

Look and Think: Efficient Multimodal Reasoning via Modality-Decoupled Compression

Xidi Cai, Junhao Zheng, Jingye Li, Boyuan Li, Shaowei Zhang, Qianli Ma*,

School of Computer Science and Engineering, South China University of Technology

xidicai067@gmail.com, qianlima@scut.edu.cn

Abstract

Multimodal large language models (MLLMs) have achieved strong performance on challenging visual question answering benchmarks, yet their inference efficiency is severely constrained by the rapidly growing context. This growth stems from two primary sources: the large number of visual tokens required to encode images, and the accumulation of intermediate reasoning traces during autoregressive generation. To address these challenges, we propose LaT (Look and Think), the first modality-decoupled compression method that enables efficient multimodal inference. LaT structures reasoning into alternating *looking* and *thinking* steps, thereby explicitly signaling when visual grounding is required. Building on this design, LaT (1) evicts visual tokens whenever visual grounding is unnecessary, and (2) applies co-learning-guided compression after each completed step, mitigating the two sources of context growth respectively. Experimental results demonstrate that LaT reduces the average context length by up to 57%, while maintaining performance comparable to the standard MLLM.

1 Introduction

Multimodal Large Language Models (MLLMs) have demonstrated remarkable capabilities on challenging visual question answering (VQA) tasks. A key factor behind this progress is the introduction of any-resolution visual encoding techniques (Liu et al., 2024; Wang et al., 2024), which allow MLLMs to flexibly process images at varying resolutions and preserve fine-grained visual details without being constrained by fixed input sizes. In parallel, techniques like Chain-of-Thought (CoT) prompting (Wei et al., 2022) boosts their performance by decomposing problems into intermediate steps. Building on this paradigm, recent o1-like thinking models (Jaech et al., 2024; Yang et al.,

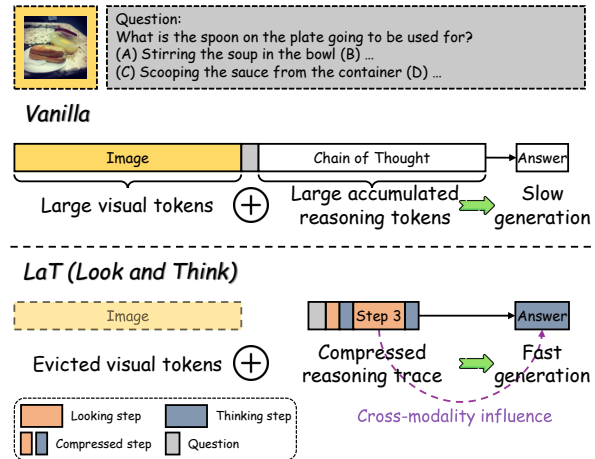


Figure 1: LaT (Look and Think) vs. vanilla MLLM inference. A vanilla MLLM keeps large visual tokens and the full accumulated reasoning trace in context, leading to slow generation. LaT decomposes inference into alternating *looking* (i.e., image-grounded observation) and *thinking* (i.e., text-only reasoning) steps: it evicts visual tokens during thinking steps and compresses completed steps into compact representations, enabling faster generation. Also, by keeping the most recent complementary step uncompressed, LaT retains fine-grained cross-modal evidence for subsequent reasoning.

2025a; Guo et al., 2025) move beyond prompting by explicitly training multi-step reasoning, leading to stronger performance.

However, as MLLM capabilities continue to advance, inference efficiency has emerged as a critical bottleneck. Owing to the transformer-based architecture underlying most MLLMs, the computational cost of the attention mechanism scales quadratically with the context length (Vaswani et al., 2017). During multimodal reasoning, the context often becomes especially long, resulting in substantial computational overhead.

In particular, the extended context in MLLMs stems from two primary sources. First, any-resolution visual encoding introduces a large num-

*Corresponding author.

ber of visual tokens to represent high-resolution images. Although these tokens are essential for preserving fine-grained visual details, they often account for the majority of tokens in the prompt (Zhang et al., 2024a). Second, advanced reasoning paradigms, such as CoT prompting, encourage the generation of lengthy intermediate reasoning traces (Chen et al., 2024e). As autoregressive decoding proceeds, these intermediate steps accumulate in the context, gradually becoming a significant contributor to context growth and further exacerbating inference inefficiency.

Existing approaches proposed to improve the inference efficiency can be broadly categorized into two directions: eviction and compression. Eviction-based methods identify low-importance visual tokens and remove them from the context, retaining only curtail visual information to reduce computation (Chen et al., 2024b; Zhang et al., 2024c). Compression-based methods instead summarize large context into a smaller representation, including visual token compression that adapts the model to reduced visual context (Wen et al., 2024), as well as reasoning compression that condenses earlier steps before further generation (Zhang et al., 2025b; Pang et al., 2024).

Despite advancements in both approaches, prior work largely addresses visual token inefficiency and reasoning trace accumulation in isolation. This separation overlooks a key characteristic of multimodal inference: visual and linguistic information interact dynamically throughout reasoning (Zhang et al., 2025f). During reasoning, visual cues continually refine the reasoning trace, and the reasoning trace guides subsequent visual attention. Optimizing visual tokens or reasoning traces in isolation therefore fails to capture this interdependence, limiting the overall effectiveness of efficiency improvements.

To improve the inference efficiency of MLLMs while still encouraging cross-modality influence, we propose LaT, the first modality-decoupled compression method. LaT contains two components, (1) visual token eviction and (2) co-learning-guided compression. Specially, LaT structures multimodal reasoning into alternating *looking* and *thinking* steps (Figure 1). During thinking steps, LaT evicts visual tokens from the context, and subsequently reintroduces them when generating looking steps. To further mitigate inefficiency arising from reasoning traces, LaT compresses each completed step into compact representations, while preserving fine-

grained evidence from the complementary modality to facilitate mutual influence.

We evaluate the effectiveness of LaT using the average sample-level context length, defined as the number of key-value (KV) caches participating in attention computation when generating each new token during inference. Experiments on M³CoT, a challenging and comprehensive multimodal reasoning benchmark, show that LaT reduces the context length by up to **57%**, while maintaining performance comparable to standard MLLM inference without eviction or compression. Our main contributions are as follows: (1) We identify two primary sources of context growths: the large number of visual tokens and the accumulation of reasoning traces. (2) We propose LaT, the first modality-decoupled compression method prompting cross-modality influence. (3) We demonstrate that LaT can effectively reduce context growth while maintaining strong task performance.

2 Related Work

Inefficiency in MLLMs mainly comes from two sources: (1) the heavy visual-token budget and (2) the long CoT traces. To reduce the redundant visual tokens, prior work typically either evicts low-importance visual tokens (Chen et al., 2024b; Zhang et al., 2024c) or compresses multiple visual tokens into a small set of compact representations (Li et al., 2024; Wen et al., 2024). To speed up reasoning, existing methods explore both inducing models to generate shorter reasoning traces (Han et al., 2025; Wang et al., 2025) and reducing generated traces online before further generation (Ramachandran et al., 2025; Zhang et al., 2025b). Due to space constraints, we provide a detailed discussion of related work in Appendix A.

3 Method

In this section, we provide a comprehensive description of LaT. We first introduce our visual dependency-aware data construction procedure, which produces step-wise rationales with explicit looking and thinking steps. We then introduce LaT’s dynamic visual token eviction mechanism, which alternates between evicting visual tokens during thinking steps and reintroducing them for looking steps as the reasoning proceeds. Next, we present LaT’s modality-interleaved co-learning design that promotes cross-modality influence. Finally, we detail the training objective used to opti-

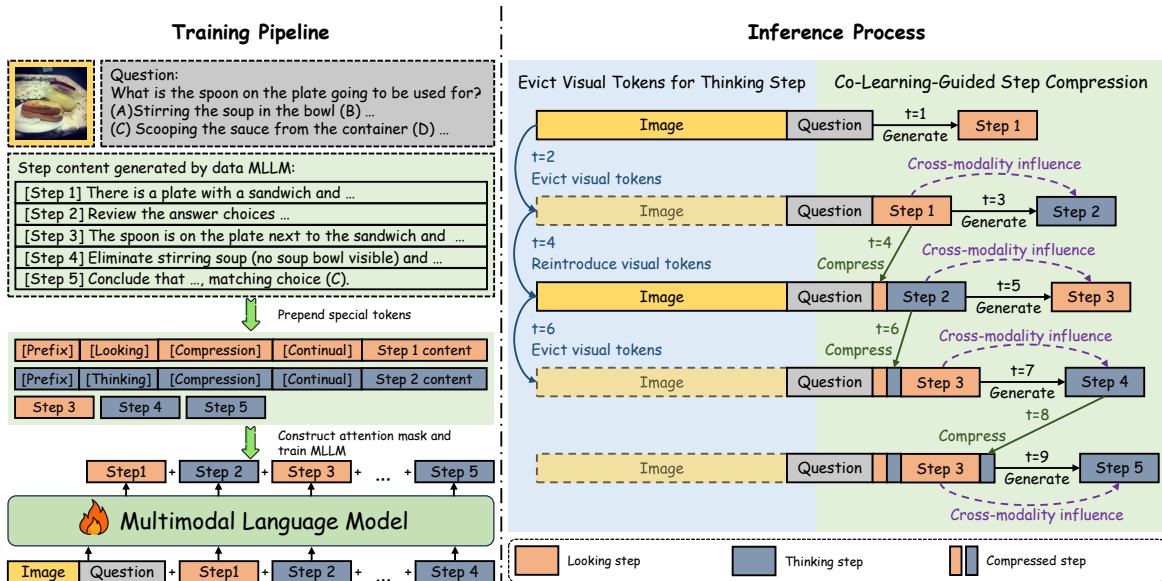


Figure 2: **Overview of LaT (Look-and-Think).** **Left: Training.** An instruction-following data MLLM generates step-wise rationales that alternate between *looking* and *thinking* steps. Each step is formatted with special tokens. **Right: Inference.** LaT evicts visual tokens for thinking steps and reintroduces them for looking steps, while compressing each completed step into a compact representation. A co-learning-guided compression strategy keeps the most recent complementary step uncompressed, preserving fine-grained cross-modal evidence and maintaining cross-modality influence. The complete rationale is provided in Figure 7.

mize LaT.

3.1 Modality-Decoupled Dataset Construction

To decouple the modalities involved in MLLM reasoning, we design two types of steps: *looking steps* and *thinking steps*. Looking steps refer to the steps requiring understanding the image, such as identifying objects or the spatial relationships, while the thinking steps refer to the steps involving reasoning based on the question and previous generated reasoning trace, such as comparing options and drawing conclusions. Each rationale is organized as a sequence of steps, with every step explicitly categorized as either a looking step or a thinking step.

We employ a data MLLM and an OCR MLLM to construct the training rationale. The data MLLM is instantiated as a strong instruction-following MLLM with robust multimodal reasoning capabilities (e.g., GPT-5 mini). Given the image, question, and ground-truth answer, the data MLLM is directly instructed to generate a step-by-step rationale. Before generating each step, the data MLLM is required to explicitly declare the step type, ensuring that each step is unambiguously associated with its underlying modality dependency.

During data construction, we observe that the data MLLM can occasionally struggle to distin-

guish whether or not the information it relies on originates from the visual modality when processing text-rich images. In such cases, image-derived text may be directly incorporated into thinking steps, weakening the intended separation of modalities.

To address this challenge, we introduce an auxiliary OCR MLLM to handle text-rich images. The OCR MLLM first assesses whether an input image contains substantial textual content. For images identified as text-rich, it extracts the embedded text and inserts it as the first step of the rationale, which is manually labeled as a looking step. This design enforces a clear separation between visual content acquisition and subsequent reasoning, thereby providing more reliable modality-aligned supervision for downstream training.

3.1.1 Special-Token Schema for Step-Wise Rationales

To facilitate modality-decoupled compression, we design five special tokens that explicitly organize the reasoning process (Jin et al., 2025a), enabling precise control over when different modalities are accessed and how contextual information is compressed.

- [Looking]: This token indicates that the subsequent step is looking step and requires vi-

sual information from the image.

- [Thinking]: This token indicates that the subsequent step is thinking step without requiring access to visual information.
- [Prefix]: Placed before [Looking] and [Thinking], this token augments the model’s computational capacity before step-type prediction by providing a type-agnostic planning space, allowing the model to make a more informed decision (Wang et al., 2023).
- [Compression]: Placed after [Looking] and [Thinking], this token compress the previous step content into its hidden state.
- [Continual]: Placed after [Compression], this token indicates potential context changes, guiding the model to rely on the revised context for further inference (Zhang et al., 2025b).

For each input x in the training dataset, we use the data MLLM and the OCR MLLM to construct its step-wise rationale $\{s_1, s_2, \dots, s_L\}$, where L is the number of steps. For each step, we prepend a sequence of special tokens as follows:

$$s'_i = ([\text{Prefix}], [\text{StepType}], [\text{Compression}], [\text{Continual}], s_i),$$

where i is the step index and $[\text{StepType}] \in \{[\text{Looking}], [\text{Thinking}]\}$. The output y corresponding to x is then formulated as:

$$y = (s'_1, s'_2, \dots, s'_L).$$

It is important to note that [Compression] in s'_i is intended to compress previous step, s_{i-1} . As a result, [Compression] in s'_1 is unnecessary. However, to maintain consistency in the special token sequence pattern, we chose not to remove [Compression] from s'_1 .

We also note that for [Looking], [Thinking], [Prefix], and [Compression], each special token can consist of a segment made up of multiple tokens. For [Prefix] and [Compression], this design offers flexibility by allowing for varying amounts of computational space and control over the difficulty of context compression. For [Looking] and [Thinking], multi-token segments provide a clearer modality-decoupling signal than a single delimiter. Their longer patterns are easier for the model to recognize and use as conditioning cues. In practice, these two segments act as

step-type-specific soft prompts that promote specialization: [Looking] biases generation toward visual grounding when visual tokens are present, while [Thinking] biases it toward textual reasoning when they are evicted.

During training, for each special token segment, the model only needs to predict the first token. During inference, when the model generates the first token of a special token segment, we incorporate the full segment in the subsequent forward pass. To simplify the discussion, we will still refer to each special token segment as a single token.

3.2 Dynamic Visual Token Eviction

Through dataset construction, we partition the reasoning process into steps according to visual dependency. For steps that do not rely on the visual modality (i.e., thinking steps), visual tokens can be excluded from the context to improve efficiency. In contrast, for steps that depend on visual information (i.e., looking steps), visual tokens should be retained in the context to support accurate visual grounding and evidence-based reasoning.

To train the model to condition its reasoning on visual tokens only when necessary, we simulate visual token eviction during training via attention masks (Figure 3). Specifically, we only change the context at [Continual]. When the upcoming step is a thinking step, we mask out all visual tokens and keep them masked until the [Continual] token of the next looking step. At inference time, we actually remove the KV cache of visual tokens for thinking steps. Importantly, LaT does not constrain the model to alternate between looking and thinking steps. When multiple consecutive steps share the same type, visual-token eviction or reintroduction is triggered only at actual step-type transitions.

3.3 Modality-Interleaved Co-Learning

LaT compresses the previous step into the the hidden state of [Compression] in the current step. However, unlike previous works, we do not always drop the hidden state of the compressed content immediately after it is compressed. Although our modality-decoupled rationales decouple visual perception and textual reasoning, different modalities still interact tightly during inference: language-based reasoning directs attention toward the most relevant cues in the image, while visually grounded evidence feeds back to inform subsequent reasoning. A straightforward compress-and-drop strategy can weaken cross-modal interactions.

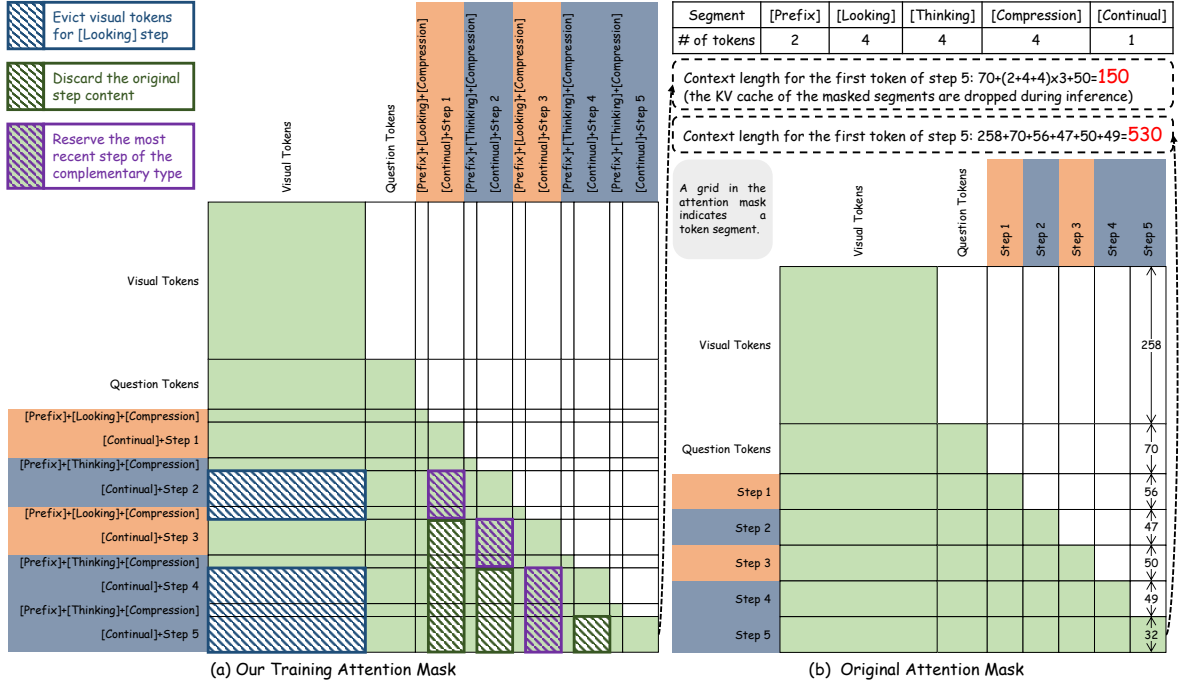


Figure 3: Comparison between our training attention mask and the original.

Building on this insight, we propose our co-learning-guided compression strategy: at any time, we keep the most recent step of the complementary type uncompressed. When the current step has the same type as the previous step, we retain only the compressed representation of the previous step. Otherwise, we retain the previous step uncompressed as the reserved complementary step and drop the content of the earlier reserved step. This guarantees that each step can still attend to the latest fine-grained evidence from the other modality. Following LightThinker (Zhang et al., 2025b), we perform compression immediately after each step is generated to ensure high-quality compression. We also allow the model to access the input and the compressed previous step during compression to further improve quality.

Similar to Section 3.2, we use attention masks during training to help the model adapt to compressed context. Concretely, we update the reserved step at each [Continual] token and mask out the contents of all other steps for the remainder of generation. During inference, we permanently remove the corresponding KV caches of dropped content from the context.

3.4 Training Objective

The training objective of LaT follows standard supervised fine-tuning, except that we apply our custom attention mask constructed as described in Sec-

tion 3.2 and Section 3.3. Formally, we minimize the negative log-likelihood:

$$\mathcal{L}_{\text{LaT}}(\theta) = -\mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\sum_{t=1}^{|y|} \log p_{\theta}(y_t \mid y_{<t}, x; \mathbf{M}(x, y)) \right],$$

where $\mathbf{M}(x, y)$ denotes the constructed attention mask.

4 Experiment

4.1 Experiment Setup

Models. We use Qwen3-VL-2B-Instruct and Qwen3-VL-2B-Thinking (Bai et al., 2025) as backbone models for training and test. GPT-5 Mini (OpenAI, 2025) serves as the data MLLM and OCR MLLM mentioned in Section 3.1.

Datasets. During training, we use questions from the M³CoT (Chen et al., 2024d) training split to construct modality-decoupled rationales. We then evaluate LaT on the M³CoT test split, and further assess LaT’s out-of-domain generalization in Section 4.3.4. Full details of the training and evaluation datasets are provided in Appendix B.1.

Baselines. In our experiments, we adopt Chain-of-Thought (CoT) prompting and supervised fine-tuning as efficiency-agnostic baselines. As an

Table 1: Main Results. Efficiency methods are recorded in black text. Bold indicates the best performance, and underlined indicates the second-best. For Qwen3-VL-2B-Thinking, we find that in the Mathematics domain, the model fails to complete the reasoning process in most cases, even when it is allowed to generate twice as many tokens as in other settings. Two ablated variants are denoted as Eviction, which applies only dynamic visual token eviction, and Rolling, which applies only the co-learning-guided compression strategy.

Method	Commonsense			Mathematics			Science			Overall		
	Acc ↑	Ctx Len ↓	Time ↓	Acc ↑	Ctx Len ↓	Time ↓	Acc ↑	Ctx Len ↓	Time ↓	Acc ↑	Ctx Len ↓	Time ↓
Qwen3-VL-2B-Instruct												
CoT	0.60	101K	2.13	0.20	1084K	8.28	0.75	211K	3.31	0.67	277K	3.59
SFT	0.62	105k	2.25	0.38	555k	5.15	0.85	133K	2.80	0.76	170K	2.93
Eviction	0.61	<u>68K</u>	2.00	0.23	<u>260K</u>	4.65	0.83	<u>73K</u>	<u>2.40</u>	0.72	<u>91K</u>	<u>2.55</u>
LightThinker	0.57	87K	<u>1.92</u>	0.28	976K	7.38	<u>0.84</u>	109K	2.38	0.73	193K	2.79
Rolling	<u>0.61</u>	162K	2.45	0.38	447K	<u>4.54</u>	0.84	123K	2.53	0.75	163K	2.72
LaT	0.61	54K	1.88	<u>0.35</u>	98K	3.84	0.84	65K	2.43	<u>0.74</u>	66K	2.46
Qwen3-VL-2B-Thinking												
CoT	0.61	585K	7.25	–	–	–	0.64	1236K	12.05	0.57	1201K	11.38
SFT	0.68	118K	2.49	0.33	738K	6.19	0.86	141K	2.95	0.77	197K	3.19
Eviction	0.58	<u>73K</u>	<u>2.04</u>	0.28	<u>198K</u>	3.79	0.83	<u>72K</u>	<u>2.39</u>	0.73	<u>84K</u>	2.47
LightThinker	0.58	99K	2.11	0.30	761K	6.25	0.82	108K	2.32	0.72	172K	2.68
Rolling	<u>0.61</u>	111K	2.27	0.33	469K	<u>4.66</u>	<u>0.86</u>	118K	2.51	<u>0.76</u>	152K	2.68
LaT	0.66	52K	1.80	<u>0.30</u>	132K	4.85	0.87	62K	2.52	0.77	67K	<u>2.62</u>

efficiency-oriented baseline, we include LightThinker (Zhang et al., 2025b), which compresses each reasoning step into a compact representation and also relies on supervised fine-tuning. In addition, we implement two ablated variants of LaT: one that applies only dynamic visual token eviction (*Eviction*), and another that incorporates only the co-learning-guided compression strategy (*Rolling*).

Evaluation Metric. For performance evaluation, we measure MLLMs using accuracy (*Acc*). To assess efficiency, we report the sample-level context length (*Ctx Len*) and inference time (*Time*). Specifically, *Ctx Len* is computed by summing the number of tokens in the context window at each forward pass during inference (Zhang et al., 2025b).

4.2 Main Result

We record the experiment result in Table 1. Our key observations are as follows.

Visual Token Eviction Substantially Reduces Context Length. Visual token eviction reduces the context length from 170K to 91K for Qwen3-VL-2B-Instruct and from 197K to 84K for Qwen3-VL-2B-Thinking. The most pronounced reductions occur in the Mathematics domain, where multimodal problems typically involve high-resolution, text-heavy images. In such cases, MLLMs can extract nearly all relevant visual information through a single OCR-like processing step (i.e., a looking step), while the subsequent solution process con-

sists primarily of lengthy textual reasoning steps that do not require continued access to visual tokens (i.e., thinking steps).

The Co-Learning-Guided Compression Strategy Preserves Model Performance. Across both versions of Qwen3-VL-2B, our compression strategy incurs only about a 1% drop in overall accuracy. In contrast, although LightThinker achieves shorter contexts in the Commonsense and Science domains, it suffers accuracy degradations of up to 10%. In the Mathematics domain, LightThinker often generates longer rationale that even exceed the maximum generation length, ultimately increasing the context length. This validates that our compression strategy effectively preserves critical cross-modality information, achieving more accurate reasoning under compressed contexts.

The Combination of Visual Token Eviction and Step Compression Leads to Better Performance and Efficiency. We observe that applying our compression strategy on top of visual token eviction further shortens the context length while consistently improving model performance. We believe this is because jointly applying the two techniques further strengthens the model’s ability to decouple the modality dependency in the reasoning process, allowing it to more accurately evict visual tokens and better benefit from our compression rule.

Table 2: Proportions of errors attributed to looking and thinking steps for Qwen3-VL-2B-Instruct across different domains in M³CoT.

Domain	Error Source	Proportion
Commonsense	Looking	0.29
	Thinking	0.71
Mathematics	Looking	0.07
	Thinking	0.92
Science	Looking	0.22
	Thinking	0.78

4.3 Further Analysis

4.3.1 Efficiency Analysis

We investigate the sources of efficiency gains achieved by LaT. As shown in Figure 4(a), our method substantially reduces the number of generated tokens compared to the SFT baseline, which directly contributes to faster inference. For Qwen3-VL-2B-Thinking, although LaT produces slightly more tokens than LightThinker and the rolling baseline, it still achieves lower inference latency. This indicates that the efficiency improvements stem not only from reduced generation length but also from more effective context compression. As illustrated in Figure 4(b), LaT is the only approach that retains less than 50% of the original context tokens. As a result, the computational cost of attention is reduced to approximately one quarter of standard MLLM inference, leading to significant efficiency gains.

Figure 4(b) further indicates that the reduction in context length primarily stems from visual token eviction. To better understand this behavior, we analyze the positional distribution of thinking steps. The resulting distribution reveals that thinking steps are concentrated near both the beginning and the later stages of the rationale. This suggests that MLLMs typically first focus on comprehending the textual prompt before attending to the image ([0, 0.4]), and then perform reasoning after visual observation ([0.4, 1.0]) to arrive at the final answer. We also observe that, during the reasoning process, MLLMs tend to alternate between looking and thinking steps, which likely facilitates more accurate and robust reasoning outcomes.

4.3.2 Error Analysis

We analyze the inference results of LaT on M³CoT generated by Qwen3-VL-2B-Instruct. We employ

Table 3: Effect of special token budget on performance and efficiency. Each configuration in the first column specifies the token budget assigned to the [Prefix], [StepType], [Compression], and [Continual] segments. Results are reported on Qwen3-VL-2B-Instruct evaluated on M³CoT.

# of Tokens	Acc ↑	Ctx Len ↓	Time ↓
(2, 4, 4, 1)	0.742	66K	2.46
(0, 4, 4, 1)	0.710	61K	2.38
(4, 4, 4, 1)	0.737	62K	2.38
(6, 4, 4, 1)	0.745	75K	2.57
(2, 2, 4, 1)	0.732	69K	2.51
(2, 6, 4, 1)	0.742	67K	2.42
(2, 8, 4, 1)	0.750	61K	2.40
(2, 8, 0, 1)	0.722	68K	2.66

GPT-5.1 to diagnose whether each error originates from the looking step or the thinking step. As shown in Table 2, errors are predominantly attributed to the thinking step across all domains, accounting for over 70% of the total errors. In particular, the dominance of thinking-step errors is most pronounced in the Mathematics domain, where over 90% of the errors arise from incorrect reasoning rather than visual perception. This indicates that the model performance is primarily constrained by deficiencies in multi-step reasoning rather than visual perception (Jin et al., 2025a).

4.3.3 Ablation Analysis on Special Token Budget

As discussed in Section 3.1.1, each special token of LaT is implemented as a segment consisting of multiple tokens. We therefore vary the token budget allocated to each segment to examine how the budget affects performance and efficiency (Table 3).

We first observe that introducing the [Prefix] segment consistently improves accuracy. Removing it entirely ((0, 4, 4, 1)) results in a substantial performance drop, whereas allocating only two tokens to [Prefix] already recovers most of the accuracy gains. Further increasing its token budget yields a modest but consistent improvement. These results suggest that [Prefix] provides effective planning space for predicting [StepType].

Next, we analyze the impact of enlarging the [StepType] segment. Increasing its token budget leads to simultaneous gains in both performance

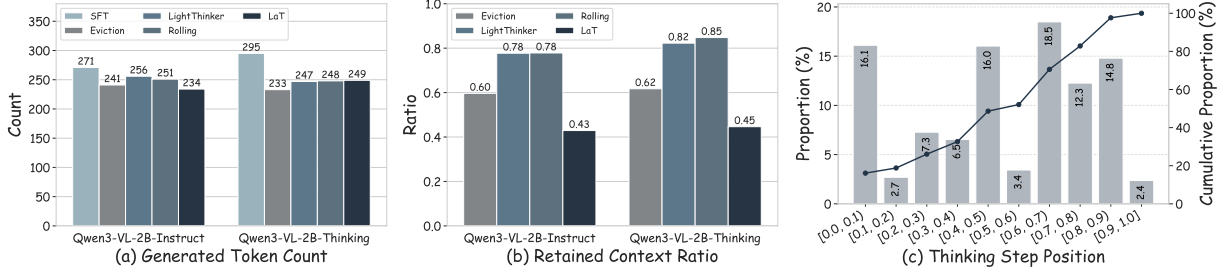


Figure 4: **Efficiency Analysis of LaT.** (a) The number of generated tokens using different Methods. (b) The retained context ratio, defined as the ratio of tokens preserved after eviction and compression to the number of tokens in the original context. (c) Positional distribution of thinking steps in rationales generated by Qwen3-VL-2B-Instruct with LaT, where each step position is defined as its step index normalized by the total number of steps in the rationale. All analyses are conducted on the test set of M³CoT.

and efficiency. Specifically, expanding [StepType] from 2 to 8 tokens improves accuracy from 0.732 to 0.750, while reducing the context length (from 69k to 61K) and accelerating inference (from 2.51 to 2.40). This indicates that a higher-capacity step-type representation enables more stable step-wise control.

Finally, we verify the necessity of explicitly decoupling current step-type declaration from previous-step compression. One might ask why we introduce a separate [Compression] segment instead of letting [StepType] jointly handle both roles. To evaluate this design choice, we fix the total step-prefix token budget but remove [Compression], reallocating its tokens to [StepType] ((2, 8, 0, 1)). This change leads to clear degradation in both accuracy and efficiency, indicating that separating step-type control and compression into distinct segments avoids role interference and yields better specialization, ultimately resulting in higher accuracy and lower inference cost.

Due to computational constraints, we do not exhaustively optimize the token budget. A more fine-grained tuning of segment-level token allocations may further improve performance on specific tasks.

4.3.4 Out-of-Domain Performance

We validate the out-of-domain performance of LaT on other multimodal benchmarks. The evaluation focuses on multimodal reasoning across general (Chen et al., 2024c), mathematical (Lu et al., 2023), and scientific domains (Yue et al., 2024, 2025). The evaluation results indicate that LaT can consistently increase the inference efficiency while preserving comparable performance with the SFT baseline.

Table 4: Results of LaT on additional challenging multimodal benchmarks using Qwen3-VL-2B-Instruct.

Method	Acc ↑	Ctx Len ↓	Time ↓
MathVista _{Mini}			
SFT	0.61	253K	4.56
LaT	0.60	108K	3.92
MMMU			
SFT	0.54	355K	5.33
LaT	0.55	208K	5.11
MMMU-Pro			
SFT	0.36	370K	5.59
LaT	0.35	199K	5.23
MMStar			
SFT	0.62	115K	2.82
LaT	0.63	75K	2.59

5 Conclusion

In this paper, we introduce LaT (Look and Think), the first modality-decoupled compression method for efficient multimodal reasoning. LaT structures reasoning into alternating looking and thinking steps. Built on this design, LaT evicts visual tokens whenever visual grounding is unnecessary and applies co-learning-guided compression after each completed step, substantially reducing context length while prompting cross-modality influence. Extensive experiments show that LaT achieves a strong balance between multimodal reasoning efficiency and performance, while also enhancing the transparency of the model’s reasoning process.

Limitations

While the LaT can effectively balance the multimodal reasoning efficiency and performance, our work still suffers from following limitations.

- Due to computational constraints, our evaluation of LaT is restricted to Qwen-3-VL-2B-Instruct and Qwen-3-VL-2B-Thinking, except for other model families or larger-scale MLLMs. Though our experiments demonstrate the potential of modality-decoupled compression, the generalizability of LaT across different architectures and model scales has not yet been fully explored.
- Although reinforcement learning (RL) has demonstrated effectiveness in improving reasoning abilities, LaT relies on supervised fine-tuning (SFT), which may introduce a higher risk of catastrophic forgetting of pre-training knowledge compared to RL (Zhang et al., 2025g).
- Our experiments primarily focus on a limited set of VQA benchmarks. Consequently, the applicability of LaT to broader task domains, such as code generation or agent-oriented tasks, has not yet been explored.

Acknowledgements

We thank the anonymous reviewers for their helpful feedbacks. The work described in this paper was partially funded by the National Key R&D Program of China (Grant No. 2023YFA1011601), the National Natural Science Foundation of China (Grant Nos. 62272173, 62273109), the Natural Science Foundation of Guangdong Province (Grant Nos. 2024A1515010089) and the Science and Technology Planning Project of Guangdong Province (Grant No. 2023A0505050106).

References

Pranjal Aggarwal and Sean Welleck. 2025. L1: Controlling how long a reasoning model thinks with reinforcement learning. *arXiv preprint arXiv:2503.04697*.

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, and 45 others. 2025. [Qwen3-vl technical report](#). *Preprint*, arXiv:2511.21631.

Rui Cai, Bangzheng Li, Xiaofei Wen, Muhao Chen, and Zhe Zhao. 2025. Diagnosing and mitigating modality interference in multimodal large language models. *arXiv preprint arXiv:2505.19616*.

Guoxuan Chen, Han Shi, Jiawei Li, Yihang Gao, Xiaozhe Ren, Yimeng Chen, Xin Jiang, Zhenguo Li, Weiyang Liu, and Chao Huang. 2024a. SepLLM: Accelerate large language models by compressing one segment into one separator. *arXiv preprint arXiv:2412.12094*.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. 2024b. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, pages 19–35. Springer.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and 1 others. 2024c. Are we on the right way for evaluating large vision-language models? *Advances in Neural Information Processing Systems*, 37:27056–27087.

Qiguang Chen, Libo Qin, Jin Zhang, Zhi Chen, Xiao Xu, and Wanxiang Che. 2024d. M³cot: A novel benchmark for multi-domain multi-step multi-modal chain-of-thought. *arXiv preprint arXiv:2405.16473*.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuqi Liu, Mengfei Zhou, Zhuosheng Zhang, and 1 others. 2024e. Do not think that much for 2+ 3=? on the overthinking of o1-like llms. *arXiv preprint arXiv:2412.21187*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shitong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Zonghao Guo, Ruyi Xu, Yuan Yao, Junbo Cui, Zanlin Ni, Chunjiang Ge, Tat-Seng Chua, Zhiyuan Liu, and Gao Huang. 2024. Llava-uhd: an lmm perceiving any aspect ratio and high-resolution images. In *European Conference on Computer Vision*, pages 390–406. Springer.

Tingxu Han, Zhenting Wang, Chunrong Fang, Shiyu Zhao, Shiqing Ma, and Zhenyu Chen. 2025. Token-budget-aware llm reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 24842–24855.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Mingyu Jin, Weidi Luo, Sitao Cheng, Xinyi Wang, Wenye Hua, Ruixiang Tang, William Yang Wang,

- and Yongfeng Zhang. 2025a. Disentangling memory and reasoning ability in large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1681–1701.
- Yizhang Jin, Jian Li, Tianjun Gu, Yexin Liu, Bo Zhao, Jinxiang Lai, Zhenye Gan, Yabiao Wang, Chengjie Wang, Xin Tan, and 1 others. 2025b. Efficient multimodal large language models: A survey. *Visual Intelligence*, 3(1):27.
- Yanwei Li, Chengyao Wang, and Jiaya Jia. 2024. Llama-vid: An image is worth 2 tokens in large language models. In *European Conference on Computer Vision*, pages 323–340. Springer.
- Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26296–26306.
- Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2023. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*.
- Haotian Luo, Li Shen, Haiying He, Yibo Wang, Shiwei Liu, Wei Li, Naiqiang Tan, Xiaochun Cao, and Dacheng Tao. 2025. O1-pruner: Length-harmonizing fine-tuning for o1-like reasoning pruning. *arXiv preprint arXiv:2501.12570*.
- Tergel Munkhbat, Namgyu Ho, Seo Hyun Kim, Yongjin Yang, Yujin Kim, and Se-Young Yun. 2025. Self-training elicits concise reasoning in large language models. *arXiv preprint arXiv:2502.20122*.
- OpenAI. 2025. [Gpt-5 mini](#). OpenAI API model documentation.
- Jianhui Pang, Fanghua Ye, Derek Wong, Xin He, Wanshun Chen, and Longyue Wang. 2024. Anchor-based large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 4958–4976.
- Xiaoye Qu, Yafu Li, Zhaochen Su, Weigao Sun, Jianhao Yan, Dongrui Liu, Ganqu Cui, Daizong Liu, Shuxian Liang, Junxian He, and 1 others. 2025. A survey of efficient reasoning for large reasoning models: Language, multimodality, and beyond. *arXiv preprint arXiv:2503.21614*.
- Akshat Ramachandran, Marina Neseem, Charbel Sakr, Rangharajan Venkatesan, Brucek Khailany, and Tushar Krishna. 2025. Thinkv: Thought-adaptive kv cache compression for efficient reasoning models. *arXiv preprint arXiv:2510.01290*.
- Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. 2025. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22857–22867.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Chenlong Wang, Yuanning Feng, Dongping Chen, Zhaoyang Chu, Ranjay Krishna, and Tianyi Zhou. 2025. Wait, we don't need to "wait"! removing thinking tokens improves reasoning efficiency. *arXiv preprint arXiv:2506.08343*.
- Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, and 1 others. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.
- Xinyi Wang, Lucas Caccia, Oleksiy Ostapenko, Xingdi Yuan, William Yang Wang, and Alessandro Sordani. 2023. Guiding language model reasoning with planning tokens. *arXiv preprint arXiv:2310.05707*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.
- Yuxin Wen, Qingqing Cao, Qichen Fu, Sachin Mehta, and Mahyar Najibi. 2024. Efficient vision-language models by summarizing visual tokens into compact registers. *arXiv preprint arXiv:2410.14072*.
- Zichen Wen, Yifeng Gao, Shaobo Wang, Junyuan Zhang, Qintong Zhang, Weijia Li, Conghui He, and Linfeng Zhang. 2025. Stop looking for important tokens in multimodal language models: Duplication matters more. *arXiv preprint arXiv:2502.11494*.
- Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2023. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*.
- Silei Xu, Wenhao Xie, Lingxiao Zhao, and Pengcheng He. 2025. Chain of draft: Thinking faster by writing less. *arXiv preprint arXiv:2502.18600*.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.
- Chenxu Yang, Qingyi Si, Mz Dai, Dingyu Yao, Mingyu Zheng, Minghui Chen, Zheng Lin, and Weiping Wang. 2025b. Test-time prompt intervention. *arXiv preprint arXiv:2508.02511*.

- Chenxu Yang, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Qiaowei Li, Minghui Chen, Zheng Lin, and Weiping Wang. 2025c. Dynamic early exit in reasoning models. *arXiv preprint arXiv:2504.15895*.
- Xubing Ye, Yukang Gan, Xiaoke Huang, Yixiao Ge, and Yansong Tang. 2025. Voco-llama: Towards vision compression with large language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 29836–29846.
- Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, and 1 others. 2024. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9556–9567.
- Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, and 1 others. 2025. Mmmu-pro: A more robust multi-discipline multimodal understanding benchmark. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15134–15186.
- Yufei Zhan, Ziheng Wu, Yousong Zhu, Rongkun Xue, Ruipu Luo, Zhenghao Chen, Can Zhang, Yifan Li, Zhentao He, Zheming Yang, and 1 others. 2025. Gthinker: Towards general multimodal reasoning via cue-guided rethinking. *arXiv preprint arXiv:2506.01078*.
- Anqi Zhang, Yulin Chen, Jane Pan, Chen Zhao, Aurojit Panda, Jinyang Li, and He He. 2025a. Reasoning models know when they’re right: Probing hidden states for self-verification. *arXiv preprint arXiv:2504.05419*.
- Jintian Zhang, Yuqi Zhu, Mengshu Sun, Yujie Luo, Shuofei Qiao, Lun Du, Da Zheng, Huajun Chen, and Ningyu Zhang. 2025b. Lightthinker: Thinking step-by-step compression. *arXiv preprint arXiv:2502.15589*.
- Peiyuan Zhang, Kaichen Zhang, Bo Li, Guangtao Zeng, Jingkang Yang, Yuanhan Zhang, Ziyue Wang, Haoran Tan, Chunyuan Li, and Ziwei Liu. 2024a. Long context transfer from language to vision. *arXiv preprint arXiv:2406.16852*.
- Qizhe Zhang, Aosong Cheng, Ming Lu, Zhiyong Zhuo, Minqi Wang, Jiajun Cao, Shaobo Guo, Qi She, and Shanghang Zhang. 2024b. [cls] attention is all you need for training-free visual token pruning: Make vlm inference faster. *arXiv e-prints*, pages arXiv–2412.
- Shaolei Zhang, Qingkai Fang, Zhe Yang, and Yang Feng. 2025c. Llava-mini: Efficient image and video large multimodal models with one vision token. *arXiv preprint arXiv:2501.03895*.
- Xiaofeng Zhang, Fanshuo Zeng, Yihao Quan, Zheng Hui, and Jiawei Yao. 2025d. Enhancing multimodal large language models complex reason via similarity computation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 10203–10211.
- Xiaoyun Zhang, Jingqing Ruan, Xing Ma, Yawen Zhu, Haodong Zhao, Hao Li, Jiansong Chen, Ke Zeng, and Xunliang Cai. 2025e. When to continue thinking: Adaptive thinking mode switching for efficient reasoning. *arXiv preprint arXiv:2505.15400*.
- Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and 1 others. 2024c. Sparsevlm: Visual token sparsification for efficient vision-language model inference. *arXiv preprint arXiv:2410.04417*.
- Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuan-dong Tian, Christopher Ré, Clark Barrett, and 1 others. 2023. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems*, 36:34661–34710.
- Zhi Zhang, Srishti Yadav, Fengze Han, and Ekaterina Shutova. 2025f. Cross-modal information flow in multimodal large language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19781–19791.
- Zhihao Zhang, Qiaole Dong, Qi Zhang, Jun Zhao, Enyu Zhou, Zhiheng Xi, Senjie Jin, Xiaoran Fan, Yuhao Zhou, Mingqi Wu, and 1 others. 2025g. Why reinforcement fine-tuning enables mllms preserve prior knowledge better: A data perspective. *arXiv preprint arXiv:2506.23508*.
- Haoran Zhao, Yuchen Yan, Yongliang Shen, Haolei Xu, Wenqi Zhang, Kaitao Song, Jian Shao, Weiming Lu, Jun Xiao, and Yueting Zhuang. 2025. Let llms break free from overthinking via self-braking tuning. *arXiv preprint arXiv:2505.14604*.
- Xin Zou, Di Lu, Yizhou Wang, Yibo Yan, Yuanhuiyi Lyu, Xu Zheng, Linfeng Zhang, and Xuming Hu. 2025. Don’t just chase "highlighted tokens" in mllms: Revisiting visual holistic context retention. *arXiv preprint arXiv:2510.02912*.

A Related Work

A.1 Visual Token Eviction and Compression

MLLMs typically encode images using a large number of visual tokens, despite these tokens being less information-dense than textual tokens (Jin et al., 2025b). Prior work reduces this redundancy mainly in two ways. One line of methods identifies and evicts unimportant visual tokens. These methods typically use attention scores (Chen et al.,

2024b; Zhang et al., 2024c,b) or visual tokens similarities (Zhang et al., 2025d; Wen et al., 2025) to identify candidates for removal. Recently, HoloV (Zou et al., 2025) further considers the semantic connectivity among visual tokens during eviction, leading to stronger performance. The other line of methods compress multiple visual tokens into a small set of compact representations (Shang et al., 2025; Guo et al., 2024). For example, LLaMA-VID (Li et al., 2024) and LLaVA-Mini (Zhang et al., 2025c) introduce extra lightweight modules that aggregate salient visual cues into a small set of tokens. Similarly, Victor (Wen et al., 2024) and VoCo-LLaMA (Ye et al., 2025) introduce additional trainable token embeddings and use the MLLM itself to summarize visual information to these embedding.

A.2 Efficient Multimodal Reasoning

Multimodal chain-of-thought (CoT) reasoning substantially improves the performance of MLLMs on challenging multimodal tasks (Zhan et al., 2025). However, long CoT traces incur significant computational overhead (Qu et al., 2025). Existing efforts to improve reasoning efficiency largely follow two directions: (1) inducing shorter reasoning traces and (2) reducing generated traces online during inference.

Methods for inducing shorter traces can be broadly categorized as training-free or training-aware. Training-free methods steer the model toward concise rationales purely at inference time: early work primarily relies on prompting to elicit fewer reasoning steps (Han et al., 2025; Xu et al., 2025), while later methods introduce inference-time interventions that enable finer-grained control over the reasoning process (Wang et al., 2025; Yang et al., 2025b,c; Zhang et al., 2025a). In contrast, training-aware methods explicitly adapt the model to produce concise outputs, typically via supervised fine-tuning (SFT) (Chen et al., 2024e; Munkhbat et al., 2025; Zhao et al., 2025) or reinforcement learning with carefully designed rewards (Zhang et al., 2025e; Luo et al., 2025; Aggarwal and Welleck, 2025).

Methods for reducing generated reasoning traces online can be further grouped into eviction-based and compression-based approaches. Eviction-based methods estimate the importance of past tokens using criteria such as their positions in the context or attention scores, and evict the KV cache of low-importance tokens before continuing gener-

ation (Zhang et al., 2023; Xiao et al., 2023; Chen et al., 2024a). Specifically, ThinkKV (Ramachandran et al., 2025) decomposes the reasoning trace into distinct thought types and designs an adaptive eviction strategy accordingly, achieving a better balance between the performance and efficiency. Compression-based methods, exemplified by Light-Thinker (Zhang et al., 2025b), instead train models to compress salient history into the hidden states of dedicated anchor tokens, while discarding the KV cache of the original tokens. Although Light-Thinker is closely related to our setting, it is designed for text-only reasoning and therefore does not capture multimodal behaviors such as cross-modality influence (Zhang et al., 2025f; Cai et al., 2025).

B Experiment

B.1 Datasets and Benchmarks

We use questions from the M³CoT training split to construct the modality-decoupled rationales, and mainly evaluate LaT’s performance on its test split. We also use four other benchmarks to assess LaT’s out-of-domain generalization.

- **M³CoT** (Chen et al., 2024d) is a challenging benchmark that spans Commonsense, Mathematics, and Science domains. Unlike most existing VQA benchmarks, M³CoT requires multi-step image and text modalities reasoning to arrive at the final answer, making it particularly suitable for assessing complex multimodal reasoning capabilities.
- **MathVista_{Mini}** (Lu et al., 2023) is the 1,000-example Test Mini split of MathVista, intended for lightweight yet reliable evaluation of mathematical reasoning in visual contexts (e.g., charts, plots, tables, and diagrams). The questions require jointly leveraging fine-grained visual perception and compositional math reasoning, covering diverse visually grounded math scenarios.
- **MMMU** (Yue et al., 2024) is a college-level, multi-discipline benchmark consisting questions collected from exams, quizzes, and textbooks. It spans six broad disciplines and many specialized subfields to evaluate expert knowledge and deliberate multimodal reasoning.
- **MMMU-PRO** (Yue et al., 2025) is a robustness-oriented variant of MMMU de-

signed to reduce shortcut solutions and more strictly test true multimodal understanding. It is constructed by filtering questions that text-only models can answer, augmenting the candidate option set, and introducing a vision-only setting where questions are embedded inside images, thereby enforcing tighter vision-language integration.

- **MMStar** (Chen et al., 2024c) is a vision-indispensable benchmark of 1,500 human-validated samples curated to minimize non-visual solvability and unintentional data leakage. It is organized around six core multimodal capabilities and 18 fine-grained axes, providing a cleaner stress test of general multimodal understanding and reasoning.

B.2 Training Data

In our experiments, we employ GPT-5 Mini as both the data MLLM and the OCR MLLM to generate modality-decoupled rationales. We access the models via OpenRouter¹. During data generation, we keep the default API settings and sampling configuration, except that we set `reasoning_effort` to low to cut API costs². Using this setup, we construct 7,861 training examples at a total cost of \$12.33. The prompts used for data MLLM and OCR MLLM are shown in Figure 5 and Figure 6.

Examples of our step-wise rationale for modality decoupling are shown in Figure 7 and Figure 8.

B.3 Implementation Details

We implement LaT on top of the Hugging Face Transformers library and fine-tune our MLLM using Hugging Face’s Trainer API. The hyper-parameters for supervised fine-tuning are provided in Table 5. During training, we save 10 checkpoints at uniform intervals and evaluate each one on a held-out M³CoT validation subset. We then select the checkpoint with the best validation accuracy and use it to run inference on the test set to obtain our final reported results. All baselines follow the same experimental protocol, using identical training settings and the same validation based checkpoint selection criterion, to ensure a fair comparison. We employ greedy decoding and set `max_generation_length` to 4096 during evaluation. All evaluations are conducted on one RTX

Table 5: The hyper-parameters of the experiment. For hyper-parameters not listed in the table, we use the default values from Hugging Face’s Trainer.

Name	Value
precision	bf16
max_seq_length	2048
epoches	4
learning_rate	5e-5
global_batch_size	4
optim	adamw_torch
warmup_ratio	0.05

3090 GPU to ensure a fair comparison across methods.

B.4 Examples

In Figure 9 and Figure 10, we show representative examples of fine-tuning Qwen3-VL-2B-Instruct with LaT on different benchmarks. The results clearly demonstrate that LaT effectively decouples the modalities involved in MLLM inference.

B.5 Error Analysis

In Section B.1, our analysis reveals that most errors originate from the thinking steps rather than the looking steps. The prompt used by GPT-5.1 to analyze the error is shown in Figure 11.

We further provide an example to illustrate how these errors arise. As shown in Figure 12, in Step 2 and Step 3 (looking steps), the model correctly identifies salient visual cues from the tangram: a pointed, slightly curled toe with a narrow, elongated upper (Step 2), together with a wider rear region resembling a sole (Step 3). However, in Step 4 (a thinking step), it makes an incorrect semantic leap when mapping these cues to the answer choices: it rules out the elf shoe by wrongly asserting that an elf shoe is “not pointed,” and instead concludes the shape corresponds to a sandal. This faulty elimination in the text-only reasoning stage cascades to the final prediction, leading to an incorrect answer despite largely accurate visual perception.

¹<https://openrouter.ai/>

²<https://platform.openai.com/docs/guides/reasoning#get-started-with-reasoning>

Data MLLM Prompt

Task Description

You will be given:

1. A **multiple-choice question** that includes both an **image** and **text**.
2. The **correct answer** to that question.

Your task is to produce a **reasoning trace** — a sequence of logical steps showing **how to arrive at the correct answer**.

Step Construction Guidelines

Each reasoning step must:

- Have a clear **name**.
- Indicate its **type**:
 - Use '[visual_understanding]' if the step requires understanding the image (e.g., identifying objects, actions, or spatial relationships).
 - Use '[textual_reasoning]' if the step involves reasoning based on the text (e.g., interpreting the question, comparing options, or drawing logical conclusions). When performing '[textual_reasoning]', you should **not refer to or access any visual information from the image**.
- Include a **content** field that explains your reasoning or visual understanding.

Special Requirements

- If a single step involves both **visual understanding** and **textual reasoning**, **split it into two separate steps** — one labeled '[visual_understanding]' and the other '[textual_reasoning]'. If it's difficult to separate the two, label the step as '[visual_understanding]'.
- If the image also contains a set of options, which may cause confusion with the set of options provided in the text,
 - Use the format **choice (A) in the given text** to refer to text-based options.
 - Use the format **option A in the image** to refer to image-based options.
- The provided correct answer is for supervision only, **Do not reference, restate, or compare with the given correct answer**.

Expected Output Format

```
“json
[
  {
    "step_name": "Step name.",
    "type": "Must be one of [visual_understanding] or [textual_reasoning].",
    "content": "Reasoning content."
  },
  ...
]
```

Figure 5: System prompt of data MLLM. The question, image, and the correct answer are provided in the user prompt.

OCR MLLM Prompt

Determine whether the image contains **rich textual information** — such as charts, diagrams, signs, documents, screenshots, or any other visuals with a significant amount of text.

Return your result in the following format:

```
““json
{
  "rich_textual_info_flag": true or false,
}
““
```

Extract the text of the image.

Return your result in the following format:

```
““json
{
  "extracted_text": "...",
}
““
```

Figure 6: Two system prompts for the OCR MLLM: one prompt judges whether OCR is needed, and the other extracts the image-embedded text. Images are provided in the user prompts.

Step-Wise Rationale



Question: What is the spoon on the plate going to be used for?

- (A) Stirring the soup in the bowl
- (B) Eating the food on the plate
- (C) Scooping the sauce from the container
- (D) Measuring the sugar for the coffee

Rationale:

[Looking]: There is a plate with a sandwich and a spoon resting on the plate in front of a container of sauce or spread. The spoon is positioned on the plate near the sandwich and the container is open with a ladle-like utensil inside.

[Thinking]: Review the answer choices: stirring soup, eating the food on the plate, scooping sauce from a container, or measuring sugar for coffee. Determine which action plausibly matches the scene without using the image.

[Looking]: The spoon is on the plate next to the sandwich and an open container of a sauce/spread, suggesting the spoon would be used with that container. There is no bowl of soup or coffee visible to justify stirring or measuring.

[Thinking]: Eliminate stirring soup (no soup bowl visible) and measuring sugar for coffee (no coffee or sugar container shown). Eating the food with that type of spoon is less likely than using it to transfer sauce from the container.

[Thinking]: Conclude that the spoon is most likely intended for scooping sauce from the container to add to the sandwich, matching choice (C).

Figure 7: Step-wise rationale of sample physical-commonsense-1434 in M³CoT.

Step-Wise Rationale



Question: Think about the magnetic force between the magnets in each pair. Which of the following statements is true?

- (A) The magnitude of the magnetic force is smaller in Pair 2.
- (B) The magnitude of the magnetic force is smaller in Pair 1.
- (C) The magnitude of the magnetic force is the same in both pairs.

Rationale:

[Thinking]: The question asks about the relative magnitude of the magnetic force between the magnets in two different pairs. We must compare Pair 1 and Pair 2 and decide which pair has the smaller force (or if they are equal).

[Looking]: Each pair shows two ring (annular) magnets side by side with the same center-to-center separation scale (3.95 cm). In both pairs the two magnets in a pair have the same outer diameters: the left magnet in each pair is small and the right magnet is larger. The colors label poles: blue for S and green for N (or vice versa), showing half-and-half pole distribution around the ring circumference.

[Looking]: In Pair 1, the small (left) magnet shows S on the left side and N on the right side; the adjacent large (right) magnet shows N on the left and S on the right. This means where the two magnets face each other across the gap, opposite poles are adjacent (small's right = N faces large's left = N). Actually the facing regions are same-pole (N next to N) at the nearest edges.

[Looking]: In Pair 2, the small magnet has S on the left and N on the right, while the large magnet has S on the left and N on the right as well. Thus at the facing edges the poles are opposite (small's right = N faces large's left = S), producing attractive alignment at the nearest regions.

[Thinking]: Magnetic force between two magnets depends on the interaction of adjacent pole regions: opposite poles (N next to S) produce attraction, same poles (N next to N or S next to S) produce repulsion. When more of the facing area has opposite poles, the net attractive force is larger; when facing areas are same-pole, the net force is reduced.

[Thinking]: Because the nearest adjacent regions in Pair 1 are same poles facing each other, those regions contribute repulsion or reduced attraction, lowering the overall magnitude of the magnetic force between the magnets.

[Thinking]: Because the nearest adjacent regions in Pair 2 are opposite poles facing each other, those regions contribute stronger attraction, increasing the overall magnitude of the magnetic force between the magnets.

[Thinking]: Comparing the two, Pair 1 has facing same poles that reduce the net attraction, while Pair 2 has facing opposite poles that increase attraction. Therefore the magnitude of the magnetic force is smaller in Pair 1.

Figure 8: Step-wise rationale of sample physics-1033 in M³CoT.

Correct Example Using LaT

Let me give you the context: Evening grosbeaks eat small, hard seeds. The shape of the 's beak is adapted to crack open small, hard seeds. Figure: evening grosbeak.



(A) Figure A



(B) Figure B

Question: Your task is to select an option where the organism shown in the picture shares similar biological adaptations to its environment.

- (A) Figure A
- (B) Figure B

LaT's Generation:

[prefix_0] [prefix_1] [thinking_0] [thinking_1] [thinking_2] [thinking_3]
[compression_0] [compression_1] [compression_2] [compression_3] [continual]
The question asks which organism shown in the picture shares similar biological adaptations to its environment. This requires comparing the pictured organism to the two choice images and selecting the one that matches the described adaptation (an insect with a beak adapted for cracking open small hard seeds).

[prefix_0] [prefix_1] [looking_0] [looking_1] [looking_2] [looking_3]
[compression_0] [compression_1] [compression_2] [compression_3] [continual]
The central image shows a bird with a stout, conical beak - a bird that cracks open small hard seeds.

[prefix_0] [prefix_1] [thinking_0] [thinking_1] [thinking_2] [thinking_3]
[compression_0] [compression_1] [compression_2] [compression_3] [continual]
The question asks which figure shares similar biological adaptations to its environment. The described adaptation is a seed-cracking beak, so we need the option that depicts a bird with a similar beak shape.

[prefix_0] [prefix_1] [looking_0] [looking_1] [looking_2] [looking_3]
[compression_0] [compression_1] [compression_2] [compression_3] [continual]
Figure A shows a bird with a short, stout, conical beak typical of insectivorous or seed-eating birds. Figure B shows a long, heavy bill (a pelican-type bill) adapted for filtering or scooping, not for cracking seeds.

[prefix_0] [prefix_1] [thinking_0] [thinking_1] [thinking_2] [thinking_3]
[compression_0] [compression_1] [compression_2] [compression_3] [continual]
The conical beak in Figure A matches the described seed-cracking adaptation, while the long bill in Figure B does not. Therefore the organism in Figure A shares similar biological adaptations to its environment.

Figure 9: Step-wise rationale generated by Qwen3-VL-2B-Instruct with LaT on M³CoT sample biography-1547.

Correct Example Using LaT



Question: Which property do these two objects have in common?

- (A) yellow
- (B) salty
- (C) nan
- (D) nan

LaT's Generation:

[prefix_0] [prefix_1] [thinking_0] [thinking_1] [thinking_2] [thinking_3]
[compression_0] [compression_1] [compression_2] [compression_3] [continual]
The question asks which property the two pictured objects have in common, so we must compare the two objects and find a shared characteristic from the given choices.

[prefix_0] [prefix_1] [looking_0] [looking_1] [looking_2] [looking_3]
[compression_0] [compression_1] [compression_2] [compression_3] [continual]
The left image shows an ocean wave labeled 'ocean water', which is a liquid body of water. The right image shows a bag of potato chips labeled 'potato chips', which are a food item made from potato.

[prefix_0] [prefix_1] [thinking_0] [thinking_1] [thinking_2] [thinking_3]
[compression_0] [compression_1] [compression_2] [compression_3] [continual]
The question asks which property these two objects have in common. We must compare the properties of ocean water and potato chips to find a shared attribute among the given choices.

[prefix_0] [prefix_1] [thinking_0] [thinking_1] [thinking_2] [thinking_3]
[compression_0] [compression_1] [compression_2] [compression_3] [continual]
Compare the properties: ocean water is typically salty (contains dissolved salts), while potato chips are not inherently salty. Therefore the shared property is 'salty'.

[prefix_0] [prefix_1] [thinking_0] [thinking_1] [thinking_2] [thinking_3]
[compression_0] [compression_1] [compression_2] [compression_3] [continual]
Compare the textual choices (A) yellow, (B) salty, (C) nan, (D) nan. The correct property is 'salty', which corresponds to choice (B).

Figure 10: Step-wise rationale generated by Qwen3-VL-2B-Instruct with LaT on a MMStar sample.

System prompt for analyzing error.

Task Description

You are given a **rationale** that leads to an **incorrect final answer**. Your task is to determine the step that mistakes happen.

—

Output Format

If you find an error in the rationale:

Return a JSON object in the following format:

```
““  
{  
  "step_index": "The index of the step where the error occurs.",  
  "reason": "A brief explanation of why this step is incorrect."  
}  
““
```

If the rationale is logically correct so far but is incomplete:

Return a JSON object in the following format:

```
““  
{  
  "reason": "Not finished."  
}  
““
```

Figure 11: System prompt for analyzing the cause of the reasoning error. The question, image, the generated rationale, and the correct answer are provided in the user prompt.

Error Case



Question: What is the object in the Tangram Shapes image representing?

- (A) A ballet slipper
- (B) A cowboy boot
- (C) A sandal
- (D) An elf shoe

LaT's Generation:

[prefix_0] [prefix_1] [looking_0] [looking_1] [looking_2] [looking_3]
[compression_0] [compression_1] [compression_2] [compression_3] [continual]
Observe the tangram composition: a long curved shape on the left (dark pieces forming a pointed toe and a curved shaft) and a wider, flared shape on the right (pink pieces forming a broad sole and an outward flare). The left piece resembles a pointed toe and a narrow shaft, while the right piece is broader and more open.

[prefix_0] [prefix_1] [thinking_0] [thinking_1] [thinking_2] [thinking_3]
[compression_0] [compression_1] [compression_2] [compression_3] [continual]
Consider the textual choices without using the image: ballet slipper (typically narrow and rounded), cowboy boot (often thick and with a prominent horn or flare), sandal (narrow and open), elf shoe (often stylized with a pointed toe and a decorative shape). Match general silhouettes to the observed tangram shapes.

[prefix_0] [prefix_1] [looking_0] [looking_1] [looking_2] [looking_3]
[compression_0] [compression_1] [compression_2] [compression_3] [continual]
The left dark shape has a pointed front and a series of connected segments that suggest a narrow, elongated front with a slight curve, consistent with a pointed toe and a narrow shaft rather than a rounded ball or thick sole.

[prefix_0] [prefix_1] [looking_0] [looking_1] [looking_2] [looking_3]
[compression_0] [compression_1] [compression_2] [compression_3] [continual]
The right side is wider and more open, with a broad triangular piece that suggests an open front or slip-on design rather than a closed, rounded silhouette.

[prefix_0] [prefix_1] [thinking_0] [thinking_1] [thinking_2] [thinking_3]
[compression_0] [compression_1] [compression_2] [compression_3] [continual]
Given the pointed front and open rear, the silhouette best matches a sandal (open-toed, pointed toe) rather than a ballet slipper (closed, rounded), cowboy boot (typically thicker and closed), or an elf shoe (often stylized and not pointed).

[prefix_0] [prefix_1] [thinking_0] [thinking_1] [thinking_2] [thinking_3]
[compression_0] [compression_1] [compression_2] [compression_3] [continual]
Given the pointed front and open rear, the tangram most closely represents a sandal, so choose option (C).

Figure 12: Error case produced by Qwen3-VL-2B-Instruct with LaT on M³CoT sample cognitive-science-504. The text highlighted in red marks the source of the mistake, which leads the model to select option (C) rather than the correct answer (D).