LantErn: Latent Visual Structured Reasoning

Anonymous authors
Paper under double-blind review

Abstract

While language reasoning models excel in many tasks, *visual* reasoning is significantly harder. As a result, most large multimodal models (LMMs) default to verbalizing perceptual content into text, a strong limitation for tasks requiring fine-grained spatial and visual understanding. While recent approaches take steps toward *thinking with images* by invoking tools or generating intermediate images, they either rely on external modules or incur unnecessary computation by reasoning directly in pixel space.

In this paper, we introduce LantErn, a framework that enables LMMs to interleave language with compact latent visual representations, allowing visual reasoning to occur directly in latent space. LantErn augments a vision-language transformer with the ability to generate and attend to continuous visual "thought" embeddings during inference. We train the model in two stages: supervised fine-tuning to ground visual features in latent states, followed by reinforcement learning to align latent reasoning with task-level utility. We evaluate LantErn on three perception-centric benchmarks (VisCoT, $V^\star$, and Blink), observing consistent improvements in visual grounding and fine-grained reasoning. In several settings, latent visual reasoning allows smaller models to approach the performance of larger baselines, suggesting that internal latent representations provide a promising direction for more efficient multimodal reasoning.

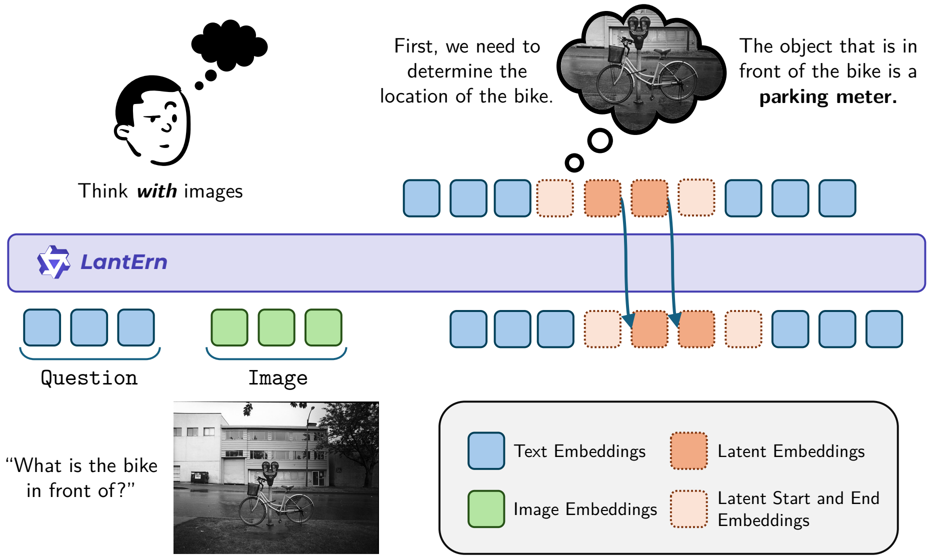
Think ***with*** images
First, we need to determine the location of the bike.
The object that is in front of the bike is a **parking meter.**
LantErn
Question
Image
"What is the bike in front of?"
Text Embeddings
Latent Embeddings
Image Embeddings
Latent Start and End Embeddings

Figure 1: The LantErn framework enables interleaved reasoning between text and latent representations that encode visual "thoughts". During inference, LantErn can automatically decide when to start latent reasoning by outputting a special token.

1 Introduction

Large Multimodal Models (LMMs) have achieved strong performance in a wide range of vision-language tasks Alayrac et al. (2022); Liu et al. (2023); Bai et al. (2025a), yet their reasoning processes remain predominantly linguistic. In most current systems, visual inputs are encoded once and all subsequent reasoning is carried out in text, a regime we refer to as *thinking about images*.
This forces high-dimensional perceptual information into a low-bandwidth symbolic medium, a limitation that becomes particularly evident on perception-heavy benchmarks, where purely textual chains of thought fail to capture fine-grained spatial and visual structure (Fu et al., 2024; Xiao et al., 2024).

To overcome these limitations, recent work has shifted toward *thinking with images*, in which visual information actively participates in the reasoning process rather than being consumed only at the input stage. Existing approaches in this category can be broadly divided into two streams. The first consists of *tool-based visual reasoning* methods, which allow models to invoke external vision modules during inference, such as cropping, object detection, or image generation tools (Yang et al., 2023; Surís et al., 2023; Team, 2025). While somewhat effective, these approaches are limited to a set of predefined tools and often incur significant computational overhead. The second stream performs reasoning by explicitly generating images during the reasoning chain, forcing intermediate visual thoughts to be expressed in pixel space and spending significant computation on photorealistic details that may be irrelevant for the task, which is wasteful.

More recently, *latent visual reasoning* has emerged as an internalized form of thinking with images, in which models maintain and manipulate continuous visual representations in latent space throughout the reasoning process (Li et al., 2025; Yang et al., 2025b). By interleaving latent visual states with text, these methods avoid explicit image generation while preserving visual structure, enabling reasoning to operate over abstract visual representations rather than pixel space.

In this work, we introduce LantErn (Latent Visual Structured Reasoning), a framework that enables MLLMs to reason using compact latent visual tokens interleaved with language. LantErn augments a vision-language transformer with the ability to emit and attend to latent visual states, allowing reasoning to occur directly in the visual feature space of the model. We train LantErn in two stages. First, we perform supervised fine-tuning on a custom dataset with annotated visual reasoning traces, grounding latent states in the outputs of the model's vision encoder. Second, we apply reinforcement learning to optimize both textual and latent reasoning as a sequential decision-making process, using final answer correctness as the reward signal.

We evaluate LantErn on challenging visual reasoning benchmarks, including Visual-CoT (Shao et al., 2024a), $V^\star$ (Cheng et al., 2025), and Blink (Fu et al., 2024). Across all settings, latent visual reasoning leads to more accurate and perceptually grounded solutions, highlighting the benefits of internal visual representations for multimodal reasoning.

2 Related Work

Text-based and tool-based visual reasoning. Early multimodal reasoning approaches extended textual chain-of-thought methods to vision-language tasks, reasoning entirely in text after a single visual encoding pass (Zhang et al., 2024; Shao et al., 2024a). To address their perceptual limitations, subsequent work introduced tool-augmented reasoning, enabling models to iteratively interact with images through external operations such as cropping, detection, and image generation (Yang et al., 2023; Surís et al., 2023; Team, 2025). While effective, these approaches depend on hand-designed tools and do not learn internal visual abstractions.

Latent visual reasoning. Latent visual reasoning aims to internalize perceptual processes by allowing models to generate continuous visual representations during inference. Li et al. (2025) propose Latent Visual Reasoning (LVR), which interleaves text with latent visual tokens. Similarly, Yang et al. (2025b) introduce machine mental imagery through latent image representations. These works show that preserving visual information in latent space can improve fine-grained reasoning. However, unlike prior approaches that primarily append latent visual representations to support downstream decoding, LantErn formulates this paradigm as an interleaved reasoning process that repeatedly alternates between text and latent visual tokens, enabling iterative refinement of internal visual representations and tighter coupling between perception and language.

3 Our Method: LantErn

LantErn provides a structured mechanism for incorporating visually grounded latent states alongside textual reasoning in LMMs. Standard LMMs are constrained to verbalize every step of visual processing, often forcing high-dimensional visual information into low-bandwidth natural language. We propose a modeling approach where the model learns to generate compressed, non-verbal "thought" vectors derived directly from visual features, interleaving these latent states with discrete text generation.

3.1 Modeling Latent Visual Reasoning

We model reasoning as a hybrid trajectory $\tau = [s_1, s_2, \ldots, s_T]$, where each state s_t lies in either the discrete vocabulary space $\mathcal{V}$ or the continuous latent space $\mathbb{R}^d$, with d denoting the model's hidden dimension. At each step t, the model outputs either a token $w_t \in \mathcal{V}$ (text mode) or a latent vector $\mathbf{z}_t \in \mathbb{R}^d$ (visual latent mode).

To implement this, we build on the Qwen2.5-VL architecture (Bai et al., 2025b) and extend its vocabulary with three control tokens: `<|lvr_start|>`, `<|lvr_sep|>`, and `<|lvr_end|>`. These tokens act as gating signals that regulate transitions between operating modes:

1. **Text Mode:** The model functions as a standard autoregressive transformer. The hidden state at time t, $\mathbf{h}_t$, is passed through the language modeling head to predict a probability distribution over the vocabulary $\mathcal{V}$.

2. **Visual Latent Mode:** Upon generating `<|lvr_start|>`, for the subsequent K time steps (where K is a fixed latent size hyperparameter), the model bypasses the language modeling head and outputs the unprojected hidden states of the final transformer layer. These hidden states are propagated internally in place of the continuous embeddings derived from text tokens (`<|lvr_sep|>`). These K vectors, denoted $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_K$ will then constitute a block of latent "thought" embeddings. After K steps a terminating token `<|lvr_end|>` is introduced to return the model to text mode. The latent vectors serve as internal reasoning context, allowing the model to attend to its own high-dimensional visual thoughts without having to verbalize that information into text.

3.2 Supervised Fine-Tuning: Grounding Latent States in Vision

A fundamental challenge in latent reasoning is defining a ground truth for the model's internal visual thoughts. Since human annotators cannot provide high-dimensional vector supervision, we devise a strategy to *ground* the latent states using the model's own visual encoder as a teacher signal.

3.2.1 Visual Feature Extraction as Supervision

We leverage the pre-trained vision encoder of the base model as a teacher for the LLM's latent states. Consider a training sample $(I, Q, A, \mathcal{T})$ consisting of an image I, a question Q, the correct answer A, and a human-written reasoning trace $\mathcal{T}$ that references certain visual regions. Let B be the set of bounding boxes in $\mathcal{T}$ corresponding to relevant regions of I that the reasoning trace attends to. For each reasoning step associated with a region $b \in B$, we construct a target latent representation $\mathbf{Z}_{\text{target}}$ as follows:

$$\mathbf{F}_b = \text{VisionEncoder}(I, b)\,, \tag{1}$$

$$\mathbf{Z}_{\text{target}} = \text{Pool}(\mathbf{F}_b) \in \mathbb{R}^{K \times d}, \tag{2}$$

where $\mathbf{F}_b$ is the feature map extracted from the vision encoder (e.g. patch embeddings) corresponding to region b. We then apply an average pooling operation to $\mathbf{F}_b$ to produce a fixed-size sequence of K pooled feature vectors. This sequence $\mathbf{Z}_{\text{target}} = [\mathbf{z}_{\text{target}}^{(1)}, \ldots, \mathbf{z}_{\text{target}}^{(K)}]$ serves as the target for the model's latent block, effectively capturing the semantic essence of the visual region b as perceived by the vision encoder.

3.2.2 Hybrid Objective Function

We train LantErn with a multi-task objective that jointly optimizes for language generation and latent visual alignment. The total loss is a sum of two components:

$$\mathcal{L}_{\text{LantErn}} = \mathcal{L}_{\text{text}} + \gamma \mathcal{L}_{\text{latent}}, \tag{3}$$

where γ is a weighting hyperparameter.

Text Generation Loss ($\mathcal{L}_{\text{text}}$). For standard (non-latent) tokens, we use a cross-entropy loss on the next-token prediction, as in conventional language model fine-tuning. This term $\mathcal{L}_{\text{text}}$ preserves the model's verbal fluency and ensures it can still articulate answers correctly in natural language.

Latent Alignment Loss ($\mathcal{L}_{\text{latent}}$). For each block of K latent tokens generated between a `<|lvr_start|>` and `<|lvr_end|>` marker, we apply a regression loss that encourages these latent vectors to match the visual encoder's features for the corresponding region of interest. Let $\mathbf{H}_{\text{gen}} = [\mathbf{h}_{\text{gen}}^{(1)}, \ldots, \mathbf{h}_{\text{gen}}^{(K)}]$ be the sequence of K hidden states produced by the LLM in latent mode, and $\mathbf{Z}_{\text{target}} = [\mathbf{z}_{\text{target}}^{(1)}, \ldots, \mathbf{z}_{\text{target}}^{(K)}]$ be the pooled target embeddings as defined above. We minimize the mean-squared error (MSE) between these two sequences:

$$\mathcal{L}_{\text{latent}} = \frac{1}{K} \sum_{i=1}^{K} \left\| \mathbf{h}_{\text{gen}}^{(i)} - \mathbf{z}_{\text{target}}^{(i)} \right\|_2^2. \tag{4}$$

This encourages the LLM to *simulate* the vision encoder's representations internally. By learning to minimize the discrepancy $||\mathbf{h}_{\text{gen}} - \mathbf{z}_{\text{target}}||^2$, the model is effectively learning to "imagine" the visual content in its latent space, reconstructing the key visual features needed to answer the question. The weighting factor γ controls the balance between this latent alignment objective and the standard language modeling objective. The goal of this phase is to distill latent visual reasoning capabilities into the LLM through explicit supervision of a predefined set of latent visual thoughts. This trains the model to perform interleaved reasoning over text and latent visual representations, grounding its explanations in internal visual states without requiring it to verbalize every aspect of its visual "thoughts." However, this supervised objective primarily enforces representational fidelity, motivating the subsequent stage to further align free-form latent reasoning with task-level utility.

3.3 Reinforcement Learning

While SFT grounds latent representations in visual features, it primarily enforces a *reconstruction* objective. This can lead to latent representations that are perceptually faithful yet suboptimal for downstream reasoning. We therefore explore Reinforcement Learning (RL) as a free-form mechanism to align latent visual reasoning with *task utility*, rather than visual fidelity alone. We formulate latent visual reasoning as a sequential decision-making problem and investigate whether policy optimization can encourage the model to generate latent states that improve final answer correctness, similar in spirit to outcome-driven RL for language models (Ouyang et al., 2022; Schulman et al., 2017). In contrast to prior work that relies purely on supervised imitation of latent traces, RL allows the model to adapt its internal reasoning strategies based solely on task-level feedback.

Policy Optimization with Hybrid Action Spaces. We adopt Group Relative Policy Optimization (GRPO) (Shao et al., 2024b) as our primary optimization algorithm. GRPO is a PPO-style method that stabilizes training by normalizing rewards within sampled groups, and has recently been shown to be effective for reasoning-heavy language model fine-tuning (Guo et al., 2025; Olmo et al., 2025). Unlike standard RL for LLMs, which operates over a discrete token space, LantErn induces a *hybrid action space*: actions correspond either to discrete text tokens or to continuous latent vectors $\mathbf{z} \in \mathbb{R}^d$. This raises a conceptual challenge, as policy gradient objectives are traditionally defined over categorical distributions. Rather than defining an explicit probability density over latent vectors, we treat latent generation as an intermediate computation that conditions subsequent text generation, following prior work on differentiable latent reasoning in language models (Li et al., 2025; Yang et al., 2025b).

Under this formulation, optimization is applied only to the likelihood of discrete text tokens, while gradients propagate through the latent states via standard backpropagation. This design implicitly encourages the model to generate latent representations that improve downstream token predictions and, ultimately, task reward.

3.3.1 Latent-Aware GRPO Objective

Given an input query q and image I, we sample a group of G rollouts $\{o_1, \ldots, o_G\}$ from the old policy $\pi_{\theta_{\text{old}}}$. Each rollout consists of interleaved text and latent blocks. We define the GRPO objective over the discrete text tokens while treating latent states as contextual conditioning variables:

$$\tilde{L}_{i,t}(\theta) = \min\Big(r_{i,t}(\theta)\hat{A}_i,\ \mathrm{clip}(r_{i,t}(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_i\Big)$$

$$\mathcal{J}(\theta) = \mathbb{E}_{\{o_i\}\sim\pi_{\theta_{\text{old}}}}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\Big(\tilde{L}_{i,t}(\theta) - \beta\, D_{\text{KL}}\big(\pi_\theta(\cdot)\,\|\,\pi_{\text{ref}}(\cdot)\big)\Big)\ \Bigg|\ q, I\right], \qquad (5)$$

where $\pi_\theta(y_{i,t}) \equiv \pi_\theta(y_{i,t} \mid q, I, \tilde{h}_i^{\text{latent}}, y_{i,<t})$, $r_{i,t}(\theta) = \pi_\theta(y_{i,t})/\pi_{\theta_{\text{old}}}(y_{i,t})$ is the importance ratio, and $\hat{A}_i$ is the group-normalized advantage.

Latent State Replay. A practical challenge arises from the fact that latent vectors are generated dynamically by the policy. Small parameter updates can lead to significant drift in latent trajectories, which destabilizes the importance sampling ratio. To mitigate this effect, we employ *latent state replay*: during policy updates, the model is forced to condition on the exact latent vectors $\mathbf{H}_{\text{rollout}}$ generated during sampling. This ensures that probability ratios reflect changes in the text policy under a fixed internal reasoning trace, while still allowing gradients to flow back to the parameters responsible for producing latent states.

3.3.2 Reward Design

Since there is no direct supervision for the quality of latent thoughts, we rely on sparse outcome-based rewards combined with structural constraints. The total reward is defined as a weighted sum:

1. **Accuracy Reward (R_{acc}):** A binary reward indicating whether the final textual answer matches the ground truth. This sparse signal serves as the primary driver of task-oriented latent reasoning, consistent with prior RL-based approaches to reasoning supervision (Ouyang et al., 2022; Li et al., 2025).
2. **Format Reward (R_{fmt}):** A structural reward that encourages the explicit use of a set of tags, such as `<think>`, `</think>` for the reasoning chain, latent reasoning delimiters `<|lvr_start|>` and `<|lvr_end|>` and `<|answer|>`, `</answer>` for the final answer. This discourages collapse to purely textual reasoning and enforces the presence of a latent block.

Overall, this RL stage is intended to test the hypothesis that outcome-driven optimization can shift latent representations from merely encoding visual appearance toward selectively representing task-critical visual information. Rather than assuming that visually faithful latent states are optimal, we explore whether RL can induce more abstract and utility-driven internal visual reasoning.

4 Experiments

We evaluate LantErn through a two-stage training pipeline. First, we perform supervised fine-tuning (SFT) to initialize latent visual reasoning by grounding latent states in perceptual features. Second, we apply reinforcement learning (RL) to evaluate whether outcome-driven optimization improves the utility of the latent representations.

4.1 Supervised Fine-Tuning Setup

Dataset Construction. To train the model to associate latent states with visually relevant regions, we construct a synthetic dataset derived from Visual-CoT (Shao et al., 2024a). Visual-CoT provides image–question pairs accompanied by detailed reasoning traces and a bounding box highlighting the region of the image the model should focus on in order to answer the question effectively, making it a suitable foundation for structured latent supervision.

We employ a large-scale reasoning-oriented multimodal model, **Qwen3-VL-235B-Thinking** (Yang et al., 2025a), as the reference model. For each image–question pair, we prompt the model to generate a structured reasoning trace consisting of three components:

1. **Pre-visual thought:** A textual plan describing the visual information required (e.g., *"I need to identify the game title shown on the screen"*).
2. **Visual grounding:** A set of bounding boxes highlighting regions of interest (ROIs) that are relevant for answering the question (matching the ground-truth boxes).
3. **Post-visual thought:** A textual deduction derived specifically from the visual content within those ROIs.

This procedure yields training samples in which bounding boxes are explicitly aligned with individual reasoning steps in the supervision signal. During SFT, these bounding boxes are used exclusively to extract the corresponding target feature representations from the vision encoder, which are then used as regression targets for the model's latent blocks. Importantly, the bounding boxes themselves are never exposed to the model during training.

Training Configuration. We initialize all models from Qwen2.5-VL-3B-Instruct (Bai et al., 2025b). To study the effect of latent capacity, we train variants with different latent block sizes, each dubbed LantErn-SFT-$\{K\}$, where $K \in \{4, 8, 16, 32\}$. We use a weighting factor $\gamma = 0.1$ for the latent alignment loss. These ablations allow us to examine how the dimensionality of latent tokens affect downstream reasoning behavior.

Baselines. To isolate the contribution of continuous latent reasoning, we also train a version with only next-token prediction, namely LantErn-NTP. This baseline uses the same backbone architecture, same data and special control tokens (e.g., `<|lvr_start|>`, `<|lvr_end|>`) but treats the intermediate reasoning sequence as standard discrete text tokens. In particular, the language modeling head is never bypassed and no latent regression loss is applied. This comparison controls for additional computation steps and token structure, ensuring that any observed differences can be attributed specifically to the presence of continuous latent representations. We also include Qwen2.5-VL-3B as the base pretrained model without task-specific fine-tuning.

Hyperparameters. All models are trained using AdamW with a learning rate of 1×10^{-5} and a cosine learning rate schedule, together with a warmup ratio of 0.05. Additionally, the vision encoder is frozen to both simplify training and improve training stability, allowing the model to focus on learning effective latent visual reasoning on top of fixed visual features.

4.2 Evaluation Benchmarks

To evaluate LantErn's performance, we used a subset of Visual-CoT and two vision-centric benchmarks $V^\star$ Wu & Xie (2023) and a subset of Blink Fu et al. (2024). $V^\star$ assesses a model's ability to perform visual search in real-world scenarios, a fundamental capability of

Model	VisCoT	$V^\star$	$V^\star_{\text{DA}}$	$V^\star_{\text{RP}}$	Blink	Blink$_{\text{OL}}$	Blink$_{\text{RP}}$
Qwen2.5-VL-3B	0.66	0.70	0.75	0.63	0.65	0.48	0.81
LantErn-NTP-4	0.80	0.72	0.71	0.72	0.60	0.45	0.72
LantErn-SFT-4	0.80	0.62	0.68	0.57	0.61	0.51	0.72
LantErn-SFT-8	0.81	0.65	0.71	0.60	0.60	0.52	0.68
LantErn-SFT-16	0.80	0.60	0.65	0.55	0.54	0.53	0.55
LantErn-SFT-32	0.79	0.72	0.72	0.71	0.58	0.49	0.66

Table 1: Evaluation results on $V^\star$, Blink and a subset of VisCoT datasets. DA = Direct Attribution, RP = Relative Position, OL = Object Localization.

human cognitive reasoning process involving visual information. Blink, on the other hand, evaluates core visual perception skills in scenarios where solving the task using text alone (textual priors) is extremely challenging. We used a subset of Blink that focus on object localization and direct attribution, as it closely aligns with the skills learned in the previous stage 4.1. Together, these benchmarks provide a robust basis for measuring the advantage of using latent visual representations.

4.3 SFT Results

As shown in Table 1, all LantErn variants improve over the Qwen2.5-VL-3B baseline on Visual-CoT, indicating generalization beyond the supervision data. However, these gains are comparable to the text-only LantErn-NTP baseline (0.80), suggesting that supervised latent grounding alone yields limited task-level benefits. SFT mainly improves perception-centric skills: for example, LantErn-SFT-8 increases Blink$_{\text{OL}}$ performance from 0.45 to 0.52, indicating stronger object localization and perceptual grounding. In contrast, performance on relational subsets ($V^\star_{\text{RP}}$ and Blink$_{\text{RP}}$) remains similar or worse than LantErn-NTP, suggesting that latent representations are not yet reliably used for complex reasoning.

Another observation is that performance does not increase monotonically with the latent size K. For example, larger latent blocks can lead to degradation on some benchmarks (e.g., Blink$_{\text{RP}}$ drops from 0.72 at K=4 to 0.66 at K=32), indicating a trade-off between latent capacity and effective reasoning. This highlights a limitation of fixed-size latent reasoning and suggests that future work could benefit from mechanisms that adapt latent capacity to match the task complexity.

Overall, the SFT results suggest that supervised latent grounding provides perceptual structure but is insufficient on its own to produce consistent task-level improvements, motivating the subsequent reinforcement learning stage.

4.4 Reinforcement Learning Setup

Following SFT, we apply RL on the LantErn-SFT-8 model to refine the latent reasoning policy and to test whether outcome-based optimization improves the usefulness of latent states for visual reasoning. At the same time we keep the same baselines as before.

Dataset. For the RL stage, we use the VIRL-39k dataset (Wang et al., 2025), which contains a diverse collection of visual reasoning problems without explicit region-level supervision. This setting allows us to evaluate whether RL can guide latent reasoning in the absence of bounding box annotations, relying only on task-level feedback.

Implementation Details. We implement the RL training loop using the TRL library (von Werra et al., 2023), extending the standard `GRPOTrainer` to support *latent state replay*. As described in Section 3, latent replay records the continuous hidden states generated during the rollout phase and reinjects them during the policy update step. This modification stabilizes training by ensuring that importance sampling ratios are computed under a fixed latent trajectory, while still allowing gradients to propagate to the parameters responsible for latent generation.

Model	Viscot	$V^\star$	$V^\star_{DA}$	$V^\star_{RP}$	Blink	Blink$_{OL}$	Blink$_{RP}$
Qwen2.5-VL-3B	0.66	0.70	0.75	0.63	0.65	0.48	**0.81**
NTP-RL	0.82	0.66	0.75	0.57	0.64	0.47	0.80
LantErn-RL-8	**0.83**	**0.71**	**0.76**	**0.67**	**0.68**	**0.54**	**0.81**

Table 2: Evaluation results on VStar, Blink and a subset of Viscot datasets. DA = Direct Attribution, RP = Relative Position, OL = Object Localization.

Hyperparameters. We train with a learning rate of 5×10^{-6} and a warmup ratio of 0.03 and latent size $k = 8$, as using 8 latent tokens seems to yield the best overall performance, as indicated in Section 4.3). We set the KL regularization coefficient to $\beta = 0.1$ to limit policy drift from the SFT initialization.

During the rollout phase, we sample $G = 4$ completions per prompt using temperature $T = 0.6$ and top-$p = 0.85$ to encourage exploration of diverse latent reasoning trajectories. The reward function combines a sparse accuracy reward (weight 1.0) with a format reward (weight 1.0), which encourages the explicit use of latent reasoning blocks and prevents collapse to purely textual solutions.

4.5 RL Results

Applying RL on top of LantErn-SFT-8 leads to consistent performance improvements, outperforming both the base model and the NTP-RL variant, across all evaluated benchmarks. The largest gains appear on **out-of-distribution, perception-heavy** benchmarks. Notably, performance on Blink$_{\mathrm{RP}}$ improves from 0.68 (SFT) to 0.81, representing a substantial gain. This trend is consistent across additional tasks: compared to the NTP baseline, performance increases on $V^\star$RP (0.57 → 0.67) and Blink$_{\mathrm{OL}}$ (0.47 → 0.54), further indicating improved spatial and relational reasoning.

These results support our hypothesis that RL is the stage at which latent states transition from preceptually faithful reconstruction to task-driven internal visual representations. Although additional ablations are needed to fully characterize this effect, the consistent gains across benchmarks indicate that RL enables more effective internal use of visual information. Finally, achieving parity with a 7B model on several benchmarks highlights the potential of latent visual reasoning as a compute-efficient alternative to model scaling for perception-centric tasks.

5 Conclusions

In this paper, we present LantErn as a novel multimodal hybrid reasoning framework that interleaves latent visual reasoning with standard text generation. Our framework is trained in two stages. First, we perform SFT to distill this capability into the model by explicitly joint supervision in the latent representations and text tokens, enabling it to align these latent representations with visual concepts and to form abstract visual thoughts. In the second stage, we apply RL to further train the model to generate its own latent representations without being constrained to maintain strict fidelity enforced by the previous stage. This grants the model greater freedom to explore task-specific solutions and to revamp abstract visual thoughts, resulting in improved performance on visual reasoning benchmarks.

Limitations and Future Work: Despite its effectiveness, the framework has some limitations. First, interleaved latent reasoning depends on the quality and diversity of multimodal trajectories, which are currently concentrated in a narrow visual domain. Second, the model uses a fixed number of latent tokens; enabling dynamically sized latent blocks that adapt to task complexity is an promising direction. Finally, a deeper analysis of latent dependencies is needed, including methods to visualize latent representations and better understand their alignment with generated text.

6 Ethics Statement

LantErn is a multimodal reasoning framework designed to improve visual reasoning capabilities in large vision–language models. While such systems can enable beneficial applications in accessibility, education, and scientific analysis, they also raise ethical considerations related to misuse, bias, and transparency. Multimodal models may inherit biases present in their training data, including cultural, demographic, or representational imbalances. These biases can affect model outputs and may disproportionately impact underrepresented groups. Although Lantern focuses on reasoning mechanisms rather than dataset expansion, it relies on existing multimodal corpora whose limitations may propagate into the model. Care should be taken when deploying such systems in high-stakes settings.

7 Reproducibility Statement

We prioritize reproducibility by providing detailed descriptions of the Lantern architecture, training pipeline, and evaluation protocols. The paper specifies the model backbone, latent reasoning mechanism, and the two-stage training procedure (supervised fine-tuning and reinforcement learning). Hyperparameters, optimization settings, and dataset compositions are reported in the main text and appendix. We use publicly available benchmarks for evaluation and clearly describe preprocessing and evaluation procedures. All experiments are conducted using deterministic training configurations where possible, including fixed random seeds and documented hardware setups. To facilitate replication, we plan to release implementation details, training scripts, and configuration files upon publication. These materials will include instructions for reproducing the main experiments, along with pretrained checkpoints where licensing permits. We acknowledge that training large multimodal models requires significant computational resources. To mitigate this barrier, we provide ablation studies and smaller-scale configurations that reproduce key findings using reduced compute budgets.

References

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

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, 2025a. URL https://arxiv.org/abs/2502.13923.

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, 2025b. URL https://arxiv.org/abs/2502.13923.

Zixu Cheng, Jian Hu, Ziquan Liu, Chenyang Si, Wei Li, and Shaogang Gong. V-star: Benchmarking video-llms on video spatio-temporal reasoning, 2025. URL https://arxiv.org/abs/2503.11495.

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.

Daya Guo, Dejian Yang, Haotian Zhang, et al. Deepseek-r1: Reinforcement learning for reasoning in large language models. *Nature*, 2025. In press.

Bangzheng Li, Ximeng Sun, Jiang Liu, Ze Wang, Jialian Wu, Xiaodong Yu, Hao Chen, Emad Barsoum, Muhao Chen, and Zicheng Liu. Latent visual reasoning, 2025. URL https://arxiv.org/abs/2509.24251.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2304.08485.

Team Olmo, :, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, Jacob Morrison, Jake Poznanski, Kyle Lo, Luca Soldaini, Matt Jordan, Mayee Chen, Michael Noukhovitch, Nathan Lambert, Pete Walsh, Pradeep Dasigi, Robert Berry, Saumya Malik, Saurabh Shah, Scott Geng, Shane Arora, Shashank Gupta, Taira Anderson, Teng Xiao, Tyler Murray, Tyler Romero, Victoria Graf, Akari Asai, Akshita Bhagia, Alexander Wettig, Alisa Liu, Aman Rangapur, Chloe Anastasiades, Costa Huang, Dustin Schwenk, Harsh Trivedi, Ian Magnusson, Jaron Lochner, Jiacheng Liu, Lester James V. Miranda, Maarten Sap, Malia Morgan, Michael Schmitz, Michal Guerquin, Michael Wilson, Regan Huff, Ronan Le Bras, Rui Xin, Rulin Shao, Sam Skjonsberg, Shannon Zejiang Shen, Shuyue Stella Li, Tucker Wilde, Valentina Pyatkin, Will Merrill, Yapei Chang, Yuling Gu, Zhiyuan Zeng, Ashish Sabharwal, Luke Zettlemoyer, Pang Wei Koh, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. Olmo 3, 2025. URL https://arxiv.org/abs/2512.13961.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744. Curran Associates, Inc., 2022.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. Visual cot: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024a. URL https://openreview.net/forum?id=aXeiCbMFFJ.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024b. URL https://arxiv.org/abs/2402.03300.

D. Surís, S. Menon, and C. Vondrick. Vipergpt: Visual inference via python execution for reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models, 2025. URL https://arxiv.org/abs/2405.09818.

Leandro von Werra, Younes Belkada, Victor Sanh, et al. Trl: Transformer reinforcement learning. https://github.com/huggingface/trl, 2023.

Haozhe Wang, Chao Qu, Zuming Huang, Wei Chu, Fangzhen Lin, and Wenhu Chen. Vl-rethinker: Incentivizing self-reflection of vision-language models with reinforcement learning. *arXiv preprint arXiv:2504.08837*, 2025.

Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal llms, 2023. URL https://arxiv.org/abs/2312.14135.

Yijia Xiao, Edward Sun, Tianyu Liu, and Wei Wang. Logicvista: Multimodal llm logical reasoning benchmark in visual contexts, 2024. URL https://arxiv.org/abs/2407.04973.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025a. URL https://arxiv.org/abs/2505.09388.

Zeyuan Yang, Xueyang Yu, Delin Chen, Maohao Shen, and Chuang Gan. Machine mental imagery: Empower multimodal reasoning with latent visual tokens, 2025b. URL https://arxiv.org/abs/2506.17218.

Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. Mm-react: Prompting chatgpt for multimodal reasoning and action, 2023. URL https://arxiv.org/abs/2303.11381.

Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models, 2024. URL https://arxiv.org/abs/2302.00923.