UniVLR: Unifying Text and Vision in Visual Latent Reasoning for Multimodal LLMs

Houcheng Jiang[1,4]*, **Jiajun Fu**[1]*, **Junfeng Fang**[3], **Chen Gao**[2,4]†,
Xiang Wang[1]†, **Xiangnan He**[1], **Yong Li**[2,4]
[1]University of Science and Technology of China, [2]Tsinghua University,
[3]National University of Singapore, [4]Zhongguancun Academy
{janghc, fujiajun}@mail.ustc.edu.cn

Abstract

Multimodal large language models are increasingly expected to perform *thinking with images*, yet existing visual latent reasoning methods still rely on explicit textual chain-of-thought interleaved with visual latent tokens. This interleaved design limits efficiency and keeps reasoning fragmented across separate text and vision channels. We propose **UniVLR**, a unified visual latent reasoning framework that treats textual reasoning and auxiliary visual evidence as a shared visual workspace. Instead of preserving text CoT as an independent inference-time path, UniVLR renders reasoning traces together with auxiliary images and learns to compress this unified representation into compact visual latent tokens. At inference time, the model reasons only through visual latents and directly decodes the final answer, avoiding both external tool calls and verbose text reasoning. Experiments on real-world perception and visual reasoning tasks show that UniVLR outperforms prior visual latent reasoning methods while using substantially fewer generated reasoning tokens, suggesting a more unified and efficient paradigm for visual thinking in MLLMs. Our code is available at: https://github.com/Warrenustc1958/UniVLR.

1 Introduction

Multimodal large language models (MLLMs) are moving beyond *thinking about images* toward *thinking with images* [1]. They are increasingly deployed in perception-heavy scenarios such as geometric problem solving, chart question answering, high-resolution visual understanding, and embodied planning, where the model must repeatedly attend to, localize, and integrate visual information throughout the reasoning process [2, 3, 4, 5]. To support this capability, the tool-based visual reasoning paradigm augments the reasoning chain with externally generated auxiliary images, such as crops, annotations, sketches, or intermediate diagrams, and interleaves them with textual chain-of-thought (CoT) [6, 7, 8, 9]. While effective, this paradigm is constrained by the rigidity of predefined tools, the latency of external calls, and the instability of generated auxiliary images. These limitations make it difficult to support flexible and continuous visual reasoning. To overcome this bottleneck, **visual latent reasoning** has recently emerged as a promising alternative: instead of invoking external tools, the model autoregressively generates latent visual tokens in the visual embedding space, enabling a form of internal visual reasoning that is not tied to explicit tool calls [10, 11, 12, 13, 14].

Despite recent progress, most existing visual latent reasoning methods adopt an interleaved design, where explicit text CoT and latent visual tokens are generated alternately, as shown in Figure 1(a)

*Equal contribution
†Corresponding author: {chgao96, xiangwang1123}@gmail.com

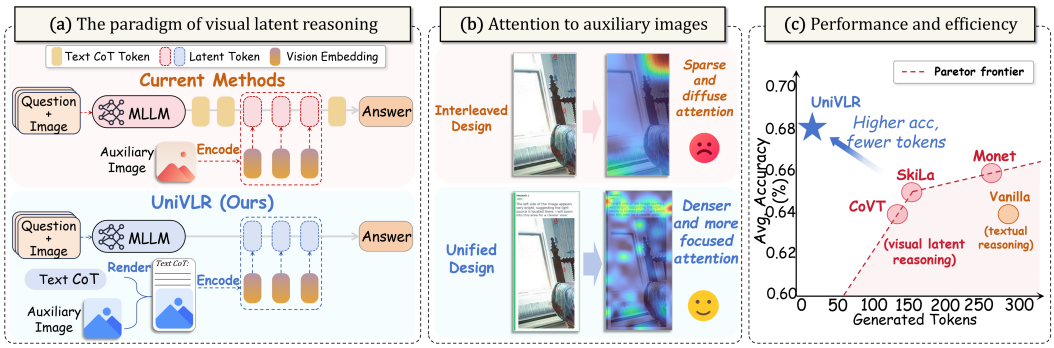

Figure 1: **Comparison between the current methods and UniVLR.** (a) Paradigm illustration comparing the interleaved design of existing visual latent reasoning methods with UniVLR. (b) Attention visualization showing that UniVLR induces denser and more focused attention to auxiliary images than the interleaved design. (c) Accuracy–efficiency comparison on visual reasoning benchmarks, where UniVLR achieves higher average accuracy with fewer generated tokens. Best viewed in color.

[10, 11, 12, 13, 14, 15]. This design has two practical limitations. First, the efficiency gain is limited. A key motivation of latent reasoning is to shorten the reasoning trajectory, yet the interleaved paradigm still requires full explicit text CoT segments between visual latent steps [16, 17, 18]. As a result, the overall generation cost is not substantially reduced compared with standard text-based CoT. Second, and more importantly, auxiliary images may not fully participate in reasoning. Because explicit text tokens and latent visual tokens appear in heterogeneous forms within a long interleaved sequence, the model can easily rely on nearby text tokens while failing to consistently attend to earlier auxiliary images. Our analysis further shows that, under interleaved design, subsequent CoT tokens exhibit sparse and diffuse attention to auxiliary images, as illustrated in Figure 1(b). This suggests that, in the interleaved setting, the model may still rely heavily on explicit textual reasoning, while auxiliary images are not always fully integrated into subsequent reasoning steps.

Does visual latent reasoning really need explicit text CoT as a separate reasoning channel? In this paper, we propose a different perspective: explicit text CoT is not necessarily an independent reasoning path that must be preserved; instead, it can be visualized and organized together with auxiliary images within a unified representation space. This view is motivated by the cognitive observation that reading is visually grounded, and in complex reasoning, humans often organize notes, annotations, and diagrams within a shared perceptual workspace, rather than switching between separate text and image channels [19, 20]. Likewise, modern MLLMs are equipped with vision encoders that have acquired strong OCR and layout understanding abilities through multimodal pretraining [21, 22, 23, 24, 25, 26], making rendered text a readily interpretable visual signal. Therefore, intermediate textual reasoning does not have to remain as explicit text tokens; it can be rendered into images, spatially composed with auxiliary images, and processed through the same visual pathway. Empirically, we observe that such unified design leads to denser and more focused attention over auxiliary images, with key regions being more consistently attended, as shown in Figure 1(b). This suggests that a unified visual representation not only preserves textual reasoning, but also enables auxiliary images to more actively participate in the reasoning process.

Based on this observation, we introduce **UniVLR**, a new visual latent reasoning paradigm that represents both textual reasoning and auxiliary images within a single unified visual representation, as shown in Figure 1(a). UniVLR is trained in two stages. In the first stage, *Visual Latent Grounding*, we use visual CoT auxiliary images as latent alignment targets, enabling the model to autoregressively generate semantically meaningful latent visual tokens in the visual embedding space. In the second stage, *Text–Vision Unified Alignment*, we render each textual reasoning step into an image and spatially concatenate it with the corresponding auxiliary image, producing a unified image that jointly carries the reasoning process and the auxiliary image. The vision encoder representation of this unified image is then used as the latent alignment target, allowing the model to learn to conduct the entire reasoning process through unified visual latent tokens. At inference time, the reasoning trajectory is composed entirely of unified visual latent tokens, and only the final answer is decoded in natural language. With this simple yet effective design, UniVLR can organize multi-step reasoning,

multiple reference images, and complex annotations on a unified visual canvas, enabling efficient visual latent reasoning without maintaining explicit text CoT as a separate channel.

We conduct extensive experiments on real-world perception and reasoning benchmarks, including V* [2], HRBench [3], and MME-RealWorld [5], with multiple base MLLMs such as Qwen2.5-VL [23]. Remarkably, even after removing explicit text CoT during inference, UniVLR outperforms interleaved visual latent reasoning methods that preserve the text channel, including Monet, LVR, and SkiLa. On average, UniVLR improves reasoning accuracy by **5.4%** while reducing the number of generated tokens by **15.2**×, as shown in Figure 1(c). Moreover, the hidden-state distributions of UniVLR's latent tokens are more coherent, suggesting that textual reasoning and auxiliary images are better aligned within a shared visual latent space. The consistent gains across different base MLLMs further demonstrate the robustness and generality of UniVLR, pushing visual latent reasoning toward a more unified form of multimodal thinking.

2 Method

In this section, we present **UniVLR**, a unified visual latent reasoning framework that represents textual reasoning and auxiliary images within a single visual latent channel, as illustrated in Figure 2. Section 2.1 introduces unified visual canvas rendering, which converts explicit text CoT and auxiliary visual evidence into a shared visual representation. Section 2.2 describes the two-stage latent alignment procedure, including Visual Latent Grounding and Text–Vision Unification. Section 2.3 presents the continuous autoregressive inference process, where the model reasons with compact unified visual latent tokens and decodes only the final answer in natural language.

2.1 Unified Visual Canvas Rendering

In this section, we describe how UniVLR converts heterogeneous reasoning traces into a unified visual representation. Existing visual latent reasoning methods usually preserve explicit text CoT as a separate discrete sequence while using auxiliary images as visual latent supervision. In contrast, UniVLR renders textual reasoning and auxiliary images into the same visual workspace, allowing both sources of information to be processed through the vision pathway.

Unified canvas construction. For each training instance, let $\mathcal{R} = \{r_1, \ldots, r_L\}$ denote the textual reasoning trace and $\mathcal{U} = \{u_1, \ldots, u_M\}$ denote the corresponding auxiliary images, such as crops, annotations, sketches, or intermediate diagrams. We define a rendering function $\Phi(\cdot)$ that composes them into a unified visual canvas:

$$c = \Phi(\mathcal{R}, \mathcal{U}). \tag{1}$$

The rendered canvas preserves the semantic content of text CoT while spatially organizing it with auxiliary visual evidence. This converts explicit text reasoning from a separate language channel into a visually readable signal.

Adaptive rendering. To make the canvas compact and readable, $\Phi(\cdot)$ applies lightweight adaptive rendering strategies. For the textual part, it adjusts font size, line wrapping, and spacing according to the length of each reasoning step. For the visual part, it resizes auxiliary images according to their aspect ratios and optionally highlights salient regions with simple visual cues such as boxes or arrows. These operations are not task-specific tools, but a general interface for representing text and visual reasoning on the same canvas. More details of the rendering algorithm are provided in Appendix B.5.

Target visual embeddings. We use the vision encoder of the base MLLM to extract a 2D visual feature map from the unified canvas:

$$\mathbf{F} = E_{\text{vis}}(c) \in \mathbb{R}^{H_f \times W_f \times d}, \tag{2}$$

where d is the hidden dimension of the language model. To obtain exactly K latent supervision targets while preserving the spatial structure of the canvas, we adopt aspect-aware 2D pooling. We choose a pooling grid $(h^\star, w^\star)$ satisfying $h^\star w^\star = K$ by minimizing the aspect-ratio distortion:

$$(h^\star, w^\star) = \arg\min_{hw=K} \left| \log \frac{w}{h} - \log \frac{W_f}{H_f} \right|. \tag{3}$$

The feature map is then adaptively pooled with this grid and flattened into the target latent sequence:

$$\tilde{\mathbf{Z}} = \Pi_{h^\star, w^\star}(\mathbf{F}) = [\tilde{\mathbf{z}}_1, \ldots, \tilde{\mathbf{z}}_K] \in \mathbb{R}^{K \times d}. \tag{4}$$

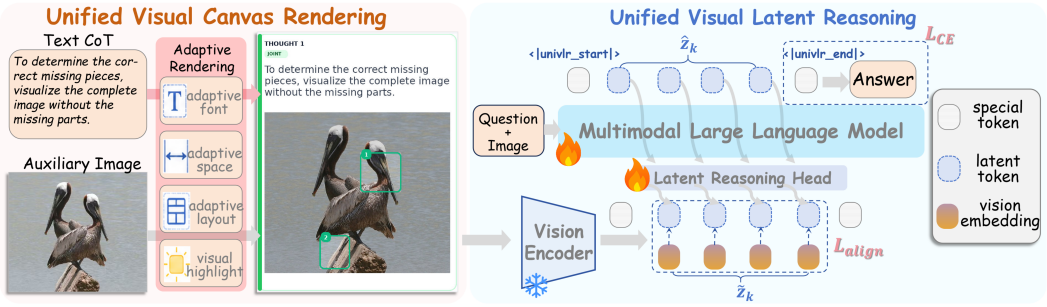

Figure 2: **Overview of the proposed UniVLR.** Left: Unified Visual Canvas Rendering. Textual reasoning traces are rendered and composed with auxiliary visual evidence into a shared canvas, enabling both reasoning semantics and visual evidence to be encoded by the same vision encoder. Right: Unified Visual Latent Alignment. The canvas embeddings are compressed into fixed-length latent targets, and the MLLM learns to autoregressively generate continuous visual latent tokens before decoding the final answer.

The resulting embeddings preserve the coarse layout of the rendered reasoning canvas and serve as visual supervision targets for latent alignment.

2.2 Unified Visual Latent Alignment

After constructing visual supervision targets, we train the model to autoregressively generate continuous visual latent tokens. Instead of reconstructing explicit text CoT, our goal is to align the model's latent reasoning states with visual embeddings that encode intermediate reasoning evidence. We use a two-stage alignment strategy to make this process stable and semantically unified.

Stage I: Visual Latent Grounding. The first stage establishes a basic continuous reasoning interface between the MLLM and its visual embedding space. We use auxiliary visual reasoning images as latent alignment targets. Given an auxiliary image, we extract its visual embeddings and compress them into a fixed-length latent sequence $\tilde{\mathbf{Z}}^{(1)} = \{\tilde{\mathbf{z}}_k^{(1)}\}_{k=1}^{K}$. The model is then trained to enter latent reasoning mode, autoregressively generate K continuous latent tokens, and return to answer decoding.

Stage II: Text–Vision Unified Alignment. The second stage performs the key unification step. Starting from the Stage-I checkpoint, we replace auxiliary-image targets with unified canvas targets. Each canvas contains both rendered textual reasoning and auxiliary visual evidence, and its visual embeddings are compressed into $\tilde{\mathbf{Z}}^{(2)} = \{\tilde{\mathbf{z}}_k^{(2)}\}_{k=1}^{K}$. Training on these targets teaches the model to represent explicit reasoning traces and visual evidence through the same latent visual channel.

Latent alignment Objective. Both stages use the same autoregressive latent training format:

$$\mathcal{S} = [\mathcal{X}, \texttt{<|univlr\_start|>}, \tilde{\mathbf{z}}_1, \ldots, \tilde{\mathbf{z}}_K, \texttt{<|univlr\_end|>}, \mathcal{A}], \tag{5}$$

where $\mathcal{X}$ is the original multimodal input, $\mathcal{A}$ is the final answer, and $\tilde{\mathbf{z}}_k$ denotes the target latent embedding from either Stage I or Stage II. At the k-th latent step:

$$\hat{\mathbf{z}}_k = g_\phi(\mathbf{h}_{k-1,i}). \tag{6}$$

We align predicted and target latent tokens with a normalized regression loss:

$$\mathcal{L}_{\text{align}} = \frac{1}{K}\sum_{k=1}^{K}\left[\frac{1}{d}\left\|\text{LN}(\hat{\mathbf{z}}_k) - \text{LN}(\tilde{\mathbf{z}}_k)\right\|_2^2 + 1 - \cos\left(\text{LN}(\hat{\mathbf{z}}_k), \text{LN}(\tilde{\mathbf{z}}_k)\right)\right]. \tag{7}$$

The final objective combines latent alignment with standard language modeling:

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda_{\text{align}}\mathcal{L}_{\text{align}}. \tag{8}$$

Here, $\mathcal{L}_{\text{CE}}$ is applied only to discrete tokens, including the two special control tokens <|univlr_start|> and <|univlr_end|>, as well as the final answer tokens in $\mathcal{A}$. During training, we apply latent teacher forcing by feeding the target latent embedding as the next latent input, which stabilizes continuous autoregressive learning.

2.3 Continuous Autoregressive Inference

After training, UniVLR performs inference without rendering text CoT or extracting target visual embeddings. The rendering function $\Phi(\cdot)$, the vision encoder used for target extraction, and all teacher latent targets are discarded. The model receives only the original multimodal prompt $\mathcal{X} = (\mathcal{Q}, \mathcal{I})$.

Latent reasoning mode. Once the model generates <|univlr_start|>, it switches from discrete token generation to continuous latent reasoning. At latent step k, the model predicts a latent token from the previous certain layer i hidden state $\mathbf{h}_{k-1,i}$: $\hat{\mathbf{z}}_k = g_\phi(\mathbf{h}_{k-1,i})$. The predicted latent token is directly fed back as the input embedding for the next step:

$$\mathbf{e}_k = \hat{\mathbf{z}}_k. \tag{9}$$

This recurrence is repeated for a predefined latent budget of K steps.

Answer decoding. After K latent steps, the model emits <|univlr_end|> and switches back to the discrete vocabulary space to decode the final answer $\mathcal{A}$. Therefore, inference consists of a compact continuous reasoning phase followed by a short natural-language answer phase. This allows UniVLR to retain the information of text CoT and auxiliary images without generating explicit intermediate reasoning tokens at test time.

3 Experiments

In this section, we conduct extensive experiments to address the following research questions:

RQ1: How does UniVLR perform compared with baselines, and can it achieve competitive or superior performance with substantially fewer reasoning tokens?

RQ2: How are the hidden-state representations of UniVLR's latent reasoning tokens distributed compared with explicit text tokens and image tokens? Do the learned latent tokens align more closely with the visual-token representation space?

RQ3: How does the number of latent reasoning tokens affect the performance and efficiency of UniVLR, and what latent token budget provides the best balance?

RQ4: How do key components of UniVLR, including visual-text rendering, unified canvas construction, contribute to the final performance?

3.1 Experimental Setup

In this subsection, we summarize the evaluation benchmarks, dataset, base models and baseline methods used in our experiments. Further details and additional experiments are provided in Appendix B and Appendix D.

Benchmarks & Metrics. We evaluate UniVLR on a diverse suite of perception-centric and visual reasoning benchmarks. Specifically, we use **V*** [2] to evaluate guided visual search and spatial reasoning, **HRBench4K** [3] and **HRBench8K** [3] to assess high-resolution fine-grained perception, and **MME-RealWorld-Lite** [5] to measure real-world multimodal comprehension. For each benchmark, we report the official overall score as well as its corresponding sub-category scores when available. In addition, we use the average of the overall scores across benchmarks as a compact indicator of general performance.

Base models & Baselines. We compare UniVLR with representative methods from three reasoning paradigms. First, *textual reasoning* baselines include GPT-4o, Qwen2.5-VL-7B, and a vanilla SFT variant of Qwen2.5-VL-7B, which rely primarily on explicit textual reasoning or direct answer generation. Second, *tool-based visual reasoning* methods, including PixelReasoner [9] and DeepEyes [8], improve perception by invoking external visual operations such as cropping, zooming, or multi-turn image inspection. Third, *visual latent reasoning* methods include LVR [11], Monet [12], SkiLa [13],CoVT[15] which introduce latent visual reasoning tokens while still preserving explicit textual reasoning channels. Unlike these methods, UniVLR removes the explicit reasoning text channel at inference time and performs reasoning through a compact sequence of unified visual latent tokens.

Implementation Details. We instantiate UniVLR on top of Qwen2.5-VL-7B-Instruct [23]. To preserve the pretrained visual prior, we freeze the vision encoder and patch-merger, and fine-tune

Table 1: Comparison with representative baselines on perception-centric and visual reasoning benchmarks. The best results are highlighted in **bold**. Results marked with '*' are taken from prior work. UniVLR achieves strong overall performance across all evaluated benchmarks.

Model	V*			HRBench4K			HRBench8K			MME-RealWorld-Lite		
	Overall	Attr.	Spa.	Overall	FSP	FCP	Overall	FSP	FCP	Overall	Rea.	Perc.
Textual Reasoning												
GPT-4o [27]	67.5*	72.2*	60.5*	59.0*	70.0*	48.0*	55.5*	62.0*	49.0*	52.0*	48.3*	54.4*
Qwen2.5-VL-7B [23]	77.4	78.3	76.3	69.0	85.8	52.2	66.0	80.3	51.8	46.2	43.1	48.2
+ vanilla SFT	75.4	80.0	68.5	69.1	81.3	57.0	63.6	73.3	54.0	45.5	39.9	49.1
Tool-based Visual Reasoning												
PixelReasoner [9]	80.6*	83.5*	76.3*	72.9*	86.0*	60.3*	66.9*	80.0*	54.3*	49.7*	44.5*	53.1*
DeepEyes [8]	83.3*	84.4*	81.6*	71.3*	83.8*	58.8*	65.1*	77.0*	53.3*	54.3*	50.5*	56.6*
Visual Latent Reasoning												
LVR [11]	80.6	81.7	79.8	69.9	84.7	55.7	66.9	77.7	56.2	39.1	37.5	40.1
Monet [12]	79.1	81.7	75.0	71.9	**89.3**	54.5	63.5	76.5	50.5	46.9	40.3	51.2
SkiLa [13]	80.1	79.1	81.6	70.3	84.7	55.7	62.9	77.5	48.3	45.6	36.3	51.6
CoVT [15]	78.0	79.1	76.3	71.9	84.2	59.5	**69.7**	**85.4**	54.0	48.2	42.9	51.6
UniVLR	**82.7**	**83.5**	**81.6**	**73.3**	86.0	**60.5**	68.8	78.8	**58.8**	**50.7**	**44.7**	**54.5**

the LLM backbone together with a lightweight MLP-based latent reasoning head. **Unless otherwise specified, the teacher forcing latent budget is fixed to $K_{train} = 24$. At inference time, we use a shorter latent reasoning budget by default, setting $K_{infer} = 12$.** We report both UniVLR-Stage1 and the final UniVLR model, where the former serves as an intermediate variant and the latter denotes our final model used for main comparison. Training details, including optimization hyperparameters and data construction, are provided in Appendix B.

3.2 Main Results and Efficiency Analysis (RQ1)

To answer RQ1, we compare UniVLR (UniVLR-Stage2) with textual reasoning, tool-based visual reasoning, and visual latent reasoning baselines on four main benchmarks. UniVLR's latent reasoning budget is fixed to $K_{\text{infer}} = 12$. This shorter inference budget is chosen according to the token-scaling analysis in Section3.4. Table 1 reports the performance comparison, and Figure 3(a) further compares the reasoning-token budget. Based on the results, we find that:

- **Obs 1: UniVLR improves over textual reasoning baselines.** Compared with Qwen2.5-VL-7B, UniVLR improves the overall scores from 77.4 to 82.7 on V*, 69.0 to 73.3 on HRBench4K, 66.0 to 68.8 on HRBench8K, and 46.2 to 50.7 on MME-RealWorld-Lite. In contrast, vanilla SFT does not yield consistent gains, indicating that the improvement comes from unified visual latent reasoning rather than simple fine-tuning.
- **Obs 2: UniVLR achieves the strongest overall performance among visual latent reasoning methods.** UniVLR obtains the best overall scores on V*, HRBench4K, and MME-RealWorld-Lite. The average overall score of UniVLR reaches 68.9, outperforming LVR, Monet, and SkiLa. This suggests that unifying text traces and auxiliary images into one visual latent channel is more effective than interleaved text–latent reasoning.
- **Obs 3: UniVLR substantially reduces generated reasoning tokens by removing explicit intermediate text generation.** As depicted in Figure 3(a), interleaved methods (*e.g.*, Monet, SkiLa, and CoVT) generate 190 to 270 reasoning tokens per instance, predominantly explicit text. Even LVR, which only enhances visual grounding, requires explicit text CoT. In stark contrast, UniVLR achieves competitive or stronger performance using only 12 latent tokens and no generated intermediate text tokens. This reduction in generated reasoning length suggests that UniVLR can encode useful intermediate reasoning information in a compact continuous format.

3.3 Unified Latent Representation Analysis (RQ2)

To answer RQ2, we analyze whether the generated latent tokens behave as meaningful visual reasoning states. We examine their token efficiency, last-hidden-state distribution, and sensitivity to inference-time perturbations. The results are shown in Figures 3 and 4, from which we can find that:

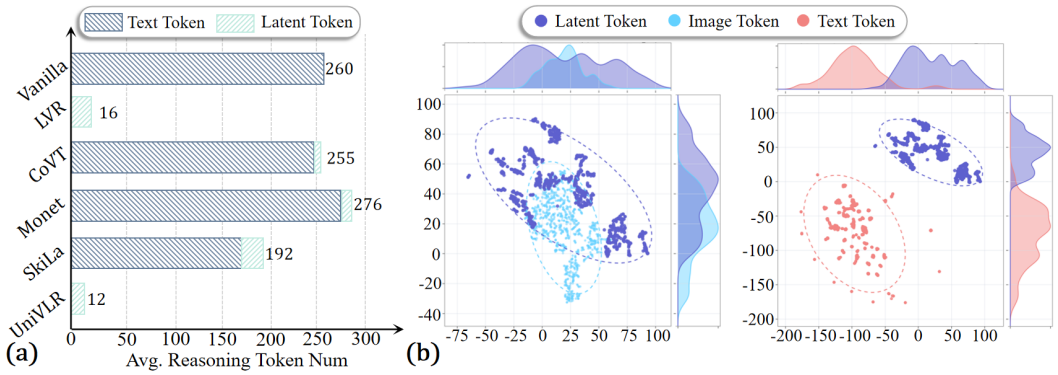

Figure 3: **Reasoning-token efficiency and latent representation distribution.** **(a)** UniVLR performs reasoning with a compact latent-token budget and no generated text CoT. **(b)** Last-hidden-state visualization shows that UniVLR latent tokens are closer to image-token representations than to text-token representations.

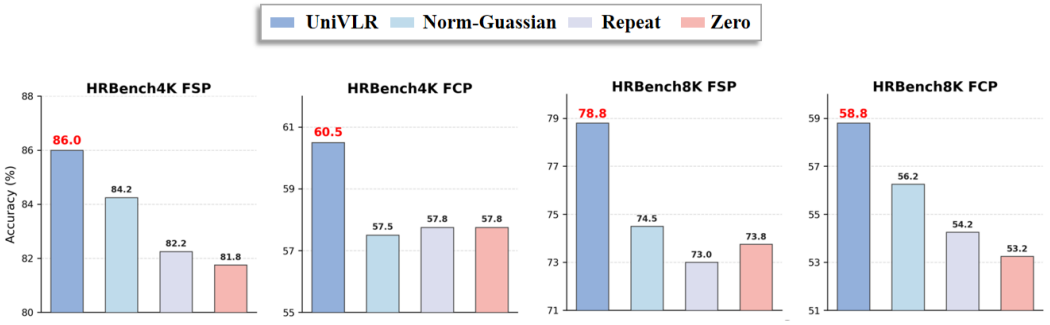

Figure 4: **Causal perturbation analysis of latent reasoning tokens.** We perturb latent hidden states during inference by zeroing them, injecting Gaussian noise, or repeating the first latent state. The resulting accuracy drops indicate that latent tokens carry task-relevant reasoning information.

- **Obs 4: UniVLR latent tokens align more closely with visual representations than textual ones.** In Figure 3(b), latent tokens overlap substantially with image-token hidden states, while remaining clearly separated from text-token states. This suggests that UniVLR's latent states are closer to visual-token representations, providing evidence that the model uses a visually grounded latent channel rather than only an implicit textual CoT channel.
- **Obs 5: Latent tokens have a direct impact on final answer prediction.** In Figure 4, all perturbations consistently reduce accuracy on HRBench. For example, zeroing latent states drops HRBench8K-FCP from 58.8 to 53.2, and repeating the first latent state drops HRBench8K-FSP from 78.8 to 73.0. This indicates that the generated latent sequence contains task-relevant information and is not merely an unused placeholder.

3.4 Effect of Latent Token Number (RQ3)

To answer RQ3, we vary the inference-time latent token number K from 0 to 36 and evaluate both UniVLR-Stage1 and UniVLR (UniVLR-Stage2). The results are summarized in Figure 5, from which we find that:

- **Obs 6: Latent reasoning improves direct answering, but more tokens are not always better.** Across benchmarks, $K = 0$ gives the weakest performance, while introducing a small number of latent tokens brings clear gains. Performance typically peaks at a moderate budget and then fluctuates or drops, suggesting that excessive latent steps may introduce representation drift.
- **Obs 7: Stage-II training makes latent scaling more stable.** Compared with UniVLR-Stage1, the final UniVLR generally achieves higher performance across nonzero K values. It also shows

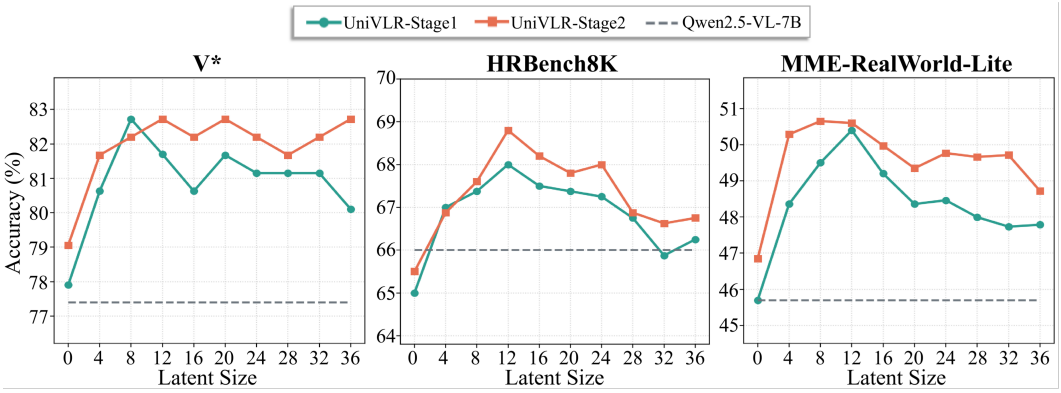

Figure 5: **Effect of the number of unified visual latent tokens during inference on test accuracy.** Both UniVLR-Stage1 and UniVLR are trained with a fixed teacher-forcing latent size of $K = 24$. The dashed line marks the zero-shot accuracy of Qwen2.5-VL-7B.

Table 2: Comprehensive Ablation Study decoupled by training stages. (Left) REPA ablations evaluated on Stage I models. **(Right)** RFC ablations evaluated on Stage II models. Both stages are evaluated consistently across V*, HRBench8K, and fine-grained subsets of MME-RealWorld-Lite. Our default UniVLR settings are highlighted in gray.

Stage I: REPA Ablations	V*	HRBench8K	MME-RealWorld-Lite		
	Overall	Overall	**Overall**	Rea.	Perc.
UniVLR-Stage1 (Default)	**81.7**	**68.0**	**50.4**	42.4	**55.6**
Extraction Strategy (Default: Avg 2D Pool)					
Avg 1D Pool	79.1	66.3	46.5	38.0	51.9
MLERP 2D Pool	71.2	59.1	48.5	41.0	53.2
Latent Reasoning Head (Default: MLP)					
w/o Head	81.2	66.5	48.7	**43.2**	52.2
GLU Head	81.2	64.4	48.4	41.9	52.5
Hidden States Aligned Layer (Default: Middle)					
Front Layer	81.5	66.3	49.2	40.7	54.7
Last Layer	81.2	67.3	48.5	41.2	53.1

Stage II: RFC Ablations	V*	HRBench8K	MME-RealWorld-Lite		
	Overall	Overall	**Overall**	Rea.	Perc.
UniVLR-Stage2 (Default)	**82.7**	**68.8**	**50.7**	44.7	54.5
Training Curriculum (Default: Two Stages)					
Stage I Only (Warm-up)	81.7	68.0	50.4	42.4	**55.6**
Single-Stage Mixed	77.5	64.8	47.1	42.1	50.2
Render Strategy (Default: Vertical Layout1)					
Left-Right Layout2	82.1	67.8	50.1	43.5	54.5
Fixed Adaptive Wrap3	81.6	66.5	48.8	44.9	51.2
Data Ablation (Default: Filtered Zebra + VisCoT)					
Unfiltered Zebra-CoT	79.1	65.5	49.9	**47.1**	51.8
Pure Visual-CoT	81.5	66.3	49.8	43.9	53.6

a wider stable plateau, especially on MME-RealWorld-Lite, indicating that unified text–vision alignment improves robustness to the inference-time latent budget.

3.5 Ablation Study (RQ4)

To answer RQ4, we rigorously disentangle the contributions of our proposed designs. As shown in Table 2, we decouple the ablation studies into two orthogonal dimensions: **Representation Alignment (REPA)** evaluated on Stage I, and **Reasoning Formulation and Curriculum (RFC)** evaluated on Stage II. We summarize our findings below:

- **Obs 8: For RFC, an incremental curriculum and vertical spatial rendering are critical for mastering complex logic.** Directly training on a single-stage mixture of warm-up and hard examples leads to a clear performance drop, with V* decreasing to 77.5. Our two-stage curriculum effectively mitigates representation collapse by anchoring the latent interface first. Furthermore, our *Vertical Layout* strategy outperforms both *Left-Right* and *Fixed Wrap* layouts. Detailed rendering algorithms are provided in Appendix B.5. Finally, rigorous data filtering is essential; using unfiltered hard examples introduces noisy or shortcut-prone supervision, which can encourage the latent tokens to encode spurious textual patterns
- **Obs 9: For REPA, preserving 2D spatial topology and utilizing a lightweight projection head are fundamental.** Replacing aspect-aware 2D pooling with Avg 1D Pool or MLERP[28] weakens the preservation of spatial layout, dropping reasoning performance on MME-RealWorld-Lite e.g., *Avg 1D* drops to 38.0. Architecturally, a simple MLP head optimally balances visual feature alignment and autoregressive stability, outperforming both head-less and over-parameterized *GLU* designs. Moreover, aligning latent tokens to middle-layer hidden states yields better performance

than aligning them to final-layer hidden states, likely because middle layers preserve richer visual and spatial information. This finding aligns with previous research ([29][30][31][32])suggesting that the middle layers of MLLMs primarily encode visual information.

4 Related Work

Thinking with Images. Recent studies have explored *thinking with images*, where visual information is no longer treated as a passive input but as an active reasoning workspace. Representative methods equip MLLMs with visual operations such as cropping, zooming, grounding, or frame selection, and train models to invoke these operations during multi-step reasoning [8, 9, 6, 7, 1]. This paradigm is especially effective for fine-grained perception and high-resolution reasoning, since models can actively revisit local visual evidence instead of relying only on the initial image encoding. However, such methods usually require explicit tool invocation, multi-turn interaction, or additional visual processing at inference time, which increases latency and makes reasoning dependent on predefined operation sets. In contrast, UniVLR does not use external visual tools during inference. We instead convert intermediate reasoning evidence into a compact sequence of unified visual latent tokens, allowing the model to internalize visual deliberation within a single latent reasoning channel.

Visual Latent Reasoning. Another line of work aims to move intermediate reasoning from discrete text tokens into continuous latent spaces. Latent visual reasoning methods train MLLMs to generate visual embeddings, latent sketches, or continuous visual thoughts as intermediate reasoning states, often by reconstructing image features or sketch-like supervision signals [11, 12, 13, 15, 14]. Recent rendered-reasoning approaches further show that textual CoT can be rendered into images and compressed into visual or latent representations, reducing the cost of verbose textual reasoning while preserving inspectable supervision signals [26, 25]. Despite these advances, existing visual latent reasoning methods usually preserve an explicit textual reasoning channel, alternate between text and latent visual tokens, or focus on compressing textual CoT alone. UniVLR differs by unifying rendered textual reasoning and auxiliary visual evidence into a single visual canvas, then aligning autoregressive latent tokens to this unified visual representation. Thus, explicit text CoT is not maintained as a separate reasoning path; both reasoning semantics and visual evidence are supervised through a shared compact visual latent space.

5 Limitations

While **UniVLR** shows that explicit text CoT can be absorbed into a unified visual latent channel, several limitations remain. First, its effectiveness depends on the vision encoder's OCR and layout understanding; models with weaker visual priors may benefit less from the same rendering strategy. Second, visual latent tokens are more efficient but less directly inspectable than natural-language rationales. Although our representation and perturbation analyses show that they carry meaningful reasoning information, they do not fully provide the transparency of explicit CoT. Third, we currently use a fixed latent-token budget, while different tasks may require different amounts of latent computation. Finally, UniVLR is not meant to replace tool-based reasoning for tasks requiring exact measurement, exhaustive high-resolution search, or executable visual manipulation, where external tools can remain complementary.

6 Conclusion

We introduced **UniVLR**, a unified visual latent reasoning framework that removes explicit text CoT as a separate inference-time channel. By rendering textual reasoning and auxiliary visual evidence into a shared visual workspace, UniVLR learns compact visual latent tokens that carry both reasoning semantics and perceptual information. At inference time, it reasons through visual latents and decodes only the final answer, avoiding verbose text reasoning and external tool calls. Experiments show that UniVLR improves over prior visual latent reasoning methods with substantially fewer generated tokens, while analyses of hidden states, latent-token scaling, and ablations support the effectiveness of unified visual latent reasoning.

References

[1] Zhaochen Su, Peng Xiang, Hangyu Guo, Zhenhua Liu, Yan Ma, Xiaoye Qu, Jiaqi Liu, Yanshu Li, Kaide Zeng, Zhengyuan Yang, Linjie Li, Yu Cheng, Heng Ji, Junxian He, and Yi R. (May) Fung. Thinking with images for multimodal reasoning: Foundations, methods, and future frontiers. *CoRR*, abs/2506.23918, 2025.

[2] Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal llms. In *CVPR*, pages 13084–13094. IEEE, 2024.

[3] Wenbin Wang, Liang Ding, Minyan Zeng, Xiabin Zhou, Li Shen, Yong Luo, Wei Yu, and Dacheng Tao. Divide, conquer and combine: A training-free framework for high-resolution image perception in multimodal large language models. In *AAAI*, pages 7907–7915. AAAI Press, 2025.

[4] 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 *ICLR*. OpenReview.net, 2024.

[5] Yifan Zhang, Huanyu Zhang, Haochen Tian, Chaoyou Fu, Shuangqing Zhang, Junfei Wu, Feng Li, Kun Wang, Qingsong Wen, Zhang Zhang, Liang Wang, and Rong Jin. Mme-realworld: Could your multimodal LLM challenge high-resolution real-world scenarios that are difficult for humans? In *ICLR*. OpenReview.net, 2025.

[6] 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. In *NeurIPS*, 2024.

[7] Mingyuan Wu, Jingcheng Yang, Jize Jiang, Meitang Li, Kaizhuo Yan, Hanchao Yu, Minjia Zhang, Chengxiang Zhai, and Klara Nahrstedt. Vtool-r1: Vlms learn to think with images via reinforcement learning on multimodal tool use. *CoRR*, abs/2505.19255, 2025.

[8] Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. Deepeyes: Incentivizing "thinking with images" via reinforcement learning. *CoRR*, abs/2505.14362, 2025.

[9] Alex Su, Haozhe Wang, Weiming Ren, Fangzhen Lin, and Wenhu Chen. Pixel reasoner: Incentivizing pixel-space reasoning with curiosity-driven reinforcement learning. *CoRR*, abs/2505.15966, 2025.

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

[11] Bangzheng Li, Ximeng Sun, Jiang Liu, Ze Wang, Jialian Wu, Xiaodong Yu, Hao Chen, Emad Barsoum, Muhao Chen, and Zicheng Liu. Latent visual reasoning. *CoRR*, abs/2509.24251, 2025.

[12] 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. *CoRR*, abs/2511.21395, 2025.

[13] Jintao Tong, Jiaqi Gu, Yujing Lou, Lubin Fan, Yixiong Zou, Yue Wu, Jieping Ye, and Ruixuan Li. Sketch-in-latents: Eliciting unified reasoning in mllms. *CoRR*, abs/2512.16584, 2025.

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

[15] 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. *CoRR*, abs/2511.19418, 2025.

[16] Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. *CoRR*, abs/2412.06769, 2024.

[17] Zhen Zhang, Xuehai He, Weixiang Yan, Ao Shen, Chenyang Zhao, Shuohang Wang, Yelong Shen, and Xin Eric Wang. Soft thinking: Unlocking the reasoning potential of llms in continuous concept space. *CoRR*, abs/2505.15778, 2025.

[18] Xinlei Yu, Zhangquan Chen, Yongbo He, Tianyu Fu, Cheng Yang, Chengming Xu, Yue Ma, Xiaobin Hu, Zhe Cao, Jie Xu, et al. The latent space: Foundation, evolution, mechanism, ability, and outlook. *arXiv preprint arXiv:2604.02029*, 2026.

[19] Barbara Tversky. Visualizing thought. In *Handbook of human centric visualization*, pages 3–40. Springer, 2013.

[20] Vinod Goel. *Sketches of thought*. MIT press, 1995.

[21] Haoran Wei, Yaofeng Sun, and Yukun Li. Deepseek-ocr: Contexts optical compression. *CoRR*, abs/2510.18234, 2025.

[22] Haoran Wei, Yaofeng Sun, and Yukun Li. Deepseek-ocr 2: Visual causal flow. *CoRR*, abs/2601.20552, 2026.

[23] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Ming-Hsuan 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. *CoRR*, abs/2502.13923, 2025.

[24] Qwen Team. Qwen3-vl technical report. *CoRR*, abs/2511.21631, 2025.

[25] Bo Lv, Yasheng Sun, Junjie Wang, and Haoxiang Shi. Onelatent: Single-token compression for visual latent reasoning. *CoRR*, abs/2602.13738, 2026.

[26] Yifan Wang, Shiyu Li, Peiming Li, Xiaochen Yang, Yang Tang, and Zheng Wei. Render-of-thought: Rendering textual chain-of-thought as images for visual latent reasoning. *CoRR*, abs/2601.14750, 2026.

[27] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[28] Minchul Kim, Shangqian Gao, Yen-Chang Hsu, Yilin Shen, and Hongxia Jin. Token fusion: Bridging the gap between token pruning and token merging. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1383–1392, 2024.

[29] Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Patel, Jalal Naghiyev, Yann LeCun, and Ravid Shwartz-Ziv. Layer by layer: Uncovering hidden representations in language models. *arXiv preprint arXiv:2502.02013*, 2025.

[30] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. 2025.

[31] Jaewoo Lee, Keyang Xuan, Chanakya Ekbote, Sandeep Polisetty, Yi R Fung, and Paul Pu Liang. Tamp: Token-adaptive layerwise pruning in multimodal large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 6892–6908, 2025.

[32] Seil Kang, Jinyeong Kim, Junhyeok Kim, and Seong Jae Hwang. Your large vision-language model only needs a few attention heads for visual grounding. pages 9339–9350, 2025.

[33] Haodong Duan, Junming Yang, Yuxuan Qiao, Xinyu Fang, Lin Chen, Yuan Liu, Xiaoyi Dong, Yuhang Zang, Pan Zhang, Jiaqi Wang, et al. Vlmevalkit: An open-source toolkit for evaluating large multi-modality models. In *Proceedings of the 32nd ACM international conference on multimedia*, 2024.

[34] Ang Li, Charles Wang, Kaiyu Yue, Zikui Cai, Ollie Liu, Deqing Fu, Peng Guo, Wang Bill Zhu, Vatsal Sharan, Robin Jia, et al. Zebra-cot: A dataset for interleaved vision language reasoning. *arXiv preprint arXiv:2507.16746*, 2025.

[35] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326, 2019.

[36] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023.

[37] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*, 2023.

[38] Runqi Qiao, Qiuna Tan, Guanting Dong, Minhui Wu, Chong Sun, Xiaoshuai Song, Zhuoma GongQue, Shanglin Lei, Zhe Wei, Miaoxuan Zhang, et al. We-math: Does your large multimodal model achieve human-like mathematical reasoning? *arXiv preprint arXiv:2407.01284*, 2024.

[39] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. *CoRR*, abs/1506.03099, 2015.

A Broader Impacts

UniVLR explores a more efficient way for MLLMs to conduct visual reasoning by replacing verbose explicit reasoning traces with compact visual latent tokens. This may reduce inference cost and latency for perception-heavy applications such as high-resolution document understanding, chart analysis, visual search, and embodied decision making. At the same time, moving reasoning from explicit text into latent visual states may reduce the direct interpretability of intermediate reasoning, which could make debugging, auditing, or safety monitoring more challenging. Although our analyses show that the learned latent tokens carry meaningful reasoning information, they do not provide the same human-readable transparency as textual CoT. Therefore, future deployments of visual latent reasoning systems should be accompanied by complementary interpretability, monitoring, and failure-diagnosis tools, especially in high-stakes domains where visual reasoning errors may lead to harmful decisions. We view UniVLR as a step toward more efficient multimodal reasoning, while emphasizing that efficiency gains should not come at the cost of accountability and safety.

B Implementation Details

B.1 Benchmark and Model Details

V* [2] is designed to evaluate MLLMs in their ability to process high-resolution images and focus on visual details.We select 191 samples to complete the evaluation.

HRBench [3] designs high-resolution multimodal benchmarks, which consists two sub-tasks: Fine-grained Single-instance Perception (FSP) and Fine-grained Cross-instance Perception (FCP). The FSP task includes 100 samples, which includes tasks such as attribute recognition, OCR, visual prompting. The FCP task also comprises 100 samples which encompasses map analysis, chart analysis and spatial relationship assessment.HR-Bench is available in two versions: HR-Bench 8K and HR-Bench 4K. The HR-Bench 8K includes images with an average resolution of 8K. Additionally, we manually annotate the coordinates of objects relevant to the questions within the 8K image and crop these image to 4K resolution.

MME-RealWorld-Lite [5] contains 13K high-quality images, annotated by 32 volunteers, resulting in 29K question-answer pairs that cover 43 subtasks across 5 real-world scenarios. We choose 1919 samples to evaluate all latent visual reasoning baselines.

Qwen2.5-VL-7B [23] is a frontier MLLM with exceptional multimodal understanding capabilities and a long context window. By default, we adopt the natural resolution mechanism to avoid distorting the aspect ratio. We set a maximum pixel of $5120\times28\times28$ and a minimum pixel of $128\times28\times28$.

B.2 Baseline Details

LVR [11] is a novel multimodal reasoning paradigm that overcomes the limitations of traditional text-only reasoning by enabling autoregressive reasoning directly within the visual embedding latent space. The method first projects images into a joint semantic space shared with the language model, and then trains the LLM to dynamically generate hidden states that reconstruct key visual tokens critical for answering queries. Its training pipeline consists of two stages: Supervised Fine-Tuning using a MSE loss for visual feature reconstruction, and Reinforcement Learning employing an adapted GRPO algorithm tailored for self-evolution in latent reasoning.

Skila [13] is a unified multimodal reasoning paradigm that expands the autoregressive capabilities of MLLMs to natively generate continuous visual embeddings, termed latent sketch tokens, as visual thoughts. During inference, the method dynamically alternates between a textual thinking mode for generating discrete text and a visual sketching mode for generating continuous features. The training pipeline employs a latent visual semantics reconstruction mechanism, leveraging an auxiliary sketch encoder to extract features from intermediate sketch images as MSE reconstruction targets, ensuring that the generated latent sketch tokens are strictly semantically grounded.

Monet [12] is a training framework that empowers MLLMs to perform abstract reasoning in the latent visual space by generating continuous embeddings as intermediate "visual thoughts," thereby eliminating the reliance on predefined external visual tools. The pipeline first utilizes a three-stage distillation-based Supervised Fine-Tuning process to mitigate the high computational cost of latent-visual alignment. Subsequently, to overcome the inability of standard GRPO to optimize continuous latent features, it introduces VLPO (Visual-latent Policy Optimization), a novel reinforcement learning method that explicitly incorporates latent embeddings into policy gradient updates by estimating their approximate output probabilities.

CoVT [15] is a framework that guides VLMs to think using continuous visual tokens, aiming to enrich the model with fine-grained, dense visual perception capabilities such as depth, segmentation, and edges. The pipeline trains the VLM to autoregressively predict these continuous visual tokens, aligning them with underlying perceptual signals using lightweight visual expert task decoders e.g. SAM, DepthAnything, PIDINet, DINO guided by reconstruction losses. During inference, the model directly constructs a chain of visual thoughts in the continuous token space, while supporting the optional decoding of these tokens back into dense prediction maps for human interpretability.

B.3 SFT Training

We build UniVLR on top of the `Qwen2.5-VL-7B-Instruct` backbone. Throughout all training stages, the pre-trained vision tower and patch-merger are strictly frozen to preserve the original visual representation capabilities. We conduct full-parameter fine-tuning on the LLM backbone and optimize a newly initialized Latent Visual Reasoning projection head.

UniVLR Projection Head and Alignment Layer. Unlike standard multimodal models that project visual features into the language space at the input layer, we extract the LLM's decoder hidden states to align with the visual target space. We empirically find that aligning intermediate layers e.g., the 14th layer out of 28 layers yields more robust spatial and reasoning features than the final layer. The extracted hidden state h_k is then passed through our UniVLR projection head, which is instantiated as a lightweight Multi-Layer Perceptron (MLP) consisting of: LayerNorm $\rightarrow$ Linear $\rightarrow$ GELU $\rightarrow$ Linear.

Training Sequence and Special Tokens. We introduce four special control tokens to regulate the latent reasoning process: <|univlr_start|>, <|univlr|>, <|univlr_end|>, and <|univlr_latent_end|>. To leverage pretrained priors, we initialize their embeddings using the existing visual boundary tokens e.g. <|im_start|> for <|univlr_start|>, <|image_token|> for <|univlr|>, and <|im_end|> for the end tokens. A complete reasoning trajectory on the assistant side is formatted as:

$$\texttt{<|univlr\_start|>}\underbrace{\texttt{<|univlr|>}\ldots\texttt{<|univlr|>}}_{K=24}\texttt{<|univlr\_end|><|univlr\_latent\_end|>}\,\mathcal{A}$$

Masking Strategy for Language Modeling. During supervised fine-tuning, the standard Cross-Entropy loss $\mathcal{L}_{\text{CE}}$ is applied exclusively to the textual answer $\mathcal{A}$ and the structural boundary tokens (<|univlr_start|> and <|univlr_end|>), teaching the model when to enter and exit the latent mode. We strictly mask out the prompt tokens, padding tokens, the intermediate <|univlr|> placeholders, and the <|univlr_latent_end|> token from the CE loss by setting their labels to `IGNORE_INDEX`.

Latent Alignment Objective. For each <|univlr|> position, the prediction is aligned with the visual target representation extracted by the frozen vision tower. To optimize both the magnitude and directional geometry of the high-dimensional latent states, we define the alignment loss $\mathcal{L}_{\text{align}}$ as a combination of Layer-Normalized Mean Squared Error and Cosine Similarity:

$$\mathcal{L}_{\text{align}} = \frac{1}{K}\sum_{k=1}^{K}\Big[\|\text{LN}(\hat{\mathbf{z}}_k) - \text{LN}(\tilde{\mathbf{z}}_k)\|_2^2 + \big(1 - \cos(\text{LN}(\hat{\mathbf{z}}_k), \text{LN}(\tilde{\mathbf{z}}_k))\big)\Big] \tag{10}$$

where $\text{LN}(\cdot)$ denotes Layer Normalization, $\hat{\mathbf{z}}_k$ is the predicted latent token, and $\tilde{\mathbf{z}}_k$ is the offline precomputed visual target obtained via aspect-aware 2D average pooling, denoted as `pool_avg` in our codebase. The total loss is defined as $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CE}} + \lambda_{\text{lvr}}\mathcal{L}_{\text{align}}$, where λ_{lvr} is set to 0.1.

Latent Teacher Forcing. To prevent error compounding during the early stages of autoregressive latent generation, we enforce full *Latent Teacher Forcing* across the standard curriculum. Specifically, during the forward pass, the input embeddings corresponding to the <|univlr|> tokens are directly replaced by the ground-truth visual targets $\tilde{\mathbf{Z}}$. Consequently, the autoregressive context used to predict the $(k+1)$-th latent state explicitly conditions on the perfect k-th visual target, ensuring training stability.

Hyperparameters All experiments are conducted using DeepSpeed ZeRO-3 optimization on 4 NVIDIA A100 GPUs. We adopt Flash Attention 2 to accelerate training. The total batch size is strictly maintained at 64 via gradient accumulation with a per-device batch size of 1. The complete set of SFT training hyperparameters is summarized in Table 3.

Table 3: Hyperparameters for UniVLR SFT Training (Stage I and Stage II).

Hyperparameter	Value
LLM Backbone	Qwen2.5-VL-7B-Instruct
Precision	`bfloat16`
Global Batch Size	64
Per-Device Batch Size	1
Optimizer	AdamW
Learning Rate (LLM Backbone)	1×10^{-5}
Learning Rate (UniVLR Head)	1×10^{-4}
Learning Rate Scheduler	Cosine
Warmup Ratio	0.05
Weight Decay	0.1
Max Gradient Norm	1.0
Epochs	1 (per stage)
Training latent budget K_{train}	24
Inference latent budget K_{infer}	12
Loss Coefficient λ_{lvr}	0.1
Alignment Target Layer	14
Image Token Max Pixels	$5120 \times 28 \times 28$
Image Token Min Pixels	$128 \times 28 \times 28$
DeepSpeed Stage	ZeRO-3

B.4 Detailed Evaluation Setup

We use VLMEvalKit [33] for all our benchmark evaluations. To accommodate the unique inference mechanism of our model, we implement a customized model wrapper that extends the standard evaluation pipeline with specific decoding support for our unified visual latent tokens.

Benchmarks. We evaluate our models on a diverse set of perception and reasoning tasks. Specifically, we evaluate guided visual search and spatial reasoning on **V*** [2], high-resolution fine-grained perception on **HRBench4K** and **HRBench8K** [3], and real-world multimodal comprehension on **MME-RealWorld-Lite** [5]. We strictly follow the evaluation protocols and data splits defined by each respective benchmark.

Decoding Strategy. During evaluation, the model is configured to use our `UniVLR` decoding strategy. The decoding process operates deterministically i.e. greedy decoding without relying on explicit sampling parameters such as `temperature` or `top_p`, unless overridden by specific generation configurations.

Crucially, the decoder enforces a structured transition between the latent and textual spaces. The model first generates a predefined number of continuous visual latent blocks, bounded by the <|univlr_start|> and <|univlr_end|> control tokens. Upon completing the allocated latent

budget, the model emits the <|univlr_latent_end|> token and seamlessly switches to standard autoregressive text generation to produce the final natural-language answer.

In our main experiments, we use $K_{\text{infer}} = 12$ continuous latent tokens for inference, while the model is trained with $K_{\text{train}} = 24$ teacher-forced latent targets. In implementation, the inference wrapper truncates the latent generation phase to the first 12 latent positions before switching to answer decoding. To ensure fair evaluation by the external judges, we employ a post-processing step `clean_univlr_output=True` that strips all internal latent markers and unreadable token placeholders, outputting only the final textual answer.

System Prompts. Unlike many existing models that require heavily engineered instructions, we evaluate our model in a zero-shot manner without inserting any explicit system prompts e.g. "You are a helpful visual reasoning assistant.". The input to the model consists purely of the standard dataset prompt interleaved with the visual input, formatted automatically by the base model's chat template.

Judging Protocol. Following recent standard practices in evaluating multimodal reasoning, we adapt an LLM-as-a-Judge pipeline for performance assessment. For the V*, HRBench4K, and HRBench8K benchmarks, we employ `gpt-4o-2024-08-06` as the external judge to assess the correctness of the generated textual answers against the ground truth. For MME-RealWorld-Lite, we utilize the standard rule-based exact-match judging protocol provided by the benchmark.

B.5 Rendering Strategy Algorithm

To robustly convert multi-step explicit reasoning traces and multi-modal auxiliary images into unified visual canvases, we propose three distinct rendering algorithms. Unlike naive fixed-size rendering which may cause severe text truncation or excessive blank areas, our strategies dynamically adjust the canvas dimensions and component layouts based on the content. The specific strategy applied depends on the reasoning step pattern and the dataset construction stage.

Table 4 summarizes the three rendering strategies and their core parameter configurations.

Table 4: Overview of MCoT Canvas Rendering Strategies and Configurations.

Strategy	Key Characteristics	Applicable Patterns	Core Hyperparameters
Vertical Layout (Algorithm 1)	Dynamic height, sequential top-to-bottom cards.	Long sequential reasoning traces.	`min_canvas_width`: 420px, `outer_padding`: 24px, `gap`: 18px.
Compact Left-Right Layout (Algorithm 2)	Dynamic height, left image panel, right text cards with directed flow arrows.	`(t,j,t)` or `(t,j,t,t)` patterns with complex spatial focus.	`canvas_width`: 1536px, `max_body_height`: 1180px, `column_gap`: 28px.
Fixed-Canvas Adaptive Wrap (Algorithm 3)	Fixed $W \times H$ canvas, bottom-left image, adaptive font-size text wrapping.	General visual grounding and baseline comparisons.	`canvas_width`: 1024px, `canvas_height`: 1024px, `font_range`: $[14, 80]$px.

1. Vertical Layout Strategy. This strategy is utilized to render lengthy reasoning traces where components naturally follow a sequential logic. As outlined in Algorithm 1, the canvas width is constrained by a minimum threshold and the scaled auxiliary image. Each textual or visual reasoning block is rendered into a rounded card, and the final canvas height is dynamically computed to accommodate all vertically stacked cards. Connecting arrows are drawn between successive cards to indicate logical progression.

2. Compact Left-Right Layout Strategy. For reasoning patterns involving a central visual grounding step surrounded by textual analysis (e.g., text $\rightarrow$ joint-image $\rightarrow$ text), we employ a compact left-right architecture. As detailed in Algorithm 2, the canvas has a fixed width but dynamic height. The left panel is dedicated to the highlighted auxiliary image and its localized description, while the right panel vertically stacks the pure-text reasoning cards. Directional arrows are rendered across columns to trace the multimodal logical dependencies explicitly.

3. Fixed-Canvas Adaptive Wrap Strategy. To ensure strict resolution control during certain experimental settings, we design a fixed-canvas strategy. As described in Algorithm 3, the canvas dimensions are rigidly constrained (e.g., 1024×1024). The auxiliary image is scaled and anchored to the bottom-left corner. The textual reasoning is then rendered using an adaptive font-size search

Algorithm 1 Vertical Layout Strategy

Input: Ordered reasoning blocks $\mathcal{B} = \{b_1, \ldots, b_N\}$, Auxiliary image I, min_width $W_{\min}$, padding p, gap g.
Output: Unified Canvas C_{vert}.

Determine Canvas Width:
Highlight missing/focus regions on I to obtain I'.
$W_{\text{canvas}} \leftarrow \max(W_{\min}, I'_{width} + 2 \times p)$
Construct Component Cards:
Initialize card list $\mathcal{L}_{\text{cards}} \leftarrow []$
for each block $b \in \mathcal{B}$ **do**
  if b is a joint image block **then**
    Render card C_b containing I' and block text, constrained to $W_{\text{canvas}} - 2p$.
  else
    Render textual card C_b containing block text, constrained to $W_{\text{canvas}} - 2p$.
  end if
  Append C_b to $\mathcal{L}_{\text{cards}}$.
end for
Dynamic Canvas Assembly:
$H_{\text{canvas}} \leftarrow 2p + \sum_{C_b \in \mathcal{L}_{\text{cards}}} \text{height}(C_b) + g \times (N - 1)$
Initialize blank canvas C_{vert} with size $(W_{\text{canvas}}, H_{\text{canvas}})$.
$Y_{cursor} \leftarrow p$
for $i = 1$ **to** N **do**
  Paste $\mathcal{L}_{\text{cards}}[i]$ onto C_{vert} at (p, Y_{cursor}).
  if $i < N$ **then**
    Draw a vertical directed arrow from the bottom of $\mathcal{L}_{\text{cards}}[i]$ to the top of $\mathcal{L}_{\text{cards}}[i+1]$.
  end if
  $Y_{cursor} \leftarrow Y_{cursor} + \text{height}(\mathcal{L}_{\text{cards}}[i]) + g$
end for
Return C_{vert}.

mechanism. The text wraps dynamically: lines above the image utilize the full canvas width, while lines adjacent to the image wrap exclusively within the remaining right-side space, perfectly avoiding the image boundaries.

B.6 Rendering Strategy Examples

To provide an intuitive understanding of our adaptive rendering algorithms, we present qualitative examples of the unified MCoT canvases generated by each of the three strategies. As demonstrated below, our rendering module effectively accommodates diverse auxiliary image aspect ratios and textual reasoning lengths, ensuring that the resulting visual supervision targets are structurally coherent and semantically dense.

C UniVLR-SFT-140K Construction

A core contribution of our work is constructing high-quality, dense visual supervision signals to train the latent reasoning interface. The unfiltered data may introduce shortcut-prone supervision, weakening the visual-latent mapping established in Stage I. By retaining 21.5K filtered Zebra-CoT samples, we reduce noisy supervision and encourage a stronger dependency between the unified canvas targets and final answers.

C.1 Data Statistics

Our UniVLR-SFT-140K dataset is primarily curated from the Visual-CoT subset of Monet-SFT-125K [12] and highly structured subsets of Zebra-CoT [34]. To effectively implement our two-stage curriculum, we utilize different data compositions in each stage:

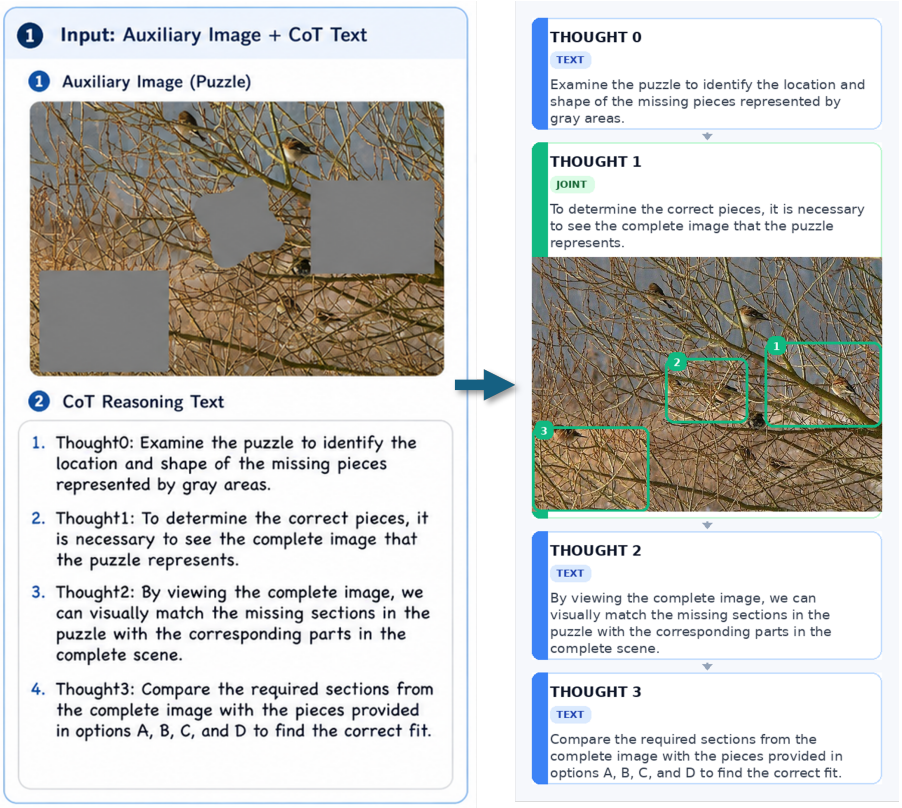

Figure 6: **Example of the Vertical Layout Strategy.** Generated by Algorithm 1, this layout dynamically extends the canvas height to accommodate a long sequence of reasoning steps.

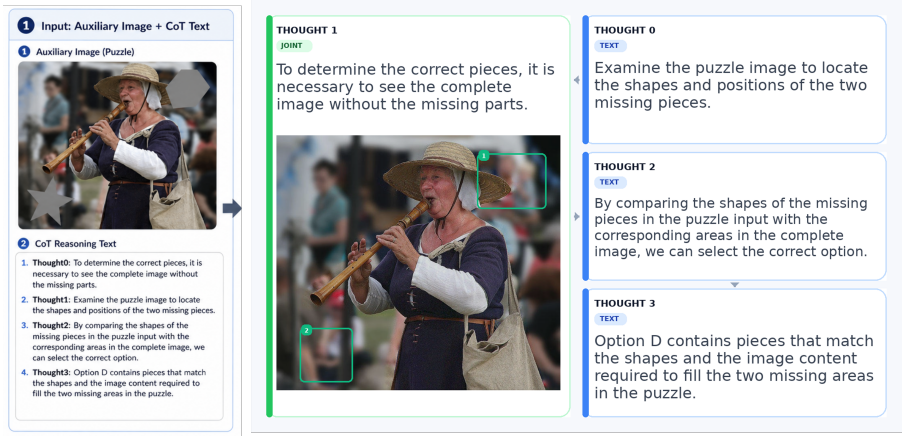

Figure 7: **Example of the Compact Left-Right Layout Strategy.** Generated by Algorithm 2, this layout partitions the canvas into two columns. The left panel houses the highlighted auxiliary image (serving as the visual grounding anchor), while the right panel vertically stacks the text-only reasoning cards. Directional flow arrows bridge the two columns, visually encoding the multimodal dependencies (e.g., text $\rightarrow$ joint $\rightarrow$ text).

Algorithm 2 Compact Left-Right Layout Strategy

Input: Text blocks $\mathcal{B}_{\text{text}}$, Joint block b_{joint}, Auxiliary image I, canvas_width W, max_body_height $H_{\max}$, margin m, column_gap g_{col}.
Output: Unified Canvas C_{compact}.

Column Partitioning:
$W_{\text{left}} \leftarrow (W - 2m - g_{\text{col}})/2$
$W_{\text{right}} \leftarrow W - 2m - g_{\text{col}} - W_{\text{left}}$
Left Panel Construction:
Highlight focus regions on I to obtain I'.
Resize I' to fit within W_{left} and $H_{\max}$.
Calculate joint text height $H_{\text{j-text}}$ and assemble the left joint panel P_{left}.
Right Panel Construction & Height Synchronization:
Calculate base height $H_{\text{body}} \leftarrow \text{height}(P_{\text{left}}) + \text{offset}$
Determine uniform row height H_{row} for right-side text cards based on H_{body} and number of text blocks $|\mathcal{B}_{\text{text}}|$.
Recalculate final H_{canvas} to perfectly align both columns.
Initialize blank canvas C_{compact} with size (W, H_{canvas}).
Assembly and Dependency Tracking:
Paste P_{left} at (m, m). Record its bounding box.
$Y_{cursor} \leftarrow m$
for each block $b_t \in \mathcal{B}_{\text{text}}$ **do**
    Render textual card C_t with size $(W_{\text{right}}, H_{\text{row}})$.
    Paste C_t at $(m + W_{\text{left}} + g_{\text{col}}, Y_{cursor})$. Record bounding box.
    $Y_{cursor} \leftarrow Y_{cursor} + H_{\text{row}} + \text{row\_gap}$
end for
Draw horizontal and vertical arrows between recorded bounding boxes based on the original chronological thought sequence
Return C_{compact}.

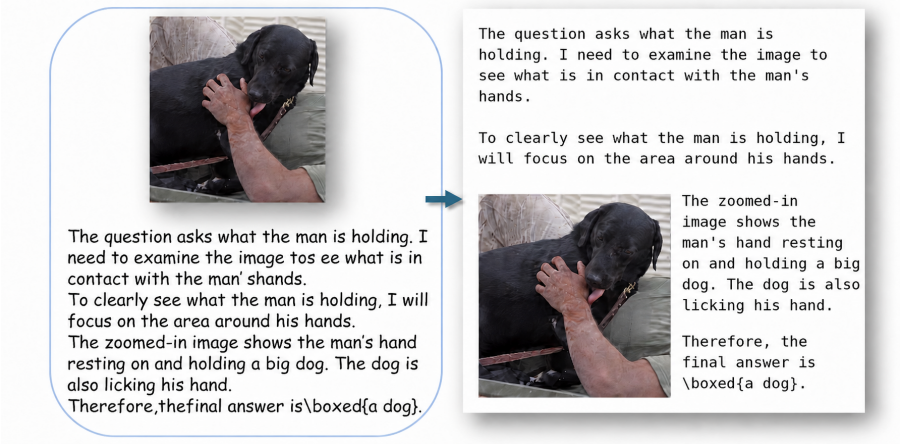

Figure 8: **Example of the Fixed-Canvas Adaptive Wrap Strategy.** Generated by Algorithm 3, the canvas dimensions are strictly bounded 1024×1024.

Algorithm 3 Fixed-Canvas Adaptive Wrap Strategy

Input: Textual reasoning trace $\mathcal{T}$, Auxiliary image I, Fixed canvas size (W, H), padding p, gap g.
Output: Unified Canvas C_{fixed}.

Image Anchor Placement:
Scale I such that it occupies at most 50% of canvas width and height.
Place I at the bottom-left corner: $X_{img} \leftarrow p$, $Y_{img} \leftarrow H - p - I_{height}$.
Adaptive Font Search & Text Wrapping:
Initialize optimal font $f_{\text{opt}} \leftarrow$ None.
for font size $f \in \{f_{\max}, f_{\max} - 1, \ldots, f_{\min}\}$ **do**
  $Y_{cursor} \leftarrow p$
  Initialize current line text.
  for each word $w \in \mathcal{T}$ **do**
    Dynamic Width Constraint:
    if $Y_{cursor} \geq Y_{img}$ **and** $Y_{cursor} \leq Y_{img} + I_{height}$ **then**
      $W_{\text{avail}} \leftarrow W - (X_{img} + I_{width} + g) - p$
    else
      $W_{\text{avail}} \leftarrow W - 2p$
    end if
    Test if appending w exceeds W_{avail}. If yes, commit line and advance $Y_{cursor} \leftarrow Y_{cursor} + f + (f/4)$.
  end for
  if $Y_{cursor} \leq H - p$ **then**
    $f_{\text{opt}} \leftarrow f$ {Found the largest font that fits all text.}
    Break
  end if
end for
If f_{opt} is not found, fallback to $f_{\min}$ and truncate overflowing text.
Initialize blank canvas C_{fixed} with size (W, H).
Paste I at (X_{img}, Y_{img}) and draw text $\mathcal{T}$ using the computed layout boundaries.
Return C_{fixed}.

- **Stage I (Latent Warm-up):** We utilize the entire Visual-CoT dataset, comprising 118K samples, as the foundational training corpus. This dataset, featuring abstract visual operations such as bounding boxes, serves to establish the initial mapping between the continuous latent space and the structural topology of the unified visual canvases.

- **Stage II (Curriculum Mixing):** We curate a high-quality mixed dataset containing approximately 30.7K instances. This mixture consists of all 21.5K rigorously filtered samples from three Zebra-CoT subsets (Visual Search, Jigsaw, and Maze) and a randomly sampled portion of the Visual-CoT dataset, maintaining a 7:3 ratio between the complex Zebra-CoT data and the foundational Visual-CoT data.

The detailed statistics, problem domains, and visual operation types of our final UniVLR-SFT-140K dataset are summarized in Table 5.

Table 5: Statistics of the UniVLR-SFT-140K. The dataset supports our two-stage curriculum, utilizing the full Visual-CoT for Stage I and a 7:3 mixture of heavily filtered Zebra-CoT subsets and sampled Visual-CoT for Stage II.

Data Source	Problem Domain	Visual Operation Type	Amount
Visual-CoT	Real-world, Documents, Charts	Abstract visual operations (e.g., bounding boxes)	118.6K
Zebra-CoT Visual Search	2D Visual Reasoning	Object localization, local focus highlighting	8.7K
Zebra-CoT Jigsaw	2D Visual Reasoning	Spatial geometry, piece alignment	8.8K
Zebra-CoT Maze	Visual Logic & Strategic Games	Pathfinding, spatial constraint satisfaction	4.0K
Total (UniVLR-SFT-140K)			**140.1K**

Comparison with Existing Latent Reasoning Datasets. To contextualize our data curation efforts, we compare the scale and composition of our training corpora with recent state-of-the-art latent and visual reasoning models.

- **Monet** [12] utilizes Monet-SFT-125K, which predominantly consists of Visual-CoT alongside smaller portions of ReFocus, CogCoM, and Zebra-CoT.
- **CoVT** [15] constructs a substantial dataset derived primarily from LLaVA-OneVision vision-centric subsets and a filtered TallyQA counting dataset.
- **LVR** [11] employs the full Visual-CoT dataset (438K instances) for its supervised fine-tuning stage.
- **SkiLa** [13] curates a 101K sample dataset entirely filtered from Zebra-CoT, excluding 3D data and excessively complex sketch images.

Our UniVLR-SFT-140K distinctively organizes its 140K samples into a two-stage curriculum, prioritizing the quality and structural regularity of the visual supervision signals over raw scale.

C.2 Data Filtering

To ensure that the latent tokens actively learn to encode essential visual semantics rather than memorizing textual shortcuts, and to maintain the high quality of the visual targets, we design a rigorous filtering pipeline consisting of bidirectional model-based filtering and geometric layout filtering.

Step 1: Lower-Bound Filtering. Many instances in standard multimodal reasoning datasets appear complex but can actually be answered correctly using pure linguistic priors or shallow visual scanning, without needing the intermediate visual reasoning steps. Training on these instances encourages the latent tokens to bypass genuine visual perception. To eliminate this, we feed the raw *Question* and *Problem Image* without any auxiliary reasoning canvas into **Qwen2.5-VL-7B**. If the model correctly answers the question zero-shot, we discard the sample, retaining only those that strictly require intermediate visual reasoning to solve.

Step 2: Upper-Bound Filtering. Conversely, some reasoning traces contain severe logical flaws or mismatched auxiliary images, which corrupt the latent alignment target. To filter these out, we feed the *Question*, *Problem Image*, and the *Unified MCoT Canvas* or auxiliary images into a much stronger teacher model, **Qwen2.5-VL-72B**. If this powerful teacher still fails to derive the correct answer despite having access to the explicit visual reasoning steps, we deem the sample unsolvable or noisy and discard it.

Step 3: Aspect-Ratio Filtering for Canvas Density. Besides semantic correctness, the geometric properties of the auxiliary images significantly impact the quality of our unified visual representations. Auxiliary images with extreme aspect ratios i.e., excessively long or wide force our rendering algorithm to apply aggressive scaling and padding to fit within the canvas boundaries. This inevitably introduces massive whitespace regions onto the unified canvas, which dilutes the information density of the extracted visual features and wastes the limited capacity of the continuous latent tokens. Therefore, we filter out samples containing auxiliary images with extreme aspect ratios, ensuring that the rendered MCoT canvas remains compact, semantically dense, and visually balanced for the vision encoder.

Impact of Filtering on Latent Alignment. Our empirical observations as detailed in Section 3.5 confirm that this bidirectional filtering is critical for stable Stage II training. When we attempted to mix the raw, unfiltered Zebra-CoT data $\sim$40K instances with Visual-CoT, the downstream accuracy degraded. The unfiltered data diluted the strong visual-latent mapping established in Stage I because the fake hard samples taught the model to exploit spurious correlations, causing the latent representation to collapse. By retaining only the rigorously filtered $21.5K$ Zebra-CoT samples, we enforce a strict causal dependency between the visual canvas target and the final answer, significantly improving the efficacy of the unified visual latent reasoning.

D Additional Experimental Results

In this section, we provide additional experimental results to complement the analyses in the main paper. These experiments further examine the sensitivity of UniVLR to the latent-alignment loss weight, its generalization behavior beyond the core visual reasoning benchmarks, and the effect of latent teacher forcing during continuous autoregressive training. Unless otherwise specified, we follow the same evaluation protocol as in the main text and report results using the corresponding UniVLR checkpoint described in each subsection.

D.1 Ablation Study for λ_{align}

UniVLR is trained with a combination of the standard autoregressive language modeling objective and the latent representation alignment objective. The coefficient λ_{align} controls the relative strength of the visual latent supervision. A very small value may provide insufficient alignment to the visual representation space, whereas an overly large value may over-constrain the latent states and reduce their flexibility for downstream answer prediction. We therefore vary $\lambda_{\text{align}} \in \{0.1, 0.3, 0.5, 0.7\}$ and report the results in Table 6.

Overall, $\lambda_{\text{align}} = 0.1$ provides the best accuracy–stability trade-off across the evaluated benchmarks. Increasing the alignment weight generally leads to performance degradation, especially on MME-RealWorld-Lite. This suggests that, while visual latent alignment is important, excessive alignment pressure can make the latent trajectory less adaptive to task-specific reasoning and answer decoding.

Table 6: Ablation study for λ_{align}. We vary the weight of the latent alignment objective while keeping the remaining training and evaluation settings unchanged.

Model	V*			HRBench4K			HRBench8K			MME-RealWorld-Lite		
	Overall	Attr.	Spa.	Overall	FSP	FCP	Overall	FSP	FCP	Overall	Rea.	Perc.
$\lambda_{\text{align}} = 0.1$	**82.7**	**83.5**	**81.6**	**73.3**	**86.0**	**60.5**	**68.8**	**78.8**	**58.8**	**50.7**	**44.7**	**54.5**
$\lambda_{\text{align}} = 0.3$	82.7	83.5	81.6	72.8	86.0	59.5	66.6	75.5	57.8	46.9	42.0	50.1
$\lambda_{\text{align}} = 0.5$	81.7	83.5	78.9	72.5	85.5	59.5	65.3	75.0	55.5	25.4	22.4	27.3
$\lambda_{\text{align}} = 0.7$	82.2	82.6	81.6	71.8	85.5	58.0	65.5	75.3	55.8	29.0	28.8	29.2

D.2 Generalization on Diverse Multimodal Tasks

A potential concern when fine-tuning MLLMs for specialized latent visual reasoning is catastrophic forgetting: the model may improve on the target reasoning tasks but lose general multimodal capabilities. To examine this issue, we evaluate UniVLR on a broader set of benchmarks that are not exclusively designed for complex visual latent reasoning. Specifically, we consider **TextVQA** [35] for text-rich image understanding, **MME$_{\text{translation}}$** [36] for multilingual multimodal alignment, **POPE** [37] for object hallucination evaluation, and **WeMath$_{\text{loose}}$** [38] for abstract mathematical reasoning.

The results are summarized in Table 7. Overall, UniVLR does not exhibit severe degradation on general-purpose multimodal tasks. Instead, it improves over the base model on TextVQA and POPE, while maintaining competitive performance on MME$_{\text{translation}}$ and WeMath$_{\text{loose}}$.

Perceptual grounding and hallucination mitigation. UniVLR achieves the best performance on TextVQA and POPE among the compared models. The improvement on TextVQA suggests that the unified visual latent training does not weaken text-rich image understanding; instead, it can improve the model's ability to extract and use visual textual evidence. On POPE, UniVLR consistently improves over the base Qwen2.5-VL-7B model across the adversarial, popular, and random splits. Since POPE is designed to measure object hallucination, these gains indicate that the learned visual latent channel may encourage the model to rely more strongly on grounded visual evidence rather than language priors.

We attribute this behavior to the proposed unified canvas supervision. During training, UniVLR aligns latent reasoning states with dense visual representations that jointly encode textual reasoning traces and auxiliary visual evidence. This encourages the latent tokens to remain anchored to the

Table 7: Generalization performance on diverse multimodal benchmarks. We evaluate whether UniVLR preserves general multimodal capabilities beyond the core visual reasoning tasks. Best results are highlighted in **bold**.

Model	TextVQA	$\text{MME}_{\text{translation}}$	POPE				$\text{WeMath}_{\text{loose}}$
			Overall	Adversarial	Popular	Random	
Qwen2.5-VL-7B [23]	77.5	185.0	86.4	85.5	86.5	87.2	**52.1**
LVR [11]	75.6	200.0	84.9	84.3	84.8	85.7	48.4
SkiLa [13]	79.3	**200.0**	87.1	86.2	87.2	87.9	46.2
CoVT [15]	66.5	192.5	88.7	87.2	88.6	90.3	50.1
UniVLR-Stage1	79.1	183.0	84.2	83.7	84.2	84.8	48.6
UniVLR (Ours)	**80.4**	**200.0**	**88.8**	**87.3**	**88.9**	**90.4**	49.4

visual input, which may reduce unsupported object predictions and improve recognition of text and fine-grained visual content.

Generalization on abstract reasoning tasks. UniVLR also remains competitive on tasks that are less directly tied to visual latent reasoning. It achieves a perfect score of 200.0 on $\text{MME}_{\text{translation}}$, indicating that the fine-tuning process does not degrade multilingual multimodal alignment in this setting. On $\text{WeMath}_{\text{loose}}$, UniVLR shows a moderate decrease compared with the base Qwen2.5-VL-7B model, from 52.1 to 49.4. This suggests a small specialization trade-off introduced by the visual reasoning curriculum. Nevertheless, UniVLR remains competitive with other visual latent reasoning baselines, outperforming LVR and SkiLa on this benchmark. These results suggest that UniVLR improves visual grounding and hallucination robustness without causing severe degradation in general multimodal capabilities.

D.3 Ablation Study on Latent Teacher Forcing

Continuous latent reasoning introduces a training–inference discrepancy. During training, the model can condition on target latent embeddings, whereas during inference it must recursively condition on its own predicted latent states. In discrete autoregressive generation, scheduled sampling [39] is often used to reduce such exposure bias by gradually replacing ground-truth inputs with model predictions. We investigate whether this strategy is also beneficial for continuous visual latent generation.

We compare full latent teacher forcing with a scheduled sampling variant during Stage II training. In the full teacher-forcing setting, all $K = 24$ latent inputs are replaced with the corresponding ground-truth visual targets. In the scheduled sampling variant, teacher forcing is applied only to the first 12 latent slots, while the remaining 12 slots use the model's detached predictions as inputs, which we refer to as half-replay. The results are reported in Table 8.

Table 8: Effect of latent teacher forcing and scheduled sampling. Experiments are conducted on the Stage II model. Full latent teacher forcing yields more stable training and better downstream accuracy across most benchmark metrics.

Training Strategy	V*			HRBench4K			HRBench8K			MME-RealWorld-Lite		
	Overall	Attr.	Spa.	Overall	FSP	FCP	Overall	FSP	FCP	Overall	Rea.	Perc.
Full Latent Teacher Forcing	**82.7**	**83.5**	**81.6**	**73.3**	**86.0**	**60.5**	**68.8**	**78.8**	**58.8**	**50.7**	44.7	**54.5**
Scheduled Sampling (Half-Replay)	82.2	82.6	**81.6**	72.4	85.0	59.8	66.8	76.5	57.0	48.8	**44.9**	51.2

Analysis. Contrary to the common intuition from discrete text generation, scheduled sampling does not improve performance in our continuous latent reasoning setting. Compared with full latent teacher forcing, half-replay reduces the overall score by 2.0 points on HRBench8K and 1.9 points on MME-RealWorld-Lite. It also underperforms on most fine-grained subcategories, with the only exception being a small gain on the reasoning split of MME-RealWorld-Lite.

We hypothesize that this behavior arises from the geometry of high-dimensional continuous latent spaces. In early training, the model's predicted latent vectors may not yet lie on a stable visual representation manifold. Feeding these imperfect continuous predictions back into the model can

therefore inject semantic noise into the autoregressive context, which in turn affects subsequent latent predictions and weakens the alignment objective. By contrast, full latent teacher forcing provides a cleaner and more stable supervision signal at every latent step, making representation learning easier and leading to stronger downstream performance. This result suggests that, for visual latent reasoning, stabilizing the latent manifold during training can be more important than directly reducing exposure bias through self-replay.

D.4 Additional Efficiency Evaluation

We provide an additional efficiency evaluation on two representative benchmarks, HRBench8K and MME-RealWorld-Lite. The goal of this experiment is to complement the reasoning-token analysis in the main paper with a lightweight runtime and memory measurement under the same evaluation pipeline.

We note that output-token throughput is not an ideal metric for this setting. Visual latent reasoning methods often generate very short final textual answers, and UniVLR uses only a compact latent reasoning budget before answer decoding. Consequently, output tokens per second can be dominated by fixed evaluation overheads and small variations in answer length. We therefore report average per-sample time, total sample time, peak allocated GPU memory, and peak reserved GPU memory. All methods are evaluated with the same benchmark wrapper and logging protocol.

Table 9: Additional efficiency comparison on HRBench8K and MME-RealWorld-Lite. We report average per-sample time, total sample time, and peak allocated/reserved GPU memory. All values are rounded to one decimal place. The relative reductions over the Qwen2.5-VL-7B baseline are shown with ↓.

Dataset	Method	# Samples	Avg. Time / Sample (s)	Total Sample Time (s)	Peak Alloc. Mem. (GB)	Peak Reserved Mem. (GB)
HRBench8K	Qwen2.5-VL-7B (Base)	800	6.8	5453.3	18.9	21.6
	UniVLR (Ours)	800	**4.5** ↓33.8%	**3620.1** ↓33.6%	**17.4** ↓7.9%	**18.9** ↓12.5%
	CoVT	800	9.4	7559.1	19.0	22.0
	LVR	800	5.0	4024.0	17.4	18.9
	SkiLa	800	9.4	7546.4	17.4	18.9
MME-RealWorld-Lite	Qwen2.5-VL-7B (Base)	1919	4.3	8251.7	19.0	21.7
	UniVLR (Ours)	1919	**3.2** ↓25.6%	**6076.7** ↓26.4%	**17.4** ↓8.4%	**18.9** ↓12.9%
	CoVT	1919	4.5	8664.1	19.0	22.0
	LVR	1919	3.6	6859.8	17.4	18.9
	SkiLa	1919	4.7	8966.3	17.5	18.9

Table 9 shows that UniVLR provides favorable runtime efficiency under the actual evaluation setting. On HRBench8K, UniVLR reduces the average per-sample time from 6.8 seconds for Qwen2.5-VL-7B to 4.5 seconds, corresponding to a 33.8% reduction. The total sample time is also reduced from 5453.3 seconds to 3620.1 seconds. Compared with prior visual latent reasoning methods, UniVLR is faster than LVR and substantially faster than CoVT and SkiLa, which require 9.4 and 9.4 seconds per sample, respectively.

A similar trend is observed on MME-RealWorld-Lite. UniVLR reduces the average per-sample time from 4.3 seconds for the base model to 3.2 seconds, and reduces the total sample time from 8251.7 seconds to 6076.7 seconds. Among visual latent reasoning baselines, UniVLR is also faster than CoVT, LVR, and SkiLa under the same evaluation protocol.

UniVLR further maintains a compact memory footprint. On HRBench8K, it reduces peak allocated memory from 18.9 GB to 17.4 GB and peak reserved memory from 21.6 GB to 18.9 GB compared with the base model. On MME-RealWorld-Lite, UniVLR similarly reduces peak allocated memory from 19.0 GB to 17.4 GB and peak reserved memory from 21.7 GB to 18.9 GB. Its memory usage is comparable to LVR and SkiLa, while remaining lower than CoVT on both benchmarks.

These results support the practical efficiency of the proposed unified visual latent reasoning interface. While the main paper focuses on reasoning-token efficiency, this additional evaluation shows that UniVLR does not introduce extra runtime or memory overhead in practice. Instead, the compact latent reasoning trajectory leads to lower latency and comparable or lower GPU memory usage than representative visual latent reasoning baselines. We emphasize that this experiment is intended as a

lightweight efficiency check rather than a full systems-level throughput benchmark, since standardized throughput protocols for visual latent reasoning are not yet established and final textual outputs are often very short.