, visual mathematics, OCR, knowledge-based VQA, and detailed image description.

The CoVT-Dataset snapshot used in our experiments contains 889,489 records. Because a record may contain multiple rounds, these records comprise 1,693,665 question–answer pairs in total. The distribution is highly imbalanced: the TallyQA source accounts for more than 40% of the records, whereas several spatial, depth, OCR, and safety-related sources are substantially smaller. Uniform sampling would largely preserve this skew, so we instead construct a task-balanced subset.

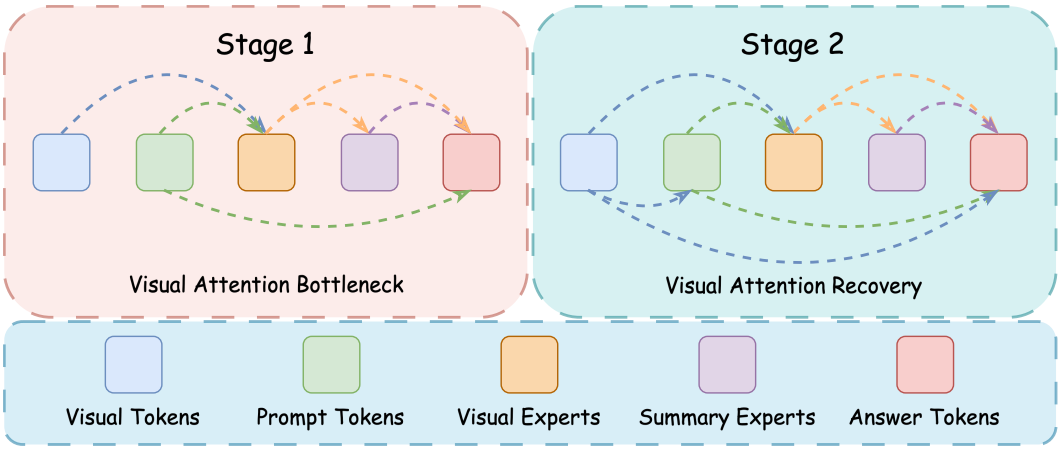


Figure 4: Attention pathways in the two-stage training pipeline. Dashed arrows denote permitted information flow between token groups.

Table 3: Composition of the task-balanced 63K training set. Counts separated by "+" correspond to the listed sources in the same order.

Task family	Source group	Samples
Visual math reasoning	MathV360K identity group (Shi et al., 2024); IconQA (Lu et al., 2021)	3,150 + 3,150
Inferential VQA	ALLaVA-LAION; ALLaVA-VFLAN (Chen et al., 2024)	3,150 + 3,150
Knowledge VQA	A-OKVQA (Schwenk et al., 2022)	6,300
Compositional reasoning	CLEVR (Johnson et al., 2017)	6,300
Safety classification	Hateful Memes (Kiela et al., 2020)	6,300
Detailed captioning	Image textualization; numeric-ID group; `sa`; ShareGPT-4o (Cui et al., 2024)	4×1,575
Scene-text OCR	ST-VQA (Biten et al., 2019)	6,300
Counting	TallyQA (Acharya et al., 2019)	6,300
Multiple-choice VQA	Visual7W (Zhu et al., 2016)	6,300
Spatial understanding	VSR (Liu et al., 2023); ADE20K relative depth (Zhou et al., 2017)	2,152 + 4,148
Total		63,000

A.2.2 Task-Balanced Sampling

We organize the source records into ten high-level task families and allocate 6,300 examples to each family. Table 3 summarizes the resulting composition. For task families supported by multiple sources, we balance those sources where their capacities permit. When a source does not contain enough examples to meet an equal quota, we retain all available examples without replacement and fill the remaining quota from another source within the same task family. For every selected record, we use only the first question–answer pair as the training example.

After assigning equal quotas to the ten task families, we further divide each family by question format or answer category and distribute its samples across these subtypes. Specifically, we stratify visual mathematics by direct, fill-in-the-blank, and multiple-choice questions; CLEVR by answer type; A-OKVQA by whether the answer contains a rationale; ST-VQA by numeric versus non-numeric answers; TallyQA by count range; and binary or multiple-choice tasks by their class or option labels. When a subtype lacks enough examples, we retain all of its available samples without replacement and fill the remaining quota with other subtypes from the same family. These labels are used only for sampling; the original answer text is preserved as the supervision target.

A.2.3 Online Sample Decontamination

We perform decontamination online as part of the task-balanced sampling procedure described above. Whenever the sampler draws a record, we immediately compare it against the samples in POPE, BLINK, V*Bench, RealWorldQA, and CV-Bench and decide whether to retain or discard it. A discarded record does not count toward its task-family quota; instead, we draw a replacement using the same sampling rule. Invalid records and unreadable images are likewise skipped and replaced.

The matching pipeline proceeds from exact matching to candidate retrieval and pairwise verification. First, we compare the file SHA-256 and normalized-pixel SHA-256 of the sampled image with those of all benchmark images. A match from either hash causes the record to be discarded immediately. Records with no exact match proceed to candidate retrieval.

We narrow the subsequent comparison set using three retrieval signals. The first two are pHash and dHash, with maximum Hamming distances of 12 and 8, respectively. The third uses DINOv2 features (Oquab et al., 2024) from six image views: one full-image view, one 80% center crop, and four 70% corner crops. We retrieve the top 20 benchmark images for each view and take their union. If the normalized question text exactly matches a benchmark question, we also include its associated images and expand DINOv2 retrieval to the top 100. These signals only select benchmark candidates for further verification and do not directly cause a record to be discarded.

We first apply whole-image verification to each retrieved candidate. The images are resized to 256×256, and a match requires all three conditions: an aspect-ratio difference of at most 0.02, a color mean absolute error (MAE) of at most 0.055, and a grayscale Pearson correlation of at least 0.97. If all three conditions hold for any candidate, the sampled record is discarded.

Candidates that do not satisfy the whole-image conditions undergo local-feature and geometric verification. We extract up to 2,500 SIFT features and apply the Lowe ratio test (Lowe, 2004) after limiting the longest image side to 768 pixels. We use a contrast threshold of 0.02 and retain matches with a ratio below 0.75. RANSAC (Fischler & Bolles, 1981) estimates the geometric alignment with a 4-pixel reprojection threshold, at most 2,000 iterations, and confidence 0.995. A geometric match requires at least 10 inliers, an inlier ratio of at least 0.28, coverage of at least 0.025 in both images and at least 0.12 in one image, a median reprojection error of at most 3.5 pixels, and an aligned overlap of at least 0.035. For aligned-pixel verification, we erode the valid mask with a 5×5 kernel and require at least 256 valid pixels. The aligned region must further satisfy either a grayscale correlation of at least 0.90 with an MAE of at most 0.12, a correlation of at least 0.97, or an MAE of at most 0.055. If any retrieved candidate satisfies the complete geometric and aligned-pixel conditions, the sampled record is discarded; otherwise, it is retained in the training set.

A.3 Training Configurations

We describe the training configurations of the default MoLE model and the two data-matched baselines, Qwen2.5-VL-7B-SFT and LIVR. All three models use Qwen2.5-VL-7B as the backbone and are trained on the same 63K-example dataset. We first present their shared settings and then specify the differences in trainable parameters, objectives, and training schedules.

A.3.1 Shared Training Setup

For all three models, we keep the vision encoder and visual projector frozen and adapt the language backbone with LoRA. LoRA is applied to the attention projections `q_proj`, `k_proj`, `v_proj`, and `o_proj`, and to the MLP projections `gate_proj`, `up_proj`, and `down_proj`. We use rank 16, scaling factor 32, and dropout 0.05.

Each image is processed independently with a visual-token budget between 64 and 8192, corresponding to a pixel budget of $[64 \times 28 \times 28,\ 8192 \times 28 \times 28]$. For multi-image examples, this budget is applied separately to every image rather than shared across the sample or batch. We use a maximum sequence length of 9,216 tokens. If an answer exceeds the remaining sequence length, we truncate it while retaining the final end-of-sequence token.

Table 4: Per-image visual-token budgets and reported metrics for the five evaluation benchmarks.

Benchmark	Tokens/image	Reported metric
POPE	64–512	Unweighted mean F1 over the adversarial, popular, and random subsets
BLINK	64–512	Unweighted mean accuracy over the 14 tasks
V*Bench	256–4,096	Accuracy over all test examples
RealWorldQA	256–4,096	Accuracy over all test examples
CV-Bench	256–2,048	2D: mean of ADE20K and COCO accuracies; 3D: Omni3D accuracy

The shared optimization settings use AdamW (Loshchilov & Hutter, 2019) with batch size 8, learning rate 10^{-4}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, weight decay 0.01, and gradient clipping at 1.0. We use BF16 precision, gradient checkpointing, 5% linear warmup, and cosine learning-rate decay.

A.3.2 DEFAULT MOLE MODEL

The default configuration uses $E = 4$ latent visual experts, $S = 4$ latent summary experts, Router Top-$K = 2$, and an ESVA hidden dimension of $d_b = 32$. The router-prior scale is $\alpha = 1.0$, with $\varepsilon = 10^{-6}$ for numerical stability, and the Value RMSNorm uses $\varepsilon_{\text{norm}} = 10^{-6}$. The latent embeddings, router projections, and ESVA down- and up-projection matrices are initialized from $\mathcal{N}(0, 0.02^2)$. The Value RMSNorm weights are initialized to one, and $\gamma_{\ell,e}$ is initialized to 0.01. In addition to the LoRA parameters, we update the embedding rows of the $E + S$ newly introduced latent experts and all newly introduced MOLE parameters in both stages.

Stage 1 is trained for two epochs, followed by one epoch of Stage 2. Both stages use the same joint objective, with $\lambda_{\text{balance}} = 10^{-3}$, $\lambda_z = 10^{-3}$, and $\lambda_{\text{div}} = 10^{-2}$. The complete training pipeline takes approximately 20 hours on two PRO 6000D GPUs.

A.3.3 QWEN2.5-VL-7B-SFT

The data-matched SFT baseline uses the same LoRA configuration as the default MOLE model and introduces no latent parameters. It is trained for three epochs on the same dataset using only the answer-token cross-entropy loss. Its visual input processing and optimization settings are otherwise identical to those described above.

A.3.4 LIVR

LIVR uses the same backbone and LoRA configuration as the default MOLE model, together with a fixed set of learnable latent embeddings. We train separate variants with 8 and 16 latent tokens. Each variant follows the same two-stage schedule as MOLE, with two epochs in Stage 1 and one epoch in Stage 2, but is optimized using only the answer-token cross-entropy loss. It uses the same training data, visual input processing, and optimization settings as MOLE.

A.4 EVALUATION CONFIGURATIONS

We use the same benchmark-specific evaluation settings for all locally evaluated models. Table 4 reports the visual-token budget and metric used for each benchmark. The visual budget is applied independently to every image; in particular, each image in a multi-image BLINK example receives the full per-image budget shown in the table.

All five benchmarks use the same Qwen chat template. The system message is fixed to "You are a helpful assistant." The user message contains the image item(s), in their original order, followed by the benchmark-specific question and response-format instruction.

Table 5: CVBench-related scores for GPT-4o and Claude-4-Sonnet as reported by CoVT.

Model	CVBench	Count	Depth	Dist.
GPT-4o	79.2	65.6	86.7	81.0
Claude-4-Sonnet	76.3	62.2	77.7	80.5

A.5 CVBench Score Calculation

CoVT (Qin et al., 2026) reports four CVBench-related scores for GPT-4o and Claude-4-Sonnet: CVBench, Count, Depth, and Dist. CVBench is the aggregate score over the full benchmark and reflects both 2D and 3D visual spatial understanding. Count corresponds to Object Counting, which asks how many instances of a specified object category appear in an image and evaluates instance recognition, object separation, and counting. Depth corresponds to Depth Order, which asks which of two objects is closer to the camera and evaluates depth perception, front–back ordering, and 3D geometry. Dist. corresponds to Relative Distance, which asks which object is closer to a specified reference object and evaluates relative 3D distance. Table 5 reproduces the scores reported by CoVT.

CV-Bench comprises four tasks: its 2D subset contains Spatial Relationship and Object Counting, whereas its 3D subset contains Depth Order and Relative Distance. We therefore compute the 3D accuracy by averaging the reported Depth and Dist. scores. Let A_{all}, A_{depth}, and A_{dist} denote the reported CVBench, Depth, and Dist. scores, respectively. The 3D accuracy is

$$A_{\text{3D}} = \frac{A_{\text{depth}} + A_{\text{dist}}}{2}. \tag{18}$$

Because the aggregate CVBench score is the mean of its 2D and 3D scores, we recover the 2D accuracy as

$$A_{\text{2D}} = 2A_{\text{all}} - A_{\text{3D}}. \tag{19}$$

For GPT-4o, these calculations give $A_{\text{3D}} = (86.7{+}81.0)/2 = 83.85$ and $A_{\text{2D}} = 2{\times}79.2{-}83.85 = 74.55$. For Claude-4-Sonnet, they give $A_{\text{3D}} = (77.7{+}80.5)/2 = 79.1$ and $A_{\text{2D}} = 2{\times}76.3{-}79.1 = 73.5$. After rounding to one decimal place, we report $(74.6, 83.9)$ for GPT-4o and $(73.5, 79.1)$ for Claude-4-Sonnet as their respective 2D and 3D results in Table 1.

A.6 Latent Representation Metrics

We quantify latent complementarity from two perspectives: whether the latent hidden states encode similar representations and whether the latents attend to the same visual evidence. Both metrics are computed per example, averaged over the examples in each benchmark, and then averaged equally across the five benchmarks.

A.6.1 Latent-State Similarity

Let L be the number of decoder layers and M the number of latent tokens. For an example x, let $\mathbf{h}_{x,i}^{(\ell)}$ denote the hidden state of latent i produced by decoder layer ℓ, and define its ℓ_2-normalized representation as

$$\widehat{\mathbf{h}}_{x,i}^{(\ell)} = \frac{\mathbf{h}_{x,i}^{(\ell)}}{\|\mathbf{h}_{x,i}^{(\ell)}\|_2}. \tag{20}$$

We compute the mean pairwise squared normalized inner product among all latent tokens at each layer:

$$s_x^{(\ell)} = \frac{2}{M(M-1)} \sum_{1 \le i < j \le M} \left(\widehat{\mathbf{h}}_{x,i}^{(\ell)\top} \widehat{\mathbf{h}}_{x,j}^{(\ell)} \right)^2. \tag{21}$$

The example-level similarity is the mean over all decoder layers:

$$\mathcal{S}_{\text{latent}}(x) = \frac{1}{L} \sum_{\ell=1}^{L} s_x^{(\ell)}. \tag{22}$$

This score lies in $[0, 1]$; a lower value indicates less redundant latent representations.

A.6.2 LATENT ATTENTION-PATTERN DIVERSITY

This metric measures whether different latents distribute their attention over different visual tokens. In a conventional latent visual reasoning method, all M latent tokens can attend to the visual input and are included in this calculation.

Suppose an example contains V visual tokens and each decoder layer has H attention heads. Let $a_{x,\ell,i,h,v}$ be the attention probability assigned by latent i at layer ℓ and head h to visual token v. We first average over attention heads:

$$\bar{a}_{x,\ell,i,v} = \frac{1}{H} \sum_{h=1}^{H} a_{x,\ell,i,h,v}. \tag{23}$$

We then accumulate the resulting visual attention across decoder layers:

$$A_{x,i,v} = \sum_{\ell=1}^{L} \bar{a}_{x,\ell,i,v}. \tag{24}$$

We add $\varepsilon = 10^{-8}$ for numerical stability and normalize over the visual-token positions to obtain an attention distribution for each latent:

$$p_{x,i,v} = \frac{A_{x,i,v} + \varepsilon}{\sum_{u=1}^{V} A_{x,i,u} + V\varepsilon}. \tag{25}$$

For two latents i and j, we denote their directed attention-pattern KL divergence by $d_{x,i\to j}$ and compute it using the natural logarithm:

$$d_{x,i\to j} = \sum_{v=1}^{V} p_{x,i,v} \log\left(\frac{p_{x,i,v}}{p_{x,j,v}}\right). \tag{26}$$

The example-level diversity is the mean over all ordered pairs of distinct latents, so both directions of each pair are included:

$$\mathcal{D}_{\text{attn}}(x) = \frac{1}{M(M-1)} \sum_{\substack{i,j=1 \\ i\neq j}}^{M} d_{x,i\to j}. \tag{27}$$

A larger value indicates that the latents attend to visual evidence in more distinct patterns. Finally, for either example-level metric m, let $\mathcal{X}_b$ denote the examples from benchmark b and $\mathcal{B}$ the set of five benchmarks. We aggregate and report the metric as

$$\bar{m}_b = \frac{1}{|\mathcal{X}_b|} \sum_{x\in\mathcal{X}_b} m(x). \tag{28}$$

We then average the benchmark-level values and apply the reporting scale:

$$m_{\text{report}} = \frac{100}{|\mathcal{B}|} \sum_{b\in\mathcal{B}} \bar{m}_b. \tag{29}$$

Thus, each benchmark contributes equally to the final value, and the plotted scale is 100 times the corresponding raw metric.

For conventional latent visual reasoning methods, all M latent tokens are used for both metrics. MoLE instead contains E latent visual experts and S latent summary experts. We compute latent-state similarity over all $M = E + S$ latent tokens, since both sets form intermediate latent representations. For attention-pattern diversity, we use only the E latent visual experts (i.e., set $M = E$ in the corresponding equations), because the summary experts aggregate the visual experts without directly attending to the original visual tokens.

A.7 MORE ABLATIONS

A.7.1 PARAMETER-MATCHED SHARED ADAPTER

ESVAs introduce additional parameters into the model's forward computation. To determine whether the resulting performance gains come merely from increased model capacity, we construct

Table 6: Comparison with a parameter-matched shared-adapter baseline.

Method	CVBench 2D	CVBench 3D	BLINK	RWQA	V*	POPE	Avg.
Qwen2.5-VL-7B	<u>75.4</u>	73.3	55.6	<u>68.9</u>	<u>76.6</u>	85.0	72.5
Parameter-matched adapter	74.9	<u>75.4</u>	<u>56.5</u>	66.7	76.3	<u>86.8</u>	<u>72.8</u>
MoLE	**78.2**	**87.8**	**63.0**	**71.5**	**80.6**	**90.2**	**78.6**

POPE reports F1; all other metrics report accuracy. Avg. averages the six metrics. Bold and underlined values are the best and second best, respectively.

Table 7: Performance of the default MoLE model across three random seeds. Mean and sample standard deviation are computed over the three runs.

Model	CVBench 2D	CVBench 3D	BLINK	RWQA	V*	POPE	Avg.
Qwen2.5-VL-7B	75.4	73.3	55.6	68.9	76.6	85.0	72.5
MoLE (seed 42)	78.2	87.8	63.0	71.5	80.6	90.2	78.6
MoLE (seed 45)	77.5	87.1	61.3	72.0	79.4	89.9	77.9
MoLE (seed 48)	78.3	86.8	62.1	71.2	79.8	89.9	78.0
Mean $\pm$ *std.*	78.0$\pm$0.4	87.2$\pm$0.5	62.1$\pm$0.9	71.6$\pm$0.4	79.9$\pm$0.6	90.0$\pm$0.2	78.2$\pm$0.4

POPE reports F1; all other metrics report accuracy. Avg. averages the six metrics.

a parameter-matched control based on Qwen2.5-VL-7B. Specifically, we add a shared value adapter after the original value projection in every attention layer and choose its size so that the resulting model has a comparable total parameter count to the default MoLE model. Unlike ESVAs, these adapters are not expert-specific and therefore do not specialize the transformation of visual evidence. We fine-tune this model on the same 63K training set used for MoLE and evaluate it under the same benchmark configurations.

As shown in Table 6, adding a comparable number of parameters improves the average performance of the backbone by only 0.3, whereas MoLE outperforms the parameter-matched model by 5.8 and performs better on every reported metric. This result indicates that the gains of MoLE cannot be explained by parameter count alone and instead support the importance of explicitly specializing latent computation.

A.7.2 Stability Across Random Seeds

To assess whether the gains of MoLE are stable with respect to training randomness, we train the default configuration independently with random seeds 42, 45, and 48. All runs use the same training data and model configuration and are evaluated under the same benchmark settings.

Table 7 shows that the average performance remains between 77.9 and 78.6 across the three runs, with a mean of 78.2 and a standard deviation of 0.4. Every run outperforms the Qwen2.5-VL-7B backbone on all six metrics, and even the lowest average exceeds the backbone by 5.4. These results indicate that the performance gains of MoLE are consistent across the evaluated random seeds.

A.7.3 Backbone and Scale Extension

We further examine whether MoLE transfers to a different backbone family and remains effective across model scales. We use Qwen3-VL (Bai et al., 2025a), whose publicly available checkpoints provide finer-grained model sizes than Qwen2.5-VL, and apply MoLE to the 2B, 4B, and 8B variants. For each scale, we independently train three models with random seeds 42, 45, and 48 on the same 63K training set and evaluate them under the same benchmark configurations.

As shown in Table 8, MoLE raises the mean average score over the corresponding Qwen3-VL backbone by 2.7, 1.3, and 1.5 at the 2B, 4B, and 8B scales, respectively. Every run improves over

Table 8: Backbone and scale extension on Qwen3-VL. Mean and sample standard deviation are computed over three MoLE runs at each model scale.

(a) Qwen3-VL-2B

	CVBench						
Model	2D	3D	BLINK	RWQA	V*	POPE	Avg.
Qwen3-VL	72.8	86.6	45.7	65.6	75.4	89.3	72.6
MoLE (42)	74.7	88.3	55.5	66.3	78.0	90.2	75.5
MoLE (45)	73.4	87.9	55.5	66.0	77.4	89.9	75.0
MoLE (48)	73.9	88.4	55.7	66.7	77.8	90.0	75.4
Mean $\pm$ *std.*	74.0$\pm$0.7	88.2$\pm$0.3	55.6$\pm$0.1	66.3$\pm$0.4	77.7$\pm$0.3	90.0$\pm$0.2	75.3$\pm$0.3

(b) Qwen3-VL-4B

	CVBench						
Model	2D	3D	BLINK	RWQA	V*	POPE	Avg.
Qwen3-VL	79.1	92.4	62.2	70.3	81.2	89.3	79.1
MoLE (42)	79.9	93.3	63.7	71.8	84.8	90.7	80.7
MoLE (45)	79.6	92.6	63.2	71.2	84.1	90.1	80.1
MoLE (48)	80.2	93.1	62.8	71.6	84.6	90.3	80.4
Mean $\pm$ *std.*	79.9$\pm$0.3	93.0$\pm$0.4	63.2$\pm$0.5	71.5$\pm$0.3	84.5$\pm$0.4	90.4$\pm$0.3	80.4$\pm$0.3

(c) Qwen3-VL-8B

	CVBench						
Model	2D	3D	BLINK	RWQA	V*	POPE	Avg.
Qwen3-VL	80.6	93.2	64.3	69.5	82.7	88.2	79.8
MoLE (42)	81.3	93.7	67.1	71.8	85.3	89.9	81.5
MoLE (45)	80.8	93.4	66.4	71.5	84.7	89.7	81.1
MoLE (48)	80.8	93.7	66.5	72.3	84.9	90.2	81.4
Mean $\pm$ *std.*	81.0$\pm$0.3	93.6$\pm$0.2	66.7$\pm$0.4	71.9$\pm$0.4	85.0$\pm$0.3	89.9$\pm$0.3	81.3$\pm$0.2

Parentheses after MoLE give the random seed. POPE reports F1; all other metrics report accuracy. Avg. averages the six metrics.

Table 9: Ablations of (a) routing density and (b) ESVA hidden dimension. Only average performance is reported.

(a) Routing density

Router Top-K	Avg.
$K = 1$	77.8
$K = 2$	**78.6**
$K = 3$	77.4
$K = 4$	<u>77.9</u>

(b) ESVA hidden dimension

Hidden dimension d_b	Avg.
16	77.2
32	**78.6**
64	77.0
128	<u>77.4</u>

Avg. averages the six metrics. The default uses $K = 2$ and $d_b = 32$. Bold and underlined values are the best and second best within each ablation, respectively.

Table 10: Average inference cost on RWQA.

Model	Time (s)	FLOPs	Peak memory (GiB)
Qwen2.5-VL-7B	0.374478	3.481782×10^{13}	15.836116
MoLE	0.460499	3.510919×10^{13}	16.081371
Relative overhead	$+22.97\%$	$+0.84\%$	$+1.55\%$

its corresponding backbone on all six metrics, while the standard deviation of the average score is at most 0.3 at each scale. These results show that the gains remain consistent across random seeds and extend beyond the default Qwen2.5-VL-7B backbone to multiple Qwen3-VL model sizes.

A.7.4 Routing Density

Router Top-K controls how many latent visual experts receive each visual token at a decoder layer. Starting from the default MoLE configuration, we independently retrain models with $K \in \{1, 2, 3, 4\}$ to examine how the density of expert–visual-token connections affects performance. As shown in Table 9(a), the default $K = 2$ achieves the best average performance. Routing each visual token to only one expert may be overly restrictive, whereas larger values of K increase the overlap in visual evidence received by different experts without providing further gains. This suggests that moderate routing density balances selective evidence allocation with sufficient expert processing.

A.7.5 ESVA Hidden Dimension

We vary the hidden dimension d_b of the two-layer ESVA MLP over $\{16, 32, 64, 128\}$ while keeping the remaining default configuration unchanged. This ablation tests whether the benefit of ESVA depends on increasing its transformation capacity. Table 9(b) shows that $d_b = 32$ performs best, while larger hidden dimensions do not improve performance. Thus, the ESVA gains do not arise from simply increasing adapter capacity, and a compact hidden dimension is sufficient for expert-specific visual transformation.

A.7.6 Inference Cost

We quantify the inference overhead of MoLE relative to its Qwen2.5-VL-7B backbone. For every RWQA example, we evaluate both models and record inference time, floating-point operations (FLOPs), and peak GPU memory, then average each measurement over the full benchmark. Table 10 reports the results.

MoLE adds only 0.84% FLOPs and 1.55% peak memory, corresponding to 0.245 GiB. Its measured inference time increases by 22.97%, but the absolute increase is 0.086 seconds per example and the average latency remains below 0.5 seconds. In return, MoLE improves the average benchmark score over the backbone by 6.1 points. We therefore consider this inference overhead acceptable for the resulting performance gain.