Visual Latents Know More Than They Say: Unsilencing Latent Reasoning in MLLMs

Xin Zhang[1] **Qiqi Tao**[1,2] **Jiawei Du**[1] **Moyun Liu**[3] **Joey Tianyi Zhou**[4,1] ✉
[1]Institute of Advanced Intelligence and Computing,
Agency for Science, Technology and Research (A*STAR), Singapore
[2]Singapore University of Technology and Design, Singapore
[3]Huazhong University of Science and Technology, China
[4]Singapore Management University, Singapore
{zhangx7, dujw, Joey_Zhou}@a-star.edu.sg
qiqi_tao@mymail.sutd.edu.sg, lmomoy@hust.edu.cn
✉ Corresponding author

Abstract

Continuous latent-space reasoning offers a compact alternative to textual chain-of-thought for multimodal models, enabling high-dimensional visual evidence to be integrated without explicit reasoning tokens. However, we identify a previously overlooked optimization pathology in existing latent visual reasoning methods: although visual latents become semantically enriched during training, their contribution to final answer prediction is systematically suppressed. Within the shared parameter space, the autoregressive objective favors shortcut reliance on direct visual input, driving latent tokens toward transition-like states rather than informative reasoning content. We term this phenomenon ***Silenced Visual Latents***. To address it, we disentangle the two conflicting objectives by directly optimizing the latent reasoning at inference time, keeping backbone parameters frozen. In Stage I, visual latents are warmed up via query-guided contrastive latent–visual alignment, improving semantic quality while preventing latent collapse. In Stage II, the latent reasoning is further optimized via a confidence-progression reward, which incentivizes predicted token distributions along the latent span to become progressively more concentrated, routing predictions through the latent reasoning rather than bypassing it. Experiments across eight benchmarks and four model backbones show that inference-time latent optimization, without any parameter updates, effectively unleashes the suppressed reasoning capacity of visual latents. Code is available at https://github.com/zhangxin-xd/Unsilencing-Latent-Reasoning.

1 Introduction

Multimodal Large Language Models (MLLMs) [43, 1, 6] reconfigure the Large Language Model (LLM) paradigm beyond text-only inputs, integrating visual encoders to enable joint reasoning over language and visual perception. While this evolution brings remarkable advances in various cross-modal reasoning tasks, such as scientific visual question answering [51, 36], mathematics solving [38, 9], and visual grounding [2, 28], the asymmetric inductive bias toward language frequently manifests as hallucinated visual understanding [33, 39, 30].

A straightforward remedy is to follow the standard Chain-of-Thought (CoT) [35, 34] paradigm, encouraging explicit textual reasoning prior to the final answer prediction [47, 48, 46]. Subsequent efforts further expand the evidential space available during reasoning by enabling active visual manipulations [24, 13, 49] or intermediate generation [42, 37, 8]. In short, the key idea is to embed additional

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

auxiliary reasoning contexts to maximize the probability of the correct answer. Another promising approach is *Latent Reasoning* [50], recently explored in LLMs (e.g., Coconut [12], CoLaR [27]), which shifts intermediate reasoning from discrete tokens to continuous latent representations [25, 26]. By operating in latent space, this paradigm increases representational bandwidth while reducing autoregressive context length. This expanded capacity is particularly advantageous for multimodal reasoning, enabling compact integration of high-dimensional visual evidence without external tool invocation or complex pixel-level generation [15, 20].

Current *Latent Visual Reasoning (LVR)* methods instantiate latent representations by aligning them with diverse forms of pre-curated visual evidence, such as trajectory maps in Mirage [41], regions of interest (ROIs) in LVR [15], 3D geometric priors in 3DThinker [5], and combinations of segmentation, depth, and edge maps in CoVT [23]. Beyond mere alignment with these pre-curated informative visual targets, the latents are simultaneously optimized to ensure accurate autoregressive answer prediction. Despite substantial improvements in multimodal reasoning, a critical inconsistency actually persists under joint optimization. Through preliminary analysis, we observe that visual latents become progressively better aligned with task-relevant visual evidence, yet answer accuracy does not improve accordingly and often fluctuates. This reveals a counterintuitive dissociation between *latent quality* and *latent effectiveness*: stronger latent representations are not necessarily translated into stronger final prediction behavior. As illustrated in Figure 1, this is because, within the shared parameter space, the autoregressive answer prediction objective tends to favor shortcut routes through direct visual input, while latent tokens are gradually pushed toward transition-like behavior rather than semantically informative reasoning states. Thus, the resulting latent representations are learned in a compromised manner and systematically under-utilized at answer time, which we term ***Silenced Visual Latents***.

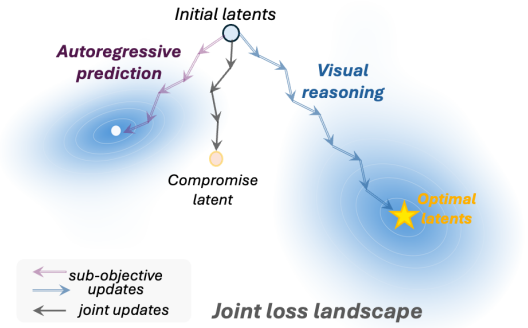

Figure 1: The joint loss landscape of Latent Visual Reasoning. Under joint optimization, the initial latents are simultaneously pulled toward two conflicting attractors: the autoregressive prediction objective (left) and the visual reasoning objective (right). The shortcut favored by the autoregressive prediction objective dissociates latent quality from latent effectiveness and bypasses meaningful latent reasoning, driving the latents toward a compromise state.

Based on this insight, we disentangle these two objectives by directly optimizing the latent reasoning at inference time, keeping backbone parameters frozen. In Stage I, we perform visual latent warm-up via query-guided contrastive latent–visual alignment: visual tokens are ranked by their relevance to the query and partitioned into chunk-wise positive and negative sets, pulling each latent toward query-relevant visual evidence while pushing it away from irrelevant patches, with chunk-wise assignment preventing latent collapse by ensuring each latent token attends to a distinct subset of visual evidence. In Stage II, we enforce latent utilization via a confidence-progression reward: the latent reasoning is iteratively perturbed and updated so that predicted token distributions along the latent span become progressively more concentrated, directly incentivizing the model to route predictions through the latent reasoning rather than bypassing it. Together, they directly address both failure modes of Silenced Visual Latents by optimizing the latent reasoning itself, without modifying any model parameters.

Our contributions are threefold: **(1)** We identify the Silenced Visual Latents phenomenon, revealing that visual latents become enriched yet remain under-utilized due to autoregressive shortcuts in the shared parameter space. **(2)** We propose an inference-time framework that disentangles latent quality from latent utilization via contrastive alignment and reward-driven reinforcement, with frozen backbone. **(3)** Extensive experiments across eight benchmarks and four model backbones confirm the effectiveness and generalizability of our framework, requiring no parameter updates.

2 Related Works

Latent space is broadly acknowledged to be more expressive than discrete token space, thanks to its continuous nature, inspiring a growing line of work that moves intermediate reasoning into the continuous domain. Latent reasoning in LLMs was pioneered by Coconut [12], which feeds the model's last hidden state back as input without decoding, enabling implicit breadth-first search in

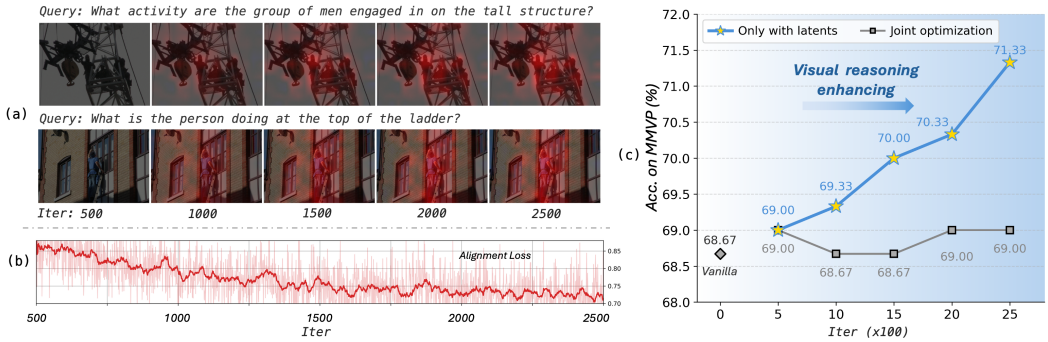

Figure 2: Enhanced visual latents improve reasoning performance, but this benefit is suppressed under joint optimization. Left: Visual latents are progressively enhanced during training, as evidenced by the deepening red color indicating higher similarity between latents and pre-curated visual clues (a), while the alignment loss decreases steadily (b). Right: Although visual latents continue to improve, the jointly optimized model's performance still fluctuates (gray line). In contrast, equipping the vanilla model with latents from each checkpoint yields a monotonically increasing trend (blue line), revealing that enhanced latents do benefit reasoning, but their contribution is suppressed under joint optimization.

latent space. Subsequent works improve upon this: CODI [25] aligns latent states with an explicit teacher via self-distillation, CoLaR [27] dynamically compresses reasoning chains to reduce inference cost, and Deliberation [17] augments the KV cache with differentiable latent vectors to enable iterative refinement of reasoning states without generating explicit tokens. These works collectively show that latent reasoning can be more expressive and efficient than textual CoT.

Extending latent reasoning to multimodal settings also brings additional advantages: the model must integrate high-dimensional visual evidence into the reasoning process without resorting to costly pixel-level generation or external tool invocations. Several paradigms have recently emerged to explore this. Mirage [41] inserts compact latent visual tokens into the autoregressive reasoning and aligns them with trajectory-map visual signals, enabling visual imagination without decoding actual images. LVR [15] supervises latent tokens with pre-curated regions of interest (ROIs) and jointly optimizes them with answer prediction. CoVT [23] enriches this supervision by incorporating diverse visual targets, such as segmentation, depth, and edge maps, providing richer visual grounding for latent tokens. Furthermore, Monet [32] improves supervision for latent visual reasoning by training MLLMs to generate continuous visual-thought embeddings through distillation-based SFT and visual-latent policy optimization. More recently, 3DThinker [5] incorporates 3D geometric priors as visual supervision, broadening the scope of latent visual reasoning to spatial understanding tasks. Despite their diversity in supervision design, most existing methods share a common optimization scheme: latent representations are optimized jointly with autoregressive answer prediction under a shared parameter space. This coupling implicitly assumes that improving the semantic quality of visual latents will naturally translate into stronger answer prediction. However, as we show in this work, the autoregressive objective can instead favor shortcut reliance on direct visual inputs, driving visual latents toward suboptimal compromise states and causing them to be under-utilized during final prediction. We refer to this joint failure as Silenced Visual Latents.

3 Methodology

Problem Formulation. Consider an MLLM $\mathcal{M} = \mathcal{F} \circ \Phi$, where $\mathcal{F} = \{f(\cdot), g(\cdot)\}$ consists of a visual encoder $f(\cdot)$[1] and a text embedding layer $g(\cdot)$. These modules encode visual and textual inputs, respectively, before they are fed into the language backbone Φ parameterized by $\boldsymbol{\theta}$. For a multimodal input $\{\texttt{Vision:}\mathcal{V} = (\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_N), \texttt{Query:}\mathcal{Q}\}$, $\Phi_{\boldsymbol{\theta}}$ is prompted to first produce a T-step reasoning sequence $\mathcal{C} = (\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_T)$ and then generate the final answer $\mathcal{A} = (\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_L)$. The key insight of Latent Visual Reasoning (LVR) is to enhance multimodal reasoning by lifting intermediate reasoning from discrete textual tokens $\mathcal{C}$ to continuous visual latent representations $\mathcal{H} = (\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_K)$[2], where $K \ll T$. Thus, current paradigms optimize $\Phi_{\boldsymbol{\theta}}$ with a joint objective that (i) ***encourages visual reasoning in the latent space***, while (ii) ***enforcing faithful***

[1]The visual encoder includes the projector; omitted for brevity.

[2]Special tokens `<latent_start>` and `<latent_end>` are prepended and appended to $\mathcal{H}$, respectively, to delimit the latent reasoning segment.

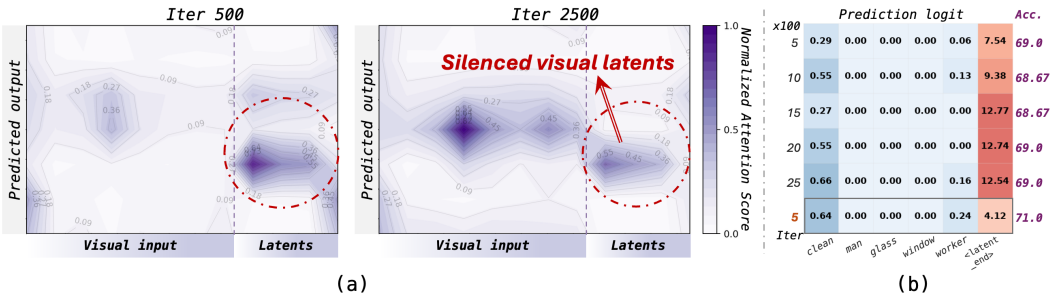

Figure 3: Joint optimization silences visual latents. (a) During training, attention gradually drifts toward the input visual tokens rather than the visual latents, indicating that visual latents progressively lose their voice in answer prediction. (b) Prediction logits of the first latent token at different training stages. The latent token is increasingly pushed toward the `<latent_end>` token, indicated by the overwhelmingly larger logit. 5 denotes the checkpoint optimized at 500_{th} *Iter* with half the weight of the autoregression loss, which alleviates the damage to the semantic meaning of the latent token, yielding lower `<latent_end>` logits and higher accuracy.

answer text autoregression,

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \underbrace{\frac{1}{K}\sum_{k=1}^{K} \|\boldsymbol{h}_k - \tilde{\boldsymbol{v}}_k\|_2^2}_{\text{① visual latent reasoning}} \underbrace{-\lambda \sum_{l=2}^{L} \log \Phi_{\boldsymbol{\theta}}\left(\boldsymbol{a}_l \mid \mathcal{V}, \mathcal{Q}, \mathcal{H}, \boldsymbol{a}_{1:l-1}\right)}_{\text{② answer autoregression}}. \tag{1}$$

Here, $\tilde{\mathcal{V}} = (\tilde{\boldsymbol{v}}_1, \tilde{\boldsymbol{v}}_2, \ldots, \tilde{\boldsymbol{v}}_K)$ denotes the visual reasoning clues (*e.g.* regions of interest (ROIs) suggested by LVR [15]) for alignment. λ is a balancing hyperparameter between the two loss terms.

3.1 Silenced Visual Latents

Observation #1: ***Better visual latents are the key to improving multimodal reasoning.***

To understand how visual latents affect multimodal reasoning, we begin by examining what they encode. As shown in Figure 2 (a) and (b), visual latents become increasingly aligned with the pre-defined visual chain as the alignment loss decreases, suggesting that they progressively capture more task-relevant visual information. Naturally, we may ask: do such progressively enhanced visual latents foster reasoning? To isolate the effect of visual latents from the jointly optimized answer prediction branch, we use vanilla Qwen2.5VL-7B (*i.e.* model at 0_{th} *Iter*) and replace only its visual latents with those produced by later optimized checkpoints. As shown by the blue line in Figure 2 (c), the accuracy improves as the donated visual latents become stronger, indicating that better visual latents indeed contribute positively to multimodal reasoning.

Observation #2: ***Joint optimization silences the visual latents.***

However, the benefit in Observation #1 is not faithfully reflected in the performance of the jointly optimized checkpoints. As shown by the gray line in Figure 2 (c), the accuracy of the actual checkpoints fluctuates rather than improving monotonically over training. For example, the accuracy at the 2500_{th} *Iter* only reaches the level of the 500_{th} *Iter*, despite the substantially better visual latents. To further investigate this discrepancy, we visualize the attention maps of models at different training stages. As shown in Figure 3 (a), after training, prediction becomes dominated by the visual input rather than the optimized latents. In addition, Figure 3 (b) shows that the prediction logits of latent tokens on answer-related text tokens increase during training; however, the logits on the transition token `<latent_end>` also increase and become overwhelmingly larger. Can a smaller λ alleviate this issue? As shown in the figure, using half of the original λ enables the logits of answer-related tokens to reach, by the 500_{th} *Iter*, a level comparable to that obtained at the 2500_{th} *Iter*. Meanwhile, it yields smaller `<latent_end>` logits, thereby improving the overall performance. These observations suggest the optimization dynamic illustrated in Figure 1: both objectives are optimized within the same parameter space. As $\boldsymbol{\theta}$ is updated toward a joint optimum, the autoregressive loss rapidly exploits shortcuts grounded in the visual input, while simultaneously driving latent tokens toward a transition-token-like role at the expense of their semantic content. Consequently, joint optimization produces compromised latent tokens whose contribution is no longer effectively retained in the final answer prediction. We term this phenomenon "***Silenced Visual Latents***."

As analyzed above, these two objectives are difficult to reconcile within a shared optimization space, and the resulting "Silenced Visual Latents" lead to unsatisfactory performance. A natural question, then, is whether their optimization spaces can be disentangled so as to preserve autoregressive answer prediction while fully exploiting the reasoning capacity of visual latents. To this end, we directly

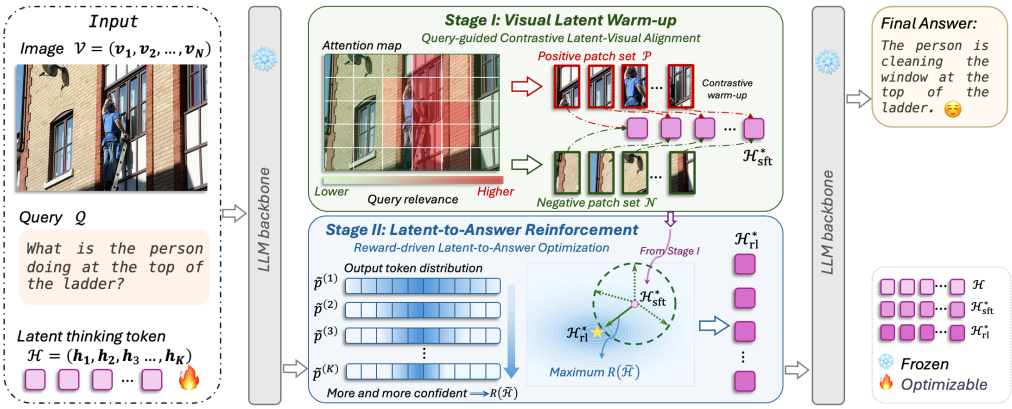

Figure 4: Overview of the proposed framework for unsilencing visual latents. Stage I performs query-guided contrastive latent–visual alignment with positive and negative patch sets to enhance latent semantics. Stage II initializes from the Stage-I latents and applies a confidence-progression reward that encourages progressively more concentrated output token distributions, yielding the final latents $\mathcal{H}^*$ for answer generation.

optimize the visual latents while retaining the original model parameters for autoregressive textual generation. Concretely, at test time, we perform per-instance latent-token optimization for each input sample, thereby enhancing latent effectiveness without compromising the model's original capabilities. Based on the aforementioned observations, we optimize visual latents in two stages: *Stage I: Visual Latent Warm-up* and *Stage II: Latent-to-Answer Reinforcement*.

3.2 Stage I: Visual Latent Warm-up

As the optimization space is disentangled, visual latents can no longer rely on the model's autoregressive generation. Accordingly, as shown in Figure 4, we explicitly reserve latent slots before answer generation to ensure that these tokens remain functionally involved in the reasoning process. Stage I is designed to enhance the semantic quality of visual latents before enforcing their contribution to final answer prediction. Existing approaches [15, 32] typically supervise latent tokens through positive alignment with visual reasoning signals. However, this positive-only supervision does not explicitly distinguish relevant visual evidence from irrelevant or misleading content. To address this limitation, we introduce a query-guided contrastive warm-up objective that imposes a discriminative constraint on each latent token.

Let $\boldsymbol{A} \in \mathbb{R}^{M \times M}$ denote the token-level attention matrix averaged across all layers and heads of $\Phi_{\boldsymbol{\theta}}$, where M is the total sequence length, and let $\mathcal{I}_{\mathcal{V}} = \{1, \ldots, N\}$ and $\mathcal{I}_{\mathcal{Q}} \subseteq \{N+1, \ldots, M\}$ denote the position index sets of $\mathcal{V}$ and $\mathcal{Q}$ within the full sequence, respectively. For each visual token $\boldsymbol{v}_n \in \mathcal{V}$, its query-guided relevance score is defined as

$$s_n = \frac{1}{|\mathcal{I}_{\mathcal{Q}}|} \sum_{q \in \mathcal{I}_{\mathcal{Q}}} \boldsymbol{A}_{q,n}, \qquad n \in \mathcal{I}_{\mathcal{V}}, \tag{2}$$

which measures the average attention that $\mathcal{Q}$ allocates to each visual token in $\mathcal{V}$.

The visual tokens in $\mathcal{V}$ are ranked in descending order of s_n to form a permutation π, where $\boldsymbol{v}_{\pi(1)}$ is the most query-relevant patch with respect to $\mathcal{Q}$. Rather than sharing a common supervision pool across all $\boldsymbol{h}_k \in \mathcal{H}$, which would cause all latent tokens to collapse onto the same top-ranked evidence, the ranked tokens are partitioned into non-overlapping chunks, with distinct chunks assigned to each $\boldsymbol{h}_k$. Formally, the positive set $\mathcal{P}_k$ and negative set $\mathcal{N}_k$ for the k-th latent token $\boldsymbol{h}_k \in \mathcal{H}$ are constructed as:

$$\mathcal{P}_k = \{\boldsymbol{v}_{\pi(t)} \mid t \in \mathcal{S}_k^+\}, \qquad \mathcal{N}_k = \{\boldsymbol{v}_{\pi(t)} \mid t \in \mathcal{S}_k^-\}, \tag{3}$$

where $\mathcal{S}_k^+$ and $\mathcal{S}_k^-$ denote the positive and negative index chunks assigned to the k-th latent token, respectively, with all assigned chunks mutually disjoint and with fixed cardinalities $|\mathcal{S}_k^+| = \texttt{pos_num}$ and $|\mathcal{S}_k^-| = \texttt{neg_num}$ for all k. This ensures that each $\boldsymbol{h}_k$ is supervised by a distinct subset of highly relevant tokens while being repelled from a distinct subset of low-relevance content, preserving diversity across $\mathcal{H}$. The latent tokens are then optimized via the following contrastive objective over

$(\mathcal{P}_k, \mathcal{N}_k)$:

$$\mathcal{H}^*_{\text{sft}} = \arg\min_{\mathcal{H}} \left(-\frac{1}{K} \sum_{k=1}^{K} \log \frac{\beta \sum_{\boldsymbol{v} \in \mathcal{P}_k} \exp(\text{sim}(\boldsymbol{h}_k, \boldsymbol{v})/\tau)}{\sum_{\boldsymbol{v} \in \mathcal{P}_k \cup \mathcal{N}_k} \exp(\text{sim}(\boldsymbol{h}_k, \boldsymbol{v})/\tau)} \right), \tag{4}$$

where $\text{sim}(\cdot, \cdot)$ denotes cosine similarity, τ is a temperature hyperparameter, and $\beta = \frac{\texttt{pos_num}+\texttt{neg_num}}{\texttt{pos_num}}$ corrects for the positive/negative set-size imbalance. Stage I optimizes only $\mathcal{H}$ with $\boldsymbol{\theta}$ frozen for N_{sft} steps (set to 5 by default), after which the warmed-up latent state $\mathcal{H}^*_{\text{sft}}$ is passed to Stage II.

3.3 Stage II: Latent-to-Answer Reinforcement

While Stage I improves the semantic quality of $\mathcal{H}$, semantic quality alone does not guarantee that $\mathcal{H}$ is effectively utilized during answer prediction. Stage II therefore targets latent utilization directly, optimizing $\mathcal{H}$ to ensure that each $\boldsymbol{h}_k$ actively contributes to the final answer. To this end, we formulate Stage II as a reward-driven optimization over $\mathcal{H}$. The reward is designed to enforce *reasoning progression*: the output distribution of each $\boldsymbol{h}_k$ should become increasingly concentrated from $\boldsymbol{h}_1$ to $\boldsymbol{h}_K$, reflecting the intuition that latent tokens grow more certain as reasoning advances along $\mathcal{H}$. Formally, the reward over a perturbed candidate $\tilde{\mathcal{H}} = \mathcal{H} + \boldsymbol{\epsilon}$ is defined as:

$$R(\tilde{\mathcal{H}}) = \frac{1}{K-1} \sum_{k=1}^{K-1} \max\left(0,\ \mathcal{E}^{(k)} - \mathcal{E}^{(k+1)}\right), \tag{5}$$

where $\mathcal{E}^{(k)} = -\sum_{j=1}^{\delta} \tilde{p}_j^{(k)} \log \tilde{p}_j^{(k)}$ denotes the top-δ entropy at the k-th latent position, and $\tilde{p}_j^{(k)}$, $j = 1, \ldots, \delta$ are the top-δ predicted token probabilities.[3] This reward increases when the output distribution of each $\boldsymbol{h}_k$ becomes progressively more concentrated from $\boldsymbol{h}_1$ to $\boldsymbol{h}_K$.

Optimization starts from $\mathcal{H}^*_{\text{sft}}$ and proceeds by stochastic local exploration for N_{rl} steps. At each step i, a Gaussian perturbation $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \sigma_i^2 \boldsymbol{I})$ is sampled to form a candidate $\tilde{\mathcal{H}}_i = \mathcal{H}_i + \boldsymbol{\epsilon}_i$, and the latent state is updated according to the NES gradient estimator:

$$\mathcal{H}_{i+1} = \mathcal{H}_i + \frac{\alpha}{\sigma_i^2} \cdot R(\tilde{\mathcal{H}}_i) \cdot \boldsymbol{\epsilon}_i, \tag{6}$$

where α is the step size and σ_i decays geometrically to gradually shrink the exploration radius around $\mathcal{H}_i$. Since the reward trajectory is generally non-monotonic, the best-performing latent state encountered during optimization is retained as $\mathcal{H}^*$ for final answer decoding.

4 Experiments

Implementation Details. All experiments are conducted with frozen MLLM backbones, and only the visual latents are optimized at inference time. Unless otherwise specified, we use Qwen2.5-VL-7B as the default backbone and set the latent reasoning length to $K = 4$. In Stage I, we compute query-guided relevance scores over visual tokens and construct chunk-wise positive and negative patch sets for each latent token. We set the number of positive and negative patches to $\texttt{pos_num} = 2$ and $\texttt{neg_num} = 4$, respectively. The latent warm-up is optimized for $N_{\text{sft}} = 5$ steps using the contrastive latent–visual alignment objective. In Stage II, we initialize from the warmed-up latent state $\mathcal{H}^*_{\text{sft}}$ and perform reward-driven latent optimization for $N_{\text{rl}} = 15$ steps. The reward encourages the predicted token distributions along the latent reasoning sequence to become progressively more concentrated, and the best latent state is retained as $\mathcal{H}^*$ for final answer decoding. The main experiments are run on 2 NVIDIA H100 GPUs, while the efficiency measurements and the additional experiments in Appendix C are run on 2 NVIDIA H200 GPUs. All reported results are evaluated by GPT-4o. The detailed evaluation prompt is provided in Appendix A.3.

Benchmarks & Baselines. We evaluate across eight benchmarks spanning four categories: (1) *Visual perception*: Counting, IQ Test, Relative Reflectance (RR) [7]; (2) *Visual reasoning*: MMVP [29], Hull-Bench [11]; (3) *Comprehensive*: ScienceQA [19], MM-Star [4]; (4) *Mathematical reasoning*: MM-Vista [18]. We compare against two categories of baselines: *non-latent reasoning* methods (MCoT [47], CCoT [21], ICoT [10]) and *latent reasoning* methods that operate in continuous latent space (DMLR [16], Monet [32], CoVT [23], LVR [15]). LVR denotes the model after its supervised stage, and LVR_{RF} the model after its additional reinforcement-learning (RF) stage. We use Qwen2.5-VL-3B/7B [1], R1-OneVision-7B [40], and VLAA-Thinking-7B [3] as MLLM backbones. Details of each benchmark and baseline method are provided in Appendix A.1 and Appendix A.2.

Algorithm 1 Optimization Procedure for Unsilencing Visual Latents

Require: Frozen backbone $\Phi_{\boldsymbol{\theta}}$, multimodal input $(\mathcal{V}, \mathcal{Q})$, initial latent state $\mathcal{H}$, warm-up steps N_{sft}, reinforcement steps N_{rl}
Ensure: Optimized latent state $\mathcal{H}^*$

Stage I: Visual Latent Warm-up
for $n = 1$ **to** N_{sft} **do**
  Compute relevance scores s_n via Eq. (2)
  Construct $\{(\mathcal{P}_k, \mathcal{N}_k)\}_{k=1}^{K}$ via Eq. (3) ▷ `chunk-wise pos/neg assignment`
  Update $\mathcal{H}$ by minimizing Eq. (4)
end for
$\mathcal{H}^*_{\text{sft}} \leftarrow \mathcal{H}$
Stage II: Latent-to-Answer Reinforcement
Initialize $\mathcal{H}_{N_{\text{sft}}} \leftarrow \mathcal{H}^*_{\text{sft}}$, $\mathcal{H}^* \leftarrow \mathcal{H}_{N_{\text{sft}}}$ ▷ `initialize from Stage I output`
for $i = N_{\text{sft}}$ **to** $N_{\text{sft}} + N_{\text{rl}} - 1$ **do**
  Sample $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \sigma_i^2 \boldsymbol{I})$
  Form candidate $\tilde{\mathcal{H}}_i \leftarrow \mathcal{H}_i + \boldsymbol{\epsilon}_i$
  Evaluate reward $R(\tilde{\mathcal{H}}_i)$ via Eq. (5) ▷ `reasoning progression reward`
  Update $\mathcal{H}_{i+1}$ via Eq. (6)
  if $R(\tilde{\mathcal{H}}_i) > R(\mathcal{H}^*)$ **then**
    $\mathcal{H}^* \leftarrow \mathcal{H}_{i+1}$ ▷ `retain best-performing latent state`
  end if
end for
return $\mathcal{H}^*$

Table 1: Comparison of different reasoning methods across multiple benchmarks. Results are reported in accuracy (%), with bracketed values showing absolute gains over the vanilla baseline. Best and second-best results are highlighted with green and lighter green shading, respectively.

Methods	Counting	IQTest	RR	MMVP	Hull-Bench	ScienceQA	MM-Star	MM-Vista
				Qwen2.5VL-7B				
Vanilla	65.00	22.67	38.90	68.67	65.40	82.30	59.30	58.70
MCoT [47]	66.67 [+1.67]	22.67 [+0.00]	35.82 [-3.08]	68.00 [-0.67]	63.62 [-1.78]	83.90 [+1.60]	57.90 [-1.40]	56.40 [-2.30]
CCoT [21]	66.67 [+1.67]	21.33 [-1.34]	43.28 [+4.38]	69.00 [+0.33]	64.88 [-0.52]	83.80 [+1.50]	58.70 [-0.60]	57.80 [-0.90]
ICoT [10]	68.33 [+3.33]	25.33 [+2.66]	37.31 [-1.59]	69.33 [+0.66]	65.51 [+0.11]	78.40 [-3.90]	60.40 [+1.10]	58.90 [+0.20]
DMLR [16]	67.50 [+2.50]	25.33 [+2.66]	34.33 [-4.57]	70.00 [+1.33]	65.83 [+0.43]	83.40 [+1.10]	60.10 [+0.80]	59.10 [+0.40]
Monet [32]	60.83 [-4.17]	20.67 [-2.00]	40.36 [+1.46]	65.67 [-3.00]	65.67 [+0.27]	78.80 [-3.50]	55.50 [-3.80]	62.40 [+3.70]
CoVT [23]	67.50 [+2.50]	30.67 [+8.00]	38.81 [-0.09]	72.67 [+4.00]	64.46 [-0.94]	83.80 [+1.50]	62.00 [+2.70]	64.30 [+5.60]
LVR [15]	68.33 [+3.33]	28.67 [+6.00]	40.30 [+1.40]	69.00 [+0.33]	66.67 [+1.27]	82.80 [+0.50]	60.40 [+1.10]	61.40 [+2.70]
LVR_{RF} [15]	68.33 [+3.33]	27.33 [+4.66]	42.54 [+3.64]	69.67 [+1.00]	65.19 [-0.21]	83.10 [+0.80]	61.80 [+2.50]	62.60 [+3.90]
Ours	68.33 **[+3.33]**	31.33 **[+8.66]**	44.78 **[+5.88]**	73.00 **[+4.33]**	67.30 **[+1.90]**	84.20 **[+1.90]**	61.80 **[+2.50]**	63.70 **[+5.00]**
				Qwen2.5VL-3B				
Vanilla	55.83	24.67	32.84	55.67	64.14	73.50	50.20	48.20
MCoT [47]	55.83 [+0.00]	21.33 [-3.34]	36.57 [+3.73]	54.33 [-1.34]	63.80 [-0.34]	74.10 [+0.60]	48.50 [-1.70]	47.30 [-0.90]
CCoT [21]	54.17 [-1.66]	31.33 [+6.66]	27.61 [-5.23]	55.33 [-0.34]	64.03 [-0.11]	73.90 [+0.40]	49.30 [-0.90]	48.00 [-0.20]
ICoT [10]	55.83 [+0.00]	21.33 [-3.34]	35.82 [+2.98]	56.00 [+0.33]	64.67 [+0.53]	68.90 [-4.60]	49.60 [-0.60]	49.80 [+1.60]
DMLR [16]	52.50 [-3.33]	22.67 [-2.00]	33.58 [+0.74]	56.67 [+1.00]	64.67 [+0.53]	73.80 [+0.30]	51.20 [+1.00]	51.00 [+2.80]
LVR [15]	59.17 [+3.34]	26.67 [+2.00]	35.82 [+2.98]	58.67 [+3.00]	59.31 [-4.83]	73.20 [-0.30]	51.40 [+1.20]	47.50 [-0.70]
LVR_{RF} [15]	60.83 [+5.00]	28.33 [+3.66]	37.31 [+4.47]	58.33 [+2.66]	60.99 [-3.15]	73.70 [+0.20]	51.50 [+1.30]	48.60 [+0.40]
Ours	59.17 **[+3.34]**	32.00 **[+7.33]**	38.06 **[+5.22]**	59.33 **[+3.66]**	64.35 **[+0.21]**	74.30 **[+0.80]**	52.40 **[+2.20]**	50.00 **[+1.80]**

4.1 Main Results

Table 1 compares our method with representative multimodal reasoning baselines across multiple benchmarks. Overall, our method achieves the best or second-best performance on most benchmarks under both Qwen2.5VL-7B and Qwen2.5VL-3B, demonstrating the effectiveness and scalability of the proposed two-stage latent optimization framework. On Qwen2.5VL-7B, our method achieves the best results on six out of eight benchmarks, including Counting, IQTest, RR, MMVP, Hull-

[3]We set $\delta = 10$ by default in all experiments.

Table 2: Ablation study of the two-stage optimization design on Qwen2.5VL-7B.

Stage I	*Stage II*	MMVP	Hull-Bench	ScienceQA	MM-Star	MM-Vista	MMMU	MMMU-Pro
○	○	68.67	65.40	82.30	59.30	58.70	52.54	37.51
✓	○	72.00 [+ 3.33]	66.25 [+ 0.85]	83.40 [+ 1.10]	62.40 [+ 3.10]	62.10 [+ 3.40]	53.01 [+ 0.47]	37.05 [− 0.46]
✓	✓	73.00 [+ 4.33]	67.30 [+ 1.90]	84.20 [+ 1.90]	61.80 [+ 2.50]	63.70 [+ 5.00]	54.31 [+ 1.77]	38.15 [+ 0.64]

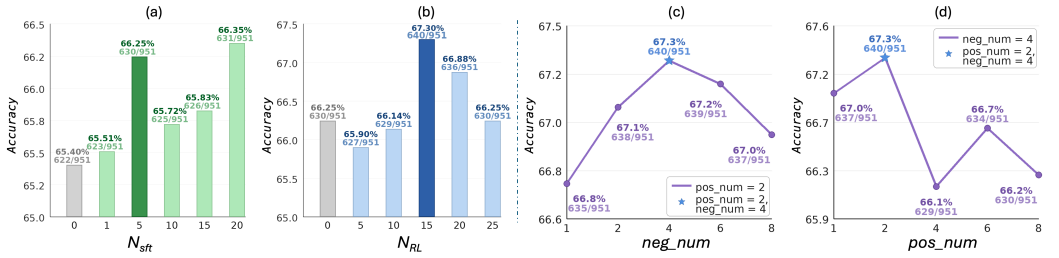

Figure 5: Ablation studies on Hull-Bench. (a,b) Effects of the Stage-I and Stage-II optimization steps, N_{sft} and N_{rl}, respectively. Gray bars denote the baseline without the corresponding component, and darker bars denote the final setting; we use $N_{\text{sft}} = 5$ and $N_{\text{rl}} = 15$ for the accuracy–efficiency trade-off. (c,d) Effects of the negative and positive patch numbers, with `pos_num`=2 and `neg_num`=4 fixed, respectively. The results favor compact positive sets with moderate negative supervision for Stage-I latent warm-up.

Bench, and ScienceQA, while obtaining competitive second-best performance on MM-Star and MM-Vista. Compared with the vanilla baseline, it brings consistent gains across all benchmarks, with particularly notable improvements on IQTest (+8.66), RR (+5.88), MMVP (+4.33), and MM-Vista (+5.00). Compared with non-latent reasoning methods (MCoT [47], CCoT [21], ICoT [10]), our method achieves consistent improvements across all benchmarks, whereas verbal chain-of-thought methods occasionally degrade below the vanilla baseline on fine-grained visual tasks, suggesting that textual reasoning alone is insufficient for these challenges. Against latent reasoning baselines, our method also shows clear advantages on perception-intensive benchmarks such as RR, MMVP, and Hull-Bench. Unlike DMLR [16], which relies on dynamically retrieved visual patches at inference time, or training-based methods (Monet [32], CoVT [23], LVR [15]) prone to visual input shortcuts, our method optimizes latent representations directly without parameter updates or visual re-injection, ensuring the latents serve as genuine reasoning substrates. On Qwen2.5VL-3B, similar trends hold despite the smaller backbone capacity, with our method achieving best or second-best results on all eight benchmarks, demonstrating generalizability across model scales. These gains are statistically robust: across three random seeds the standard deviation is at most 0.69 on every benchmark, and the advantage persists under sampling-based decoding (Appendix C.3). On the expert-knowledge benchmarks MMMU [44] and MMMU-Pro [45], our method also achieves the best accuracy (+1.77 and +0.64), while all other latent reasoning methods fall below Vanilla on MMMU-Pro (Appendix C.2).

4.2 Ablation Study

Two-Stage Optimization Design. We analyze the contribution of each stage and the allocation of optimization steps. As shown in Table 2, Stage I improves MMVP from 68.67 to 72.00 and Hull-Bench from 65.40 to 66.25, indicating the benefit of enhancing latent semantics. Adding Stage II further raises performance to 73.00 and 67.30, respectively, confirming that Stage I and Stage II are largely complementary: the former improves latent quality, while the latter promotes answer-time utilization. On the expert-knowledge benchmarks MMMU and MMMU-Pro, Stage II adds +1.30 and +1.10 on top of Stage I; on MMMU-Pro, it is Stage II that turns the overall gain positive. We also vary the optimization steps in both stages. As shown in Figure 5 (a) and (b), on Hull-Bench, increasing N_{sft} brings moderate gains, while performance is more sensitive to N_{rl}, peaking at $N_{\text{rl}} = 15$ before declining, suggesting that over-optimization in Stage II may hurt the quality of latent-guided decoding. We therefore set $N_{\text{sft}} = 5$ and $N_{\text{rl}} = 15$ for a balance between effectiveness and efficiency. Stage II is also robust to its NES hyperparameters: over a broad range of perturbation scales $\sigma \in [0.05, 0.2]$ and step sizes $\alpha \in [0.0005, 0.002]$, the reward rises nearly monotonically and accuracy varies only marginally across seeds (Appendix C.4).

Length of Latent Reasoning. We study the effect of latent reasoning length K and compare with LVR_{RF} [15]. As shown in Table 3, our method yields roughly consistent performance gains as K increases, with MMVP improving from 72.33 to 73.67 and Hull-Bench from 66.67 to 69.09 as

Table 3: Ablation on latent reasoning length on MMVP and Hull-Bench. Our method benefits more from longer latent reasoning, while LVR_{RF} [15] shows weaker utilization.

Latents Length K	MMVP		Hull-Bench	
	Ours	LVR_{RF} [15]	Ours	LVR_{RF} [15]
0	68.67		65.40	
2	72.33 **[+3.66]**	70.33 [+1.66]	66.67 [+1.27]	64.46 [-0.94]
4	73.00 **[+4.33]**	71.33 [+2.66]	67.30 [+1.90]	65.19 [-0.21]
6	71.67 [+3.00]	71.67 [+3.00]	68.71 **[+3.31]**	65.19 [-0.21]
8	73.33 **[+4.66]**	71.67 [+3.00]	68.98 **[+3.58]**	65.40 [+0.00]
10	73.67 **[+5.00]**	71.67 [+3.00]	69.09 **[+3.69]**	65.51 [+0.11]

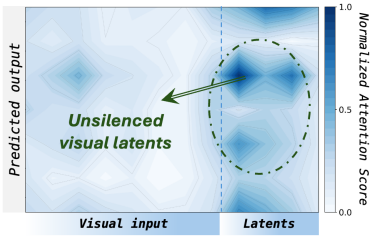

Figure 6: Attention visualization on an MMVP sample. Optimized latents receive more focused attention than visual inputs, reducing shortcuts compared with Figure 3.

Table 4: Comparison of different reasoning methods on VLAA Thinking-7B and R1 OneVision-7B. Results are reported in accuracy (%), with bracketed values showing absolute gains over the vanilla baseline. Best and second-best results are highlighted with green and lighter green shading, respectively.

Methods	Counting	IQTest	RR	MMVP	Hull-Bench	ScienceQA	MM-Star	MM-Vista
VLAA Thinking-7B								
Vanilla	65.00	26.67	32.84	68.30	62.00	84.50	58.90	61.10
MCoT [47]	62.50 [-2.50]	22.00 [-4.67]	32.84 [+0.00]	67.20 [-1.10]	62.80 [+0.80]	84.70 [+0.20]	57.10 [-1.80]	59.60 [-1.50]
CCoT [21]	68.33 [+3.33]	26.00 [-0.67]	34.33 [+1.49]	68.00 [-0.30]	64.60 [+2.60]	84.60 [+0.10]	59.00 [+0.10]	60.50 [-0.60]
ICoT [10]	60.83 [-4.17]	27.33 [+0.66]	32.84 [+0.00]	68.30 [+0.00]	65.90 [+3.90]	83.80 [-0.70]	58.20 [-0.70]	61.40 [+0.30]
DMLR [16]	64.17 [-0.83]	25.00 [-1.67]	35.07 [+2.23]	69.40 [+1.10]	67.90 [+5.90]	85.00 [+0.50]	59.20 [+0.30]	62.90 [+1.80]
Ours	68.33 **[+3.33]**	26.67 **[+0.00]**	35.82 **[+2.98]**	73.67 **[+5.37]**	67.93 **[+5.93]**	85.40 **[+0.90]**	62.70 **[+3.80]**	65.70 **[+4.60]**
R1 OneVision-7B								
Vanilla	59.17	22.67	29.85	67.00	62.10	85.60	52.10	51.20
MCoT [47]	58.33 [-0.84]	28.00 [+5.33]	30.60 [+0.75]	68.00 [+1.00]	62.50 [+0.40]	86.50 [+0.90]	51.60 [-0.50]	52.50 [+1.30]
CCoT [21]	60.00 [+0.83]	21.33 [-1.34]	33.58 [+3.73]	68.90 [+1.90]	63.00 [+0.90]	86.30 [+0.70]	53.50 [+1.40]	53.40 [+2.20]
ICoT [10]	59.17 [+0.00]	22.67 [+0.00]	29.85 [+0.00]	69.60 [+2.60]	63.80 [+1.70]	85.50 [-0.10]	54.00 [+1.90]	55.60 [+4.40]
DMLR [16]	54.17 [-5.00]	25.33 [+2.66]	35.07 [+5.22]	71.90 [+4.90]	64.10 [+2.00]	83.50 [-2.10]	56.20 [+4.10]	58.00 [+6.80]
Ours	61.67 **[+2.50]**	36.67 **[+14.00]**	37.30 **[+7.45]**	71.00 **[+4.00]**	69.51 **[+7.41]**	86.40 **[+0.80]**	56.80 **[+4.70]**	57.30 **[+6.10]**

K grows from 2 to 10, demonstrating that our method can effectively leverage additional latent reasoning capacity. In contrast, LVR_{RF} [15] shows little sensitivity to K, saturating early on MMVP and improving only marginally on Hull-Bench, which further evidences the underutilization of latent reasoning in LVR. More results on additional datasets are provided in Appendix B.1.

Contrastive Patch Configuration. We analyze the effect of positive and negative patch allocation in Stage I. As shown in Figure 5 (c), performance increases steadily from 66.8% to 67.3% as `neg_num` grows from 1 to 4, then slightly declines, indicating that too few negatives provide insufficient contrastive signal while too many may introduce noise. As shown in (d), `pos_num` has a more pronounced effect: accuracy peaks at `pos_num` $= 2$ (67.3%) and drops notably at `pos_num` $= 4$ (66.1%), suggesting that enlarging the positive set dilutes the alignment signal with less relevant patches. We therefore set `pos_num` $= 2$ and `neg_num` $= 4$.

Attention Visualization. We visualize the attention patterns on an MMVP sample in Figure 6. As expected, the optimized latents attract substantially stronger attention from the predicted outputs compared to the raw visual input tokens, and attend more to task-relevant visual regions. This confirms that the two-stage optimization effectively unsilences the visual latents, enabling them to serve as genuine reasoning substrates rather than being bypassed. More visualizations are provided in Appendix B.2.

Latent Intervention. To verify that the unsilenced latents are causally used, we intervene on the latent tokens by removing them (Knockout), injecting Gaussian noise, or swapping them with those of a mismatched sample. As shown in Table 6, the jointly trained LVR and LVR_{RF} are nearly insensitive to all interventions ($|\Delta| \leq 1.00$), and knockout even slightly improves their accuracy, directly confirming that their latents are silenced and their gains mainly stem from a reinforced shortcut to the raw visual input. In contrast, our accuracy drops by 1.90–5.26 on Hull-Bench and 4.33–7.00 on MMVP, showing that the optimized latents actively drive answer prediction.

Efficiency. We measure the efficiency ratio of each method as performance gain relative to output token count (details in Appendix B.3). As shown in Figure 7, our method achieves the highest ratio (+0.75), surpassing LVR (+0.62), CoVT (+0.54), and DMLR (+0.18), while non-latent methods cluster near zero and Monet yields a negative ratio (-0.32). This demonstrates that our method

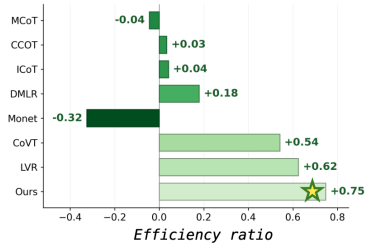

Figure 7: Efficiency ratio of different methods on MMVP. Higher values indicate a better performance-efficiency balance. Calculation details are provided in Appendix B.3.

Table 5: Inference overhead on MMVP (Qwen2.5-VL-7B, 2×H200). TTFT: time to the first answer token, including latent optimization; WCT: end-to-end time per sample; Mem: peak GPU memory.

Method	Training-free	TTFT (s)↓	WCT (s)↓	Mem (GB)↓	Acc.↑
Vanilla	✓	0.24	2.92	31.18	68.67
MCoT	✓	2.95	4.12	31.28	68.00
CCoT	✓	3.38	4.56	31.35	69.00
ICoT	✓	0.27	2.86	31.26	69.33
DMLR	✓	6.67	7.87	48.09	70.00
Monet	✗	0.44	2.30	31.24	65.67
CoVT	✗	0.24	1.58	31.18	72.67
LVR	✗	0.20	0.64	31.14	69.00
LVR_{RF}	✗	0.20	0.64	31.14	69.67
Ours	✓	2.56	3.72	36.10	**73.00**

Table 6: Latent-intervention sensitivity (Qwen2.5-VL-7B). Entries report Δ = Acc(intervened) − Acc(original).

Method	Hull-Bench					MMVP				
	Acc.	Knockout	Noise σ=0.5	Noise σ=1.0	Swap	Acc.	Knockout	Noise σ=0.5	Noise σ=1.0	Swap
LVR	66.67	+0.21	+0.32	+0.53	+0.42	69.00	+0.33	−0.34	−0.34	−0.34
LVR_{RF}	65.19	+0.42	−0.42	−0.42	+0.11	69.67	+0.33	−1.00	−0.67	0.00
Ours	67.30	−1.90	−3.36	−3.47	−5.26	73.00	−4.33	−5.00	−5.67	−7.00

delivers superior performance gains with competitive token efficiency. The comparison is fair: the optimization operates on continuous latents and generates no tokens. Since the visual input and the query are unchanged, their KV cache is computed once and reused across all optimization steps. Thus, as shown in Table 5, our method adds only about 0.8 s per sample over Vanilla, comparable to CoT-style baselines and well below the test-time method DMLR in both time and memory, without any training (see Appendix C.1 for Hull-Bench).

Generalization to Various Models.[4] We evaluate our two-stage latent optimization framework on VLAA Thinking-7B and R1 OneVision-7B. As shown in Table 4, our method consistently improves over the vanilla baseline and achieves the best or second-best results on all benchmarks. Notably, it improves MMVP, Hull-Bench, and MM-Vista by +5.37, +5.93, and +4.60 on VLAA Thinking-7B, and IQTest, RR, and Hull-Bench by +14.00, +7.45, and +7.41 on R1 OneVision-7B. These results demonstrate the model-agnostic effectiveness of our framework across diverse MLLMs.

5 Conclusion

We identify Silenced Visual Latents, a phenomenon where visual latents become semantically enhanced but are underutilized during answer prediction. To address this, we propose a frozen-backbone two-stage latent optimization framework: Stage I improves latent quality through query-guided contrastive latent–visual alignment, while Stage II enhances latent utilization via reward-driven latent-to-answer reinforcement. Experiments across multiple benchmarks and model backbones show consistent improvements, demonstrating that disentangling latent quality and latent utilization is effective for multimodal reasoning.

6 Limitations

Our framework currently targets image-based visual reasoning tasks, and we look forward to exploring its potential in richer multimodal settings such as video understanding and embodied perception in future work. We also expect that scaling to larger and more capable backbone models will further amplify the benefits observed in our experiments.

Acknowledgments and Disclosure of Funding

This research is supported by Xin Zhang's A*STAR Career Development Fund (CDF) (Project No. H26-KSR0051). The authors declare no competing interests.

[4] Monet [32], CoVT [23], and LVR [15] are excluded as they fundamentally alter the model's reasoning behavior via training, making them incompatible with these reasoning-capable backbones.

References

[1] 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, 2025.

[2] Sule Bai, Mingxing Li, Yong Liu, Jing Tang, Haoji Zhang, Lei Sun, Xiangxiang Chu, and Yansong Tang. Univg-r1: Reasoning guided universal visual grounding with reinforcement learning. *arXiv preprint arXiv:2505.14231*, 2025.

[3] Hardy Chen, Haoqin Tu, Fali Wang, Hui Liu, Xianfeng Tang, Xinya Du, Yuyin Zhou, and Cihang Xie. Sft or rl? an early investigation into training r1-like reasoning large vision-language models, 2025.

[4] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Are we on the right way for evaluating large vision-language models? *arXiv preprint arXiv:2403.20330*, 2024.

[5] Zhangquan Chen, Manyuan Zhang, Xinlei Yu, Xufang Luo, Mingze Sun, Zihao Pan, Yan Feng, Peng Pei, Xunliang Cai, and Ruqi Huang. Think with 3d: Geometric imagination grounded spatial reasoning from limited views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

[6] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24185–24198, 2024.

[7] 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. *arXiv preprint arXiv:2404.12390*, 2024.

[8] Xingyu Fu, Minqian Liu, Zhengyuan Yang, John Richard Corring, Yijuan Lu, Jianwei Yang, Dan Roth, Dinei Florencio, and Cha Zhang. Refocus: Visual editing as a chain of thought for structured image understanding. In *International Conference on Machine Learning*, pages 17783–17805. PMLR, 2025.

[9] Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, Zhengyang Tang, Benyou Wang, Daoguang Zan, Shanghaoran Quan, Ge Zhang, Lei Sha, Yichang Zhang, Xuancheng Ren, Tianyu Liu, and Baobao Chang. Omni-MATH: A universal olympiad level mathematic benchmark for large language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

[10] Jun Gao, Yongqi Li, Ziqiang Cao, and Wenjie Li. Interleaved-modal chain-of-thought, 2025.

[11] Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, Dinesh Manocha, and Tianyi Zhou. Hallusionbench: An advanced diagnostic suite for entangled language hallucination & visual illusion in large vision-language models, 2023.

[12] 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.

[13] Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, and Ranjay Krishna. Visual sketchpad: Sketching as a visual chain of thought for multimodal language models. *Advances in Neural Information Processing Systems*, 37:139348–139379, 2024.

[14] Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

[15] 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*, 2026.

[16] Chengzhi Liu, Yuzhe Yang, Yue Fan, Qingyue Wei, Sheng Liu, and Xin Eric Wang. Reasoning within the mind: Dynamic multimodal interleaving in latent space. *arXiv preprint arXiv:2512.12623*, 2025.

[17] Luyang Liu, Jonas Pfeiffer, Jiaxing Wu, Jun Xie, and Arthur Szlam. Deliberation in latent space via differentiable cache augmentation. In *Forty-second International Conference on Machine Learning*, 2025.

[18] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *International Conference on Learning Representations (ICLR)*, 2024.

[19] Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[20] Jizheng Ma, Xiaofei Zhou, Geyuan Zhang, Yanlong Song, and Han Yan. Multimodal reasoning via latent refocusing. *arXiv preprint arXiv:2511.02360*, 2025.

[21] Chancharik Mitra, Brandon Huang, Trevor Darrell, and Roei Herzig. Compositional chain-of-thought prompting for large multimodal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14420–14431, June 2024.

[22] Mihir Prabhudesai, Lili Chen, Alex Ippoliti, Katerina Fragkiadaki, Hao Liu, and Deepak Pathak. Maximizing confidence alone improves reasoning. *arXiv preprint arXiv:2505.22660*, 2025.

[23] 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. *arXiv preprint arXiv:2511.19418*, 2025.

[24] 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. *Advances in Neural Information Processing Systems*, 37:8612–8642, 2024.

[25] 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*, pages 677–693, 2025.

[26] Dachuan Shi, Abedelkadir Asi, Keying Li, Xiangchi Yuan, Leyan Pan, Wenke Lee, and Wen Xiao. Swireasoning: Switch-thinking in latent and explicit for pareto-superior reasoning LLMs. In *The Fourteenth International Conference on Learning Representations*, 2026.

[27] Wenhui Tan, Jiaze Li, Jianzhong Ju, Zhenbo Luo, Ruihua Song, and Jian Luan. Think silently, think fast: Dynamic latent compression of llm reasoning chains. *Advances in Neural Information Processing Systems*, 2025.

[28] Wei Tang, Yanpeng Sun, Qinying Gu, and Zechao Li. Visual position prompt for mllm based visual grounding. *IEEE Transactions on Multimedia*, 2026.

[29] Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms, 2024.

[30] Chenxi Wang, Xiang Chen, Ningyu Zhang, Bozhong Tian, Haoming Xu, Shumin Deng, and Huajun Chen. MLLM can see? dynamic correction decoding for hallucination mitigation. In *The Thirteenth International Conference on Learning Representations*, 2025.

[31] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021.

[32] Qixun Wang, Yang Shi, Yifei Wang, Yuanxing Zhang, Pengfei Wan, Kun Gai, Xianghua Ying, and Yisen Wang. Monet: Reasoning in latent visual space beyond images and language. *arXiv preprint arXiv:2511.21395*, 2025.

[33] Weixing Wang, Zifeng Ding, Jindong Gu, Rui Cao, Christoph Meinel, Gerard de Melo, and Haojin Yang. Image tokens matter: Mitigating hallucination in discrete tokenizer-based large vision-language models via latent editing. *arXiv preprint arXiv:2505.21547*, 2025.

[34] Yaoting Wang, Shengqiong Wu, Yuecheng Zhang, Shuicheng Yan, Ziwei Liu, Jiebo Luo, and Hao Fei. Multimodal chain-of-thought reasoning: A comprehensive survey. *arXiv preprint arXiv:2503.12605*, 2025.

[35] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[36] Yixuan Weng, Minjun Zhu, Qiujie Xie, QiYao Sun, Zhen Lin, Sifan Liu, and Yue Zhang. Deepscientist: Advancing frontier-pushing scientific findings progressively. In *The Fourteenth International Conference on Learning Representations*, 2026.

[37] Wenshan Wu, Shaoguang Mao, Yadong Zhang, Yan Xia, Li Dong, Lei Cui, and Furu Wei. Mind's eye of llms: visualization-of-thought elicits spatial reasoning in large language models. *Advances in Neural Information Processing Systems*, 37:90277–90317, 2024.

[38] Zhifei Xie, Ziyang Ma, Zihang Liu, Kaiyu Pang, Hongyu Li, Jialin Zhang, Yue Liao, Deheng Ye, Chunyan Miao, and Shuicheng Yan. Mini-omni-reasoner: Token-level thinking-in-speaking in large speech models. *arXiv preprint*, 2025.

[39] Zhongxing Xu, Zhonghua Wang, Zhe Qian, Dachuan Shi, Feilong Tang, Ming Hu, Shiyan Su, Xiaocheng Zou, Wei Feng, Dwarikanath Mahapatra, Yifan Peng, Mingquan Lin, and Zongyuan Ge. Thinking in uncertainty: Mitigating hallucinations in mlrms with latent entropy-aware decoding. *arXiv preprint arXiv:2603.13366*, 2026.

[40] Yi Yang, Xiaoxuan He, Hongkun Pan, Xiyan Jiang, Yan Deng, Xingtao Yang, Haoyu Lu, Dacheng Yin, Fengyun Rao, Minfeng Zhu, Bo Zhang, and Wei Chen. R1-onevision: Advancing generalized multimodal reasoning through cross-modal formalization, 2025.

[41] Zeyuan Yang, Xueyang Yu, Delin Chen, Maohao Shen, and Chuang Gan. Machine mental imagery: Empower multimodal reasoning with latent visual tokens. *arXiv preprint arXiv:2506.17218*, 2025.

[42] Jiacheng Ye, Shansan Gong, Liheng Chen, Lin Zheng, Jiahui Gao, Han Shi, Chuan Wu, Xin Jiang, Zhenguo Li, Wei Bi, et al. Diffusion of thought: Chain-of-thought reasoning in diffusion language models. *Advances in Neural Information Processing Systems*, 37:105345–105374, 2024.

[43] Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A survey on multimodal large language models. *National Science Review*, 11(12):nwae403, 2024.

[44] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. 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*, 2024.

[45] Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, et al. MMMU-Pro: A more robust multi-discipline multimodal understanding benchmark. *arXiv preprint arXiv:2409.02813*, 2024.

[46] Jusheng Zhang, Kaitong Cai, Xiaoyang Guo, Sidi Liu, Qinhan Lv, Ruiqi Chen, Jing Yang, Yijia Fan, Xiaofei Sun, Jian Wang, et al. Mm-cot: a benchmark for probing visual chain-of-thought reasoning in multimodal models. *arXiv preprint arXiv:2512.08228*, 2025.

[47] Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models. *Transactions on Machine Learning Research*, 2024.

[48] Xueliang Zhao, Wei Wu, Jian Guan, and Lingpeng Kong. Promptcot: Synthesizing olympiad-level problems for mathematical reasoning in large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 18167–18188, 2025.

[49] Qiji Zhou, Ruochen Zhou, Zike Hu, Panzhong Lu, Siyang Gao, and Yue Zhang. Image-of-thought prompting for visual reasoning refinement in multimodal large language models. *arXiv preprint arXiv:2405.13872*, 2024.

[50] Rui-Jie Zhu, Tianhao Peng, Tianhao Cheng, Xingwei Qu, Jinfa Huang, Dawei Zhu, et al. A survey on latent reasoning. *arXiv preprint arXiv:2507.06203*, 2025.

[51] Yicheng Zou, Dongsheng Zhu, Lin Zhu, Tong Zhu, Yunhua Zhou, Peiheng Zhou, Xinyu Zhou, Dongzhan Zhou, Zhiwang Zhou, Yuhao Zhou, et al. Intern-s1-pro: Scientific multimodal foundation model at trillion scale. *arXiv preprint arXiv:2603.25040*, 2026.

A Detailed Experimental Settings

A.1 Benchmarks

Counting [7]. This benchmark evaluates the MLLM abilities in detection, recognition, and compositional reasoning in complex scenes where objects may overlap, be occluded, or have varying sizes. It contains 120 test samples.

IQ Test [7]. This benchmark evaluates the ability of MLLMs to engage in graphical reasoning, without requiring any domain-specific knowledge. It contains 150 test samples.

Relative Reflectance [7]. This benchmark requires comparing the reflectance (albedo) of two pixels, where MLLMs need an understanding of material properties and their interaction with light. It contains 134 test samples.

MMVP [29]. This benchmark is specially crafted to measure MLLM's visual capability via VQA, which consists of "CLIP-blind pairs" to reveal systematic visual shortcomings of MLLMs. It contains 300 test samples.

Hull-Bench [11]. This image-context reasoning benchmark evaluates MLLMs on their susceptibility to hallucination, visual illusion, and logical inconsistency through carefully designed questions. It contains 951 test samples.

ScienceQA [19]. This benchmark consists of multi-modal multiple-choice questions covering diverse scientific subjects, featuring rich visual and textual contexts alongside lecture materials and explanations to assess both answer accuracy and reasoning quality. We evaluate on the first 1,000 samples of the test split.

MM-Star [4]. This vision-centric benchmark ensures visual dependency, minimal data leakage, and the requirement for advanced multi-modal capabilities. We evaluate on the first 1,000 samples of the test split.

MM-Vista [18]. This benchmark consolidates mathematical reasoning within visual contexts, encompassing diverse and complex challenges in visual perception and mathematical reasoning. It contains 1,000 test samples.

MMMU [44]. This benchmark evaluates college-level, multi-discipline multimodal understanding and reasoning that requires expert knowledge. We evaluate on the validation split, which contains 847 multiple-choice questions.

MMMU-Pro [45]. This benchmark is a more robust and difficult variant of MMMU that filters text-only answerable questions and augments the candidate options. We use the standard 10-option setting with 1,730 questions.

A.2 Reasoning Baselines

MCoT (Multi-modal CoT) [47] extends CoT reasoning to multi-modal settings by first generating a reasoning rationale conditioned on multi-modal inputs, then combining it with the original input to produce the final answer.

CCoT (Compositional CoT) [21] enables multi-modal reasoning by converting the visual inputs into a JSON-formatted scene graph, based on which the reasoning is performed.

ICoT (Interleaved CoT) [10] performs visual-text interleaved CoT by dynamically injecting visual clues during the reasoning process, which is triggered when specific tokens appear during reasoning.

LVR [15] is a latent reasoning method that enables MLLMs to perform autoregressive reasoning directly in the visual embedding space. Specifically, the model is trained to generate latent states to reconstruct query-relevant visual tokens, which are interleaved with standard text generation to enhance fine-grained visual understanding. LVR denotes the model after its supervised stage, and LVR_{RF} denotes the model after its additional reinforcement-learning (RF) stage.

Monet [32] trains MLLMs to perform latent visual reasoning by learning to dynamically interleave text generation with latent thinking, where the model autonomously decides when to emit a `<latent>` token to initiate visual reasoning in the embedding space.

You are an answer equivalence judge for multiple choice questions. Determine if the candidate answer matches the ground truth answer.

Question (with options):
{question}

Ground Truth: {ground_truth}

Candidate Answer: {answer}

Rules (in priority order):
1. FIRST check option letters: If candidate answer contains an option letter (A/B/C/D or a/b/c/d), it MUST match the ground truth's option letter.
- Example: Ground truth "(a) A historical site", answer "(c)" -> FALSE (c != a)
- Example: Ground truth "(a) A historical site", answer "(a)" -> TRUE (a = a)

2. If candidate answer has NO option letter, compare the content:
- Example: Ground truth "(d) Dawn", answer "Dawn" -> TRUE (content matches)
- Example: Ground truth "(b) Grenada", answer "Grenada" -> TRUE (content matches)

3. Be STRICT about option letters: "(c)" answering "(a): content" is WRONG, even if content seems related.

Is the candidate answer correct? Respond with ONLY "True" or "False".

Figure 8: GPT-4o judging prompt used for answer evaluation. The prompt asks the evaluator to compare the model prediction with the ground-truth answer and judge semantic correctness.

CoVT [23] is a training-based latent reasoning method that interleaves various continuous visual tokens into reasoning chains, including complementary perceptual cues such as segmentation, depth, and edges. Consequently, the model learns to selectively generate different types of visual tokens without requiring reinforcement learning.

DMLR [16] is a test-time latent reasoning framework that iteratively optimizes latent tokens and updates visual context through confidence-guided policy gradient optimization, dynamically injecting the most relevant visual patches into the reasoning process at inference time.

LEAD [39] is a training-free latent entropy-aware decoding method that mitigates hallucinations in multimodal reasoning models by performing next-state prediction over the full token distribution carried by the hidden states. We compare with LEAD in Appendix C.7.

A.3 Evaluation Prompt

Following DMLR [16], we use a fixed GPT-4o judging prompt for all GPT-based evaluations. The evaluator receives the question, the ground-truth answer, and the model prediction, and is asked to determine whether the prediction is semantically correct. This design avoids brittle string-matching and handles equivalent expressions (e.g., equivalent numerical forms or paraphrased answers) gracefully. The full prompt is shown in Figure 8.

B More Ablation Study

B.1 Length of Latent Reasoning

We further evaluate the effect of latent reasoning length on ScienceQA, MM-Star, and MM-Vista. As shown in Table 7, our method consistently benefits from longer latent reasoning across all three benchmarks. For example, as the latent length increases from $K = 2$ to $K = 10$, performance improves from 83.40 to 85.20 on ScienceQA, from 61.30 to 62.60 on MM-Star, and from 63.00 to 64.30 on MM-Vista. In contrast, LVR exhibits weaker and less stable gains: it gradually drops on ScienceQA, saturates early on MM-Star, and shows only marginal variation on MM-Vista. These results further support that our method can more effectively exploit extended latent reasoning and translate additional latent capacity into improved answer prediction.

Table 7: Ablation on latent reasoning length on ScienceQA, MM-Star, and MM-Vista. Our method benefits more from longer latent reasoning, while LVR shows weaker utilization.

Latents Length K	ScienceQA		MM-Star		MM-Vista	
	Ours	LVR_{RF} [15]	Ours	LVR_{RF} [15]	Ours	LVR_{RF} [15]
0	82.30		59.30		58.70	
2	83.40 [+1.10]	83.60 [+1.30]	61.30 [+2.00]	62.40 [+3.10]	63.00 [+4.30]	62.40 [+3.70]
4	84.20 [+1.90]	83.10 [+0.80]	61.80 [+2.50]	61.80 [+2.50]	63.70 [+5.00]	62.60 [+3.90]
6	84.90 [+2.60]	83.30 [+1.00]	62.00 [+2.70]	61.50 [+2.20]	63.80 [+5.10]	62.20 [+3.50]
8	85.10 [+2.80]	83.20 [+0.90]	62.20 [+2.90]	61.90 [+2.60]	64.10 [+5.40]	62.70 [+4.00]
10	85.20 [+2.90]	82.70 [+0.40]	62.60 [+3.30]	61.90 [+2.60]	64.30 [+5.60]	62.60 [+3.90]

B.2 Additional Attention Visualizations

We provide more attention visualizations in Figure 9 across Hull-Bench, ScienceQA, and MM-Vista. The optimized visual latents consistently receive concentrated attention during prediction, indicating that the model actively leverages the latent reasoning span rather than diffusely relying on raw visual inputs. These results further support that our two-stage optimization framework improves not only latent quality, but also their effective utilization in answer generation.

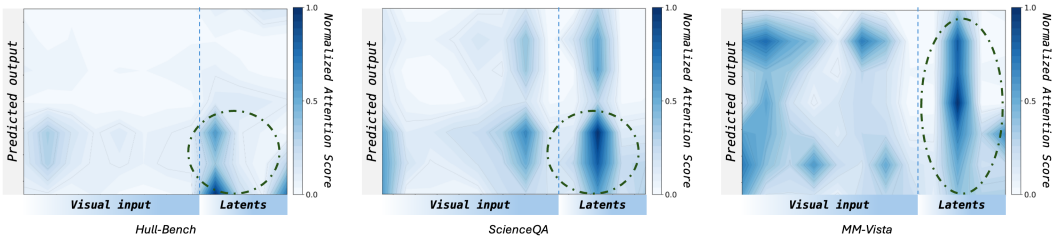

Figure 9: More attention visualizations.

B.3 Efficiency Ratio Calculation

We define the efficiency ratio to jointly measure the performance gain and token efficiency of each method. Specifically, let Δ_m denote the average accuracy improvement of method m over the vanilla baseline across all benchmarks, and let T_m denote the average number of output tokens generated per sample. The efficiency ratio is defined as:

$$\text{Efficiency Ratio} = \frac{\Delta_m}{T_m} \times 10 \tag{7}$$

where the factor of 10 is applied for readability. A higher efficiency ratio indicates that the method achieves greater performance gains per unit of output token generation. Methods with negative ratios generate more tokens while degrading performance relative to the vanilla baseline. For our method, the inference-time optimization operates on continuous latent vectors and generates no tokens; answer decoding runs only once after the optimization finishes, so T_m counts exactly the tokens of the final output, consistent with all other methods. The wall-clock cost of the optimization is reported separately in Appendix C.1.

C Additional Experiments and Analyses

Unless otherwise specified, all experiments in this section use Qwen2.5-VL-7B with the default settings in Section 4.

C.1 Computational Overhead

Although our method optimizes the latents at inference time, its overhead remains moderate for three reasons. (i) The optimization is applied only to the K latent tokens while the visual input and the

query remain unchanged, so their prefill is performed only once and the KV cache is reused across all optimization steps; each step only recomputes the latent tokens, which is far cheaper than prefilling the full multimodal sequence. (ii) The optimization operates on continuous vectors and generates no tokens; decoding runs only once after the optimization finishes, and the high token efficiency of our method (Figure 7) yields shorter generation, offsetting much of the optimization cost. (iii) Unlike Monet, CoVT, and LVR, our method requires no training at all. We report TTFT, end-to-end wall-clock time (WCT) per sample, and peak GPU memory on MMVP and Hull-Bench in Table 8, measured on the same hardware for all methods (2×NVIDIA H200, fp32, 2 workers). Our method adds only about 1 s of WCT over Vanilla on both benchmarks, comparable to CoT-style baselines and far below DMLR, the other test-time method, in both time and memory.

Table 8: Computational overhead on MMVP and Hull-Bench. TTFT: time to the first token of the final answer; WCT: end-to-end time per sample; Mem: peak GPU memory; Δ: absolute accuracy gain over Vanilla.

Method	Training	MMVP				Hull-Bench			
		TTFT (s)↓	WCT (s)↓	Mem (GB)↓	Acc. (Δ)↑	TTFT (s)↓	WCT (s)↓	Mem (GB)↓	Acc. (Δ)↑
Vanilla	None	0.24	2.92	31.18	68.67 (–)	0.29	3.39	31.38	65.40 (–)
MCoT	None	2.95	4.12	31.28	68.00 (−0.67)	3.37	4.39	31.61	63.62 (−1.78)
CCoT	None	3.38	4.56	31.35	69.00 (+0.33)	4.37	6.15	32.74	64.88 (−0.52)
ICoT	None	0.27	2.86	31.26	69.33 (+0.66)	2.32	3.39	31.38	65.51 (+0.11)
DMLR	None	6.67	7.87	48.09	70.00 (+1.33)	8.50	10.16	55.91	65.83 (+0.43)
Monet	Required	0.44	2.30	31.24	65.67 (−3.00)	0.56	2.56	31.21	65.67 (+0.27)
CoVT	Required	0.24	1.58	31.18	72.67 (+4.00)	0.30	1.62	31.24	64.46 (−0.94)
LVR	Required	0.20	0.64	31.14	69.00 (+0.33)	0.50	1.64	31.23	66.67 (+1.27)
LVR_{RF}	Required	0.20	0.64	31.14	69.67 (+1.00)	0.50	1.64	31.23	65.19 (−0.21)
Ours	**None**	2.56	3.72	36.10	**73.00 (+4.33)**	3.06	4.35	38.63	**67.30 (+1.90)**

C.2 Generalization to Expert-Knowledge Benchmarks

Since Stage I aligns latents with fine-grained local visual chunks and Stage II encourages sharper predictions, a natural concern is whether the method generalizes beyond perception-grounded reasoning. We therefore evaluate on MMMU [44] (validation split, 847 multiple-choice questions) and its more difficult variant MMMU-Pro [45] (standard 10-option setting, 1,730 questions). As shown in Table 9, our method remains robust as the task shifts from perception toward abstract, multi-disciplinary reasoning, achieving the best accuracy on both benchmarks (+1.77 on MMMU and +0.64 on MMMU-Pro). Notably, on MMMU-Pro all other latent reasoning methods fall below Vanilla (−0.40 to −3.52), whereas ours still yields a positive gain. The stage-wise ablation on these two benchmarks is included in Table 2.

Table 9: Performance on MMMU and MMMU-Pro. Δ: absolute accuracy gain over Vanilla.

Method	MMMU (val) Acc. (Δ)↑	MMMU-Pro Acc. (Δ)↑
Vanilla	52.54	37.51
MCoT	53.96 (+1.42)	37.86 (+0.35)
CCoT	52.07 (−0.47)	37.28 (−0.23)
ICoT	44.86 (−7.68)	32.14 (−5.37)
DMLR	54.19 (+1.65)	37.11 (−0.40)
Monet	52.66 (+0.12)	33.99 (−3.52)
CoVT	50.41 (−2.13)	34.16 (−3.35)
LVR	52.30 (−0.24)	34.86 (−2.65)
LVR_{RF}	53.60 (+1.06)	35.95 (−1.56)
Ours	**54.31 (+1.77)**	**38.15 (+0.64)**

C.3 Statistical Significance

The only source of randomness in our method is the Gaussian perturbation in Stage II. We therefore run the full method with three random seeds under greedy decoding (Table 10): the seed-level standard deviation is at most 0.69 across all seven benchmarks, well below our gains over Vanilla. Since the

main experiments use greedy decoding for reproducibility, we further evaluate under sampling-based decoding (temperature 0.7, three runs; Table 11). Sampling introduces larger run-to-run variance, yet our method retains the best mean accuracy on all four benchmarks with the smallest variance, confirming that its advantage is not an artifact of greedy decoding.

Table 10: Mean $\pm$ std of our method over three random seeds (greedy decoding).

	MMVP	Hull-Bench	ScienceQA	MM-Star	MM-Vista	MMMU	MMMU-Pro
Mean	73.11	67.21	84.30	61.47	63.43	53.91	38.17
Std	0.51	0.51	0.10	0.31	0.25	0.69	0.09

Table 11: Comparison under sampling-based decoding (temperature 0.7, mean $\pm$ std over three runs).

Method	MMVP	Hull-Bench	MMMU	MMMU-Pro
Vanilla	68.45 ± 0.69	65.44 ± 0.58	52.54 ± 0.59	37.47 ± 0.64
MCoT	68.11 ± 1.50	63.72 ± 1.21	53.92 ± 1.13	37.88 ± 1.19
LVR	69.67 ± 0.67	65.16 ± 0.58	53.64 ± 0.65	35.93 ± 0.61
Ours	$\mathbf{73.56 \pm 0.51}$	$\mathbf{67.33 \pm 0.37}$	$\mathbf{54.20 \pm 0.35}$	$\mathbf{38.26 \pm 0.26}$

C.4 Sensitivity and Stability of Stage-II Optimization

Perturbation-based optimization in high-dimensional spaces can suffer from high variance or non-monotonic reward trajectories. To quantify this, we sweep the perturbation scale σ and the step size α in Equation 6 on Hull-Bench, repeating each setting with three random seeds. As shown in Table 12, there is a broad safe zone ($\sigma \in [0.05, 0.2]$, $\alpha \in [0.0005, 0.002]$) in which the reward rises nearly monotonically and saturates within the 15-step budget, with consistently high accuracy and small seed-level variance. Across all tested settings, accuracy varies within 1.8 points and never falls below Vanilla (65.40). This is because Stage II performs only a localized refinement in the neighborhood of the Stage-I latent rather than a global search over the latent space, which largely avoids the pathologies of high-dimensional perturbation-based search. Our default setting ($\sigma = 0.1$, $\alpha = 0.001$) lies inside this zone.

Table 12: Sensitivity to the perturbation scale σ (with $\alpha = 0.001$) and step size α (with $\sigma = 0.1$) on Hull-Bench over three seeds. Reward: first-step $\rightarrow$ last-step reward, with the peak reward over the 15 steps in brackets.

Setting	Acc. per seed	Mean $\pm$ std	Reward (peak)
$\sigma = 0.01$	66.35 / 66.46 / 66.25	66.35 ± 0.11	+8.11 $\rightarrow$ +6.53 (+12.95)
$\sigma = 0.05$	67.51 / 67.61 / 67.93	67.68 ± 0.22	+5.21 $\rightarrow$ +10.50 (+12.94)
$\sigma = 0.1$	67.51 / 67.30 / 67.82	67.54 ± 0.26	+6.98 $\rightarrow$ +12.16 (+12.16)
$\sigma = 0.2$	67.93 / 67.72 / 68.03	67.89 ± 0.16	+8.75 $\rightarrow$ +11.81 (+11.81)
$\sigma = 0.5$	66.56 / 66.77 / 66.46	66.60 ± 0.16	+11.97 $\rightarrow$ +10.92 (+11.97)
$\sigma = 1.0$	66.35 / 66.04 / 66.56	66.32 ± 0.26	+11.42 $\rightarrow$ +10.35 (+11.42)
$\alpha = 0.0001$	66.14 / 66.25 / 65.83	66.07 ± 0.22	+7.04 $\rightarrow$ +7.20 (+7.20)
$\alpha = 0.0005$	67.09 / 67.30 / 66.88	67.09 ± 0.21	+7.06 $\rightarrow$ +11.20 (+11.20)
$\alpha = 0.001$	67.51 / 67.30 / 67.82	67.54 ± 0.26	+6.98 $\rightarrow$ +12.16 (+12.16)
$\alpha = 0.002$	67.19 / 67.30 / 66.98	67.16 ± 0.16	+7.18 $\rightarrow$ +10.32 (+12.86)
$\alpha = 0.005$	66.35 / 66.46 / 66.56	66.46 ± 0.11	+7.42 $\rightarrow$ +9.95 (+13.00)
$\alpha = 0.01$	66.98 / 66.56 / 67.19	66.91 ± 0.32	+7.36 $\rightarrow$ +6.50 (+12.41)

C.5 Rationale and Boundary of the Confidence-Progression Reward

Why confidence. The most direct way to increase latent utilization would be to maximize the attention from the answer to the latent tokens, but this requires a full answer generation at every optimization step, which is prohibitively expensive. We instead use the prediction confidence at the

latent positions, which is available before decoding. Since the backbone is frozen, any confidence gain can only come from the latents themselves, forcing answer-relevant information to be written into them, and the distribution at the last latent position directly conditions answer generation. Moreover, entropy minimization is a classical objective in test-time adaptation [31], confidence is known to be well calibrated with correctness in language models [14], and maximizing confidence alone has been shown to improve reasoning [22]. We verify the resulting link to utilization directly: after optimization, the answer attends substantially more to the latents (Figure 6 and Figure 9), and corrupting the latents clearly degrades accuracy (Table 6).

Why progression. Entropy is not required to decrease monotonically along the latent span: the reward in Equation 5 is zero-floored, so non-improving positions are simply not rewarded rather than penalized. Requiring confidence to increase step by step, instead of pushing every position to be uniformly sharp, preserves information diversity: each latent must add information beyond its predecessor, preventing all latents from collapsing onto the same sharp state. Consistently, a non-progressive variant that minimizes the overall entropy of the latent span shows a clear accuracy drop, although it remains above Vanilla (Table 13).

Table 13: Progressive vs. non-progressive entropy reward.

Reward design	Hull-Bench Acc.↑	MMVP Acc.↑
Vanilla	65.40	68.67
LVR	66.67 (+1.27)	69.00 (+0.33)
w/o progression (minimize overall entropy)	66.67 (+1.27)	71.67 (+3.00)
w/ progression (Equation 5, ours)	**67.30 (+1.90)**	**73.00 (+4.33)**

Failure cases. We manually inspected the MMMU samples that the Stage-I-only variant answers correctly but the full method answers incorrectly. They are mostly calculation-type questions whose exact answer is not among the given options: the Stage-I model hedges and picks the closest option, which is often correct, whereas after Stage II the prediction becomes more decisive and commits to a specific but wrong option. In other words, the confidence prior can turn hedged-but-correct answers into confident-but-wrong ones on questions where decisiveness does not imply correctness, which marks a boundary of the confidence-progression assumption.

C.6 Latent Initialization

The K latent tokens are initialized from a special token of the vanilla MLLM. Concretely, we insert K copies of the <|endoftext|> word embedding into the input embedding sequence immediately after the visual and query embeddings; these vectors constitute the initial latent state $\mathcal{H}_0 \in \mathbb{R}^{K\times d}$, which is then optimized by the two-stage procedure. Unlike LVR, which initializes each latent from the last hidden state of the network, our latents reside in the input embedding space. To test whether $\mathcal{H}_0$ affects the optimization, we compare three initializations with all other hyperparameters fixed (Table 14): (A) the default <|endoftext|> embedding; (B) random Gaussian vectors $\mathcal{N}(0, 0.02^2)$ of matched scale, which carry no semantic prior; and (C) the embedding of a different special token, <|im_end|>. The three variants perform comparably. We further compare the answer-token distributions produced by latents optimized from different initializations on 50 randomly drawn MMVP samples (Table 15): the cross-initialization JS divergences (below 7×10^{-4}) are more than two orders of magnitude smaller than the reference divergence between different samples (0.19), with 100% top-1 agreement versus 44% for the reference. This indicates that, despite different starting points, the optimization converges to essentially the same solution.

Table 14: Ablation on latent initialization.

Initialization	Hull-Bench	MMVP
A. <|endoftext|> (default)	67.30	73.00
B. Random, matched scale	67.82	72.33
C. <|im_end|>	66.88	73.00

Table 15: Similarity of answer-token distributions across initializations (50 MMVP samples).

Comparison	JS (nats)	Top-1 agree.
Same sample, A vs. B	4.5×10^{-4}	100%
Same sample, A vs. C	4.6×10^{-4}	100%
Same sample, B vs. C	6.9×10^{-4}	100%
Reference (diff. samples)	0.19	44%

C.7 Comparison with LEAD

LEAD [39], DMLR [16], and our method share the high-level idea of exploiting the high bandwidth of latent representations to enhance multimodal reasoning without training, and both LEAD and DMLR inject visual patches to assist the final prediction. The two lines of work, however, differ in what the latents are expected to encode. LEAD inherits the latent-reasoning paradigm established for LLMs: instead of committing to the single most probable token at each decoding step, it performs next-state prediction over the full token probability distribution carried by the hidden states, retaining information that explicit text generation would discard. In contrast, DMLR, LVR, and our method instantiate latent reasoning specifically for MLLMs, expecting the latents to encode fine-grained visual evidence (a vision selector for DMLR, ROIs for LVR, and contrastive visual chunks for ours), analogous to the extension from textual CoT to multimodal CoT. The two mechanisms therefore operate at different levels and are complementary: LEAD-style entropy-aware decoding can be applied directly on top of our optimized latents. We re-implement LEAD with its official settings and evaluate all methods under the same protocol. As shown in Table 16, consistent with its hallucination-mitigation focus, LEAD shows a clear advantage on Hull-Bench, while our method delivers strong gains on both benchmarks. Combining the two (Table 17) yields the best Hull-Bench accuracy (69.09), and a similar gain holds for DMLR, confirming their complementarity.

Table 16: Comparison with LEAD.

Method	MMVP (Δ)	Hull-Bench (Δ)
Vanilla	68.67	65.40
DMLR	70.00 (+1.33)	65.83 (+0.43)
LEAD	69.00 (+0.33)	**67.93 (+2.53)**
Ours	**73.00 (+4.33)**	67.30 (+1.90)

Table 17: Combining LEAD-style decoding with optimized latents.

Setting	MMVP	Hull-Bench
DMLR	70.00	65.83
DMLR + LEAD decoding	70.00	67.19
Ours	**73.00**	67.30
Ours + LEAD decoding	72.67	**69.09**

D Broader Impacts

This work improves the visual reasoning capabilities of MLLMs at test time without modifying model weights. As a foundational methods paper, it has no direct path to harmful applications. More capable multimodal models could in principle be misused for surveillance or disinformation, and we encourage responsible deployment with appropriate safeguards. All experiments use publicly available benchmarks and pretrained checkpoints, with no personal data involved.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the three core contributions: the Silenced Visual Latents phenomenon, the two-stage inference-time optimization framework with frozen backbone, and generalization validated across eight benchmarks and four model backbones.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper includes a dedicated Limitations section (Section 6).

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not present formal theorems or proofs; it is an empirical methods paper with algorithmic design and experimental validation.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper provides method details, objectives, optimization stages, benchmark settings, and ablation protocols needed to reproduce the main conclusions (Section 3, Section 4, and Appendix A).

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code is publicly available at `https://github.com/zhangxin-xd/Unsilencing-Latent-Reasoning`. All benchmarks used are publicly available, and implementation details are provided in Section 4 and Appendix A.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies model backbones, benchmark suites, inference-stage design, and ablation configurations necessary to interpret the results (Section 3, Section 4, and Appendix A).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Main results use a fixed random seed and greedy decoding for reproducibility. We additionally report the mean and standard deviation over three random seeds for the stochastic Stage-II optimization, as well as results under sampling-based decoding over three runs (Appendix C.3), and seed-level variance in the hyperparameter sensitivity analysis (Appendix C.4).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section 4 provides the hardware configuration and key implementation details sufficient to estimate the computational cost of the experiments. Per-sample latency (TTFT and wall-clock time) and peak GPU memory of all methods are reported in Table 5 and Appendix C.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: To the best of our knowledge, this research follows the NeurIPS Code of Ethics, uses standard benchmark evaluations, and does not involve deceptive or harmful experimental practices.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: A broader impact discussion is provided in Appendix D.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: This work does not release a new high-risk model, dataset, or generative system; it studies optimization behavior and evaluation on existing benchmarks.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We cite all third-party assets and use them under their respective licenses and terms of use.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The only new asset is our code, released at `https://github.com/zhangxin-xd/Unsilencing-Latent-Reasoning` under the MIT License, with a README documenting setup and usage. No new datasets or models are introduced.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing experiments or research with human participants.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: No human-subject study is conducted in this work, so IRB or equivalent approval is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The MLLM backbones (e.g., Qwen2.5-VL-7B) are the subject of study and are fully cited. GPT-4o is used as an evaluator and is explicitly described in Section 4. No LLM is used as an undisclosed auxiliary component.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.