TRAVIS: LEARNING LATENT VISUAL TRAJECTORIES WITH CONTEXT-ADAPTIVE VISUAL TARGETING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal large language models (MLLMs) often rely on localized visual evidence, yet existing supervision provides limited guidance on what intermediate latent states should visually predict. We introduce TraViS, a framework for learning latent visual trajectories with context-adaptive visual targets. Instead of using predefined intermediate supervision, TraViS enables each latent state to identify task-relevant visual evidence from the evolving multimodal context. Specifically, a frozen visual teacher encodes the selected region into a continuous visual target, which is aligned with the corresponding latent state and fed back to guide subsequent latent computation during training. At inference, the crop teacher and per-step target construction are gated to training only. TraViS therefore reverts to self-feedback latent rollout over a single global image encoding, so the target-construction machinery used during training adds no inference-time cost. Across multiple backbones and eight multimodal reasoning benchmarks, TraViS delivers over 2.5% average improvement while reducing inference latency.

1 INTRODUCTION

Multimodal large language models (MLLMs) have achieved strong performance on general image understanding, yet many image based questions are determined by a small amount of task relevant visual evidence, such as fine text, object attributes, or relations between local regions (Bai et al., 2025b; Huang et al., 2026; An et al., 2026). Existing reasoning methods increase computation through textual chain of thought or continuous latent states (Wei et al., 2022; Hao et al., 2024). However, additional computation alone does not specify what visual information intermediate states should learn to predict. Once the image has been encoded, their intermediate visual semantics are often only weakly constrained by the final response supervision.

This limitation is particularly important for latent visual reasoning. Existing approaches supervise intermediate states using language, visual embeddings, or synthesized visual concepts (Hao et al., 2024; Li et al., 2026; Wang et al., 2026a), but the supervision source is typically specified in advance. The evolving latent trajectory therefore has limited ability to determine which visual evidence should become relevant as its context changes. This is restrictive for image based reasoning, where different questions call for different regions at different stages of computation: a fine-grained attribute question needs early grounding on a small patch, while a global layout question needs late grounding on the whole frame. Figure 1 summarizes the paper.

How should a latent visual trajectory determine context adaptive visual targets?

We address this question with **TraViS**, a framework for learning latent visual trajectories with context adaptive visual targets. At each latent step, the evolving multimodal context determines a relevant visual region, which a frozen visual teacher encodes into a continuous target. We align the current latent state with this target and feed the target into the following latent slot during training, allowing each state to acquire task relevant visual evidence while shaping subsequent computation. When a ground truth region is available, it anchors only the terminal target, while intermediate targets remain determined by the evolving trajectory. Through this alignment, the learned latent state can approximate its visual target and replace explicit target construction with self-feedback at inference. The teacher and target construction are therefore used only during training, while inference

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

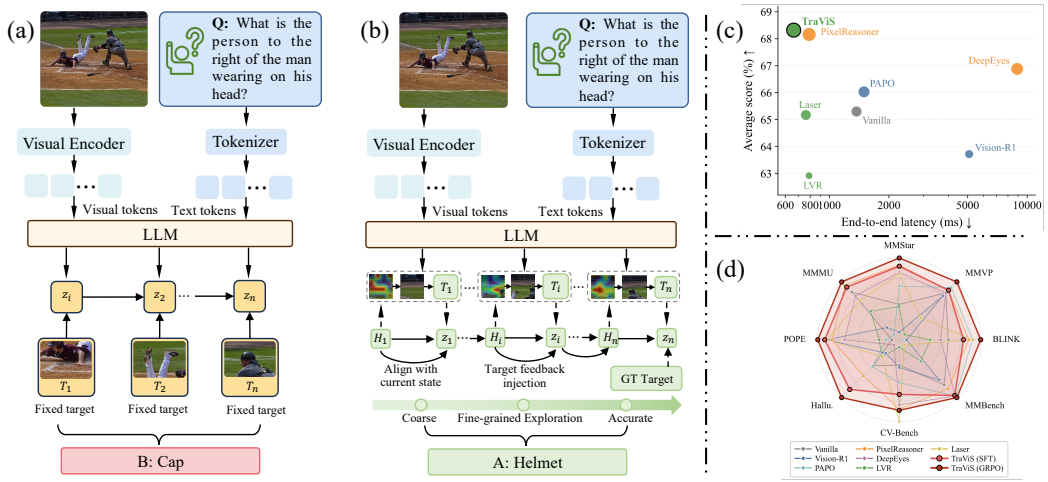

Figure 1: Overview of TraViS. (a) Existing latent reasoning relies on prescribed visual targets with limited visual refinement. (b) TraViS learns a latent visual trajectory through adaptive target feedback, with the latent state aligned to its target at every step and an optional ground-truth anchor at the final step. (c) Accuracy–latency comparison with representative reasoning methods. (d) Performance comparison across eight multimodal benchmarks.

requires only a single global image encoding and standard decoding, with no extra target construction cost.

Context adaptive targeting is central to the effectiveness of TraViS. Replacing it with a fixed crop while keeping the remaining framework unchanged reduces the mean score from 72.05 to 68.16 across six benchmarks, and the adaptive variant performs better on every benchmark. This result shows that the benefit comes not only from introducing visual supervision, but also from allowing the latent trajectory to adapt its visual targets as the multimodal context evolves.

Across three Qwen backbones (Bai et al., 2025a) and LLaVA-OV-1.5 (An et al., 2025), with eight multimodal benchmarks, TraViS outperforms its baseline across every backbone, with average gains exceeding 2.5%.

This work makes the following contributions.

- We formulate latent visual trajectory learning as a target-guided supervision problem, where each intermediate latent state selects its own visual target from the evolving multimodal context.
- We propose **TraViS**, which combines context-adaptive visual targeting with target feedback to align each latent state to task-relevant evidence and shape subsequent computation.
- **TraViS** delivers consistent gains across three Qwen backbones, LLaVA-OV-4b, and eight multimodal benchmarks, while achieving faster inference.

2 Related Work

Multimodal Large Language Models. MLLMs align visual features with language-model representations through large-scale image–text pretraining and instruction tuning (Liu et al., 2023; Li et al., 2024; McKinzie et al., 2024; Bai et al., 2025b; Qiao et al., 2025; 2026; Li, 2022; Zhang et al., 2025; Shao & Li, 2025). Recent works improve multimodal reasoning through different forms of supervision, including explicit reasoning traces (Wei et al., 2022; Zhang et al., 2023; Shao et al., 2024; Xu et al., 2024; Song et al., 2025), response-level reinforcement learning (Huang et al., 2026; Meng et al., 2025; Liu et al., 2025; Shen et al., 2025; Qiu et al., 2026), and visual-centric objectives based on activation or perceptual feedback (Wang et al., 2026b; Min et al., 2026; Hu et al., 2026). However, these approaches mainly supervise generated responses or predefined visual signals, while

the continuous latent states before response generation remain largely unexplored. TraViS addresses this gap by learning context-adaptive visual targets to guide latent visual trajectories.

Vision-Language Model Reasoning. Recent vision-language reasoning methods improve visual understanding by introducing explicit visual evidence manipulation, including region-grounded chains, sketchpads, crops, and zooms (Wu & Xie, 2024; Shao et al., 2024; Hu et al., 2024; Zheng et al., 2026; Su et al., 2025). Other approaches further explore executable programs, adaptive inspection, and region search to dynamically access informative visual content (Song et al., 2026; Liu et al., 2026b; Zhang et al., 2026). Additional studies leverage generated descriptions, perception-oriented objectives, or auxiliary observers to enhance visual reasoning (Xia et al., 2025; Xiao et al., 2025; Liu et al., 2026a; Jia et al., 2026). However, these approaches mainly rely on external visual operations or additional reasoning signals during inference. TraViS instead learns context-adaptive visual targets during training and optionally anchors the terminal state with region annotations. Therefore, TraViS differs in both the supervised signals and the inference procedure, and performs latent visual rollout using only the initial global image encoding at inference.

Latent-Space Visual Reasoning. Recent works explore latent reasoning by replacing explicit reasoning traces with continuous intermediate representations and investigating different forms of latent supervision (Zelikman et al., 2024; Xu et al., 2025; Fu & Luo, 2026; Chang et al., 2026). Coconut learns recurrent hidden states by predicting future language tokens without explicit textual reasoning steps (Hao et al., 2024); LVR introduces ROI visual embeddings as latent inputs and reconstruction targets during SFT (Li et al., 2026); Laser aligns latent states with a dynamic distribution over future synthesized visual-concept tokens (Wang et al., 2026a). These methods demonstrate the potential of latent computation, but their intermediate supervision is determined by predefined language or visual targets. TraViS instead learns state-conditioned visual targets through adaptive crop selection, uses frozen crop features as training-time feedback, and optionally anchors the terminal state with region annotations. Therefore, TraViS differs in both the supervised object and target construction process: language tokens in Coconut, ROI features in LVR, future visual-token distributions in Laser, and context-adaptive visual targets with feedback in TraViS.

3 METHOD

TraViS formulates latent reasoning as a target-guided visual trajectory learning problem, where intermediate latent states are progressively shaped by context-adaptive visual targets derived from the evolving multimodal context. Unlike existing latent reasoning approaches that rely on predefined intermediate supervision, TraViS enables each latent state to construct its own visual target according to the current reasoning stage, allowing the latent trajectory to continuously capture task-relevant visual evidence. The full architecture is detailed in Figure 2.

3.1 LATENT VISUAL TRAJECTORY FORMULATION

Given an image I and a question q, TraViS introduces a sequence of continuous latent states before generating the final response. We augment the original multimodal sequence with latent slots:

$$[I, q, \langle look \rangle, \langle z \rangle_{1:n}, \langle /look \rangle, o], \tag{1}$$

where $\{z_i\}_{i=1}^{n}$ are continuous latent slots to provide state-dependent visual supervision for latent trajectory learning. Unlike fixed intermediate targets, the target in TraViS evolves with the current latent context and reflects the visual evidence required at different reasoning stages and o denotes the final response. These latent slots participate in Transformer self-attention but are excluded from the language modeling loss, providing intermediate states for latent reasoning.

Let $p_1, \ldots, p_n$ denote the latent slot positions. For the i-th step, we define

$$H_i = h_{p_i - 1}, \qquad z_i = e_{p_i}, \tag{2}$$

where H_i is the hidden state before the latent slot and z_i is its continuous input. Since H_i aggregates the image, question, and previous latent context, it serves as the state for constructing the visual target. The sequence of latent states $\{H_i\}_{i=1}^{n}$ forms a latent visual trajectory, where each step adapts to the multimodal context rather than relying on a fixed intermediate target space, such that target variations mainly come from the evolving latent context.

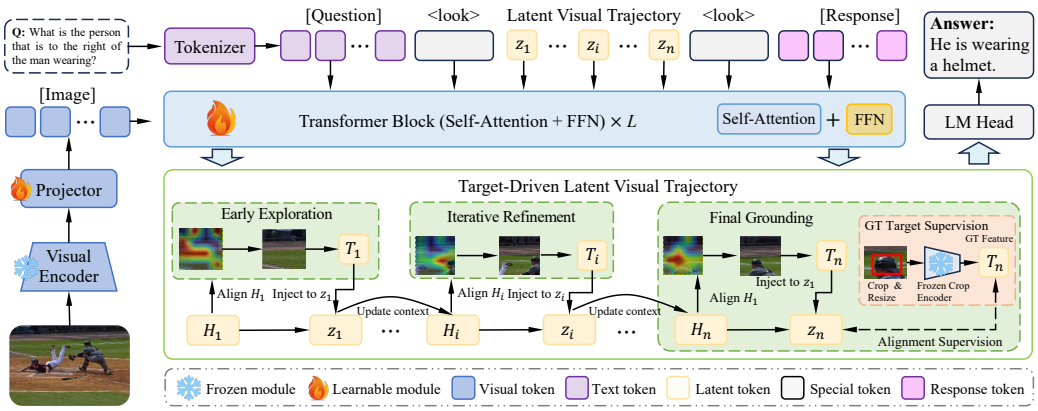

Figure 2: Overview of TraViS. Intermediate latent states form a visual trajectory, where context adaptive visual targets provide state dependent supervision and feedback to progressively guide the evolution of latent computation.

3.2 Context Adaptive Visual Targeting

TraViS constructs a context-adaptive visual target at each latent step to provide state-dependent visual supervision for latent trajectory learning. Unlike fixed intermediate targets, the target in TraViS evolves with the current latent context and reflects the visual evidence required at different reasoning stages. For $i < n$, the current state H_i selects a relevant local region B_i, which is encoded by a frozen visual teacher as:

$$T_i = \mathrm{sg}\left[\phi(B_i)\right], \tag{3}$$

where sg denotes stop-gradient and ϕ is the frozen target encoder. Thus, different latent states receive different visual supervision while remaining in a stable feature space.

Frozen visual target encoding. Let $\mathcal{E}_{\mathrm{T}}$ denote the frozen crop teacher, including the visual encoder and visual-language projector. Given a crop B, we resize and encode it with the teacher, followed by mean pooling of the projected visua tokens:

$$\phi(B) = \mathrm{mean}\left[\mathcal{E}_{\mathrm{T}}\left(\mathrm{resize}_{s\times s}(I|_B)\right)\right] \in \mathbb{R}^d. \tag{4}$$

The frozen teacher provides a stable target space, such that target variations mainly come from the evolving latent context.

Context-adaptive region selection. To construct a target matching the current reasoning stage, TraViS performs state-conditioned region selection from the global visual representation. At latent step i, H_i measures compatibility with visual tokens, with attention scores averaged across heads:

$$a_i = \frac{1}{N_h}\sum_{h=1}^{N_h}\frac{q_h(H_i)K_h^\top}{\sqrt{d_k}} \in \mathbb{R}^M, \qquad K_h = [k_h(v_1);\ldots;k_h(v_M)]. \tag{5}$$

We retain the highest scoring visual tokens:

$$\mathcal{S}_i = \mathrm{TopK}(a_i, k), \qquad k = \min\left\{M, \max\left(\left\lfloor\frac{M}{10}\right\rfloor, 4\right)\right\}. \tag{6}$$

The selected tokens are mapped back to the $H_g \times W_g$ merged patch grid, where $M = H_g W_g$. Token j corresponds to

$$r(j) = \left\lfloor\frac{j}{W_g}\right\rfloor, \qquad c(j) = j \bmod W_g. \tag{7}$$

We then search for the $w \times w$ window containing the largest number of retained tokens:

$$(r_i^\star, c_i^\star) = \underset{(r,c)\in\Omega_w}{\arg\max}\sum_{j\in\mathcal{S}_i}\mathbf{1}\left[(r(j), c(j)) \in \mathcal{W}(r,c)\right], \tag{8}$$

where

$$\Omega_w = \{0, \ldots, H_g - w\} \times \{0, \ldots, W_g - w\}. \tag{9}$$

For an input image of size $H_I \times W_I$, the selected grid window is mapped back to image coordinates:

$$\rho_h = \frac{H_I}{H_g}, \qquad \rho_w = \frac{W_I}{W_g}, \tag{10}$$

$$B_i = [c_i^\star \rho_w, (c_i^\star + w)\rho_w] \times [r_i^\star \rho_h, (r_i^\star + w)\rho_h], \tag{11}$$

with coordinates clipped to the image boundary.

The selection and crop encoding process is performed only for target construction and does not participate in gradient propagation. Since the crop size is fixed, different latent states adapt the supervision through the selected visual evidence rather than the target granularity.

Terminal target anchoring. When region annotations are available, TraViS uses them only as a terminal anchor rather than intermediate supervision:

$$B_R = \mathrm{bbox}(R), \qquad T_n = \mathrm{sg}\left[\phi(B_R)\right]. \tag{12}$$

All preceding targets remain context adaptive. Without region annotations, T_n is constructed using the same adaptive selection strategy. Region annotations are not required during inference.

3.3 Latent Trajectory Learning with Target Feedback

TraViS treats the visual targets as a representation alignment signal rather than a teacher-forcing dependency. The alignment loss pulls each latent state into the teacher's visual representation space, while target feedback propagates that aligned visual signal through subsequent latent states. Once trained, the latent state H_i approximates T_i closely enough to substitute for it at inference.

Visual state alignment. We align each pre-slot state H_i with its corresponding target T_i:

$$\mathcal{L}_{\mathrm{align}} = \frac{1}{n} \sum_{i=1}^{n} \|H_i - T_i\|_2^2 . \tag{13}$$

The loss is evaluated before the slot content arrives, so it encourages each latent state to anticipate the visual evidence needed by the next reasoning step.

Target feedback. The slot input differs between training and inference:

$$z_i = e_{p_i} = \begin{cases} \mathrm{sg}[T_i], & \text{training}, \\ H_i, & \text{inference}. \end{cases} \tag{14}$$

The substitution is justified by the alignment above: once trained, the two states lie in the same representation space, so feeding H_i back is approximately equivalent to feeding the teacher's signal forward. The heavy target construction stays at training time only; inference reverts to self-feedback over a single global image encoding, with self-consistency established by the conditional stabilization result in Section A.5.

Training objective. The objective combines response generation with latent visual alignment:

$$\mathcal{L} = \mathcal{L}_{\mathrm{CE}} + \lambda \mathcal{L}_{\mathrm{align}}, \tag{15}$$

where $\mathcal{L}_{\mathrm{CE}}$ is applied only to the visible response o. Unless otherwise specified, we use $n = 8$, $w = 4$, $s = 280$, and $\lambda = 0.1$. The language model is optimized jointly by the two losses, while the crop teacher remains frozen. At inference, TraViS performs finite latent rollout using a single global image encoding and then generates the response autoregressively. Additional analysis of target stability is provided in Appendix A.4.

Table 1: Results across Qwen and LLaVA backbones, where LLaVA-OV-4b denotes LLaVA-OneVision-1.5-4B. The Avg. column reports the mean across all eight benchmarks. **Bold** values mark the best score within each backbone pair. Each Δ row reports the per-backbone gain of **TraViS** over its own baseline.

Base Model	MMStar	MMVP	BLINK	MMBench	CV-Bench	Hallu.	POPE	MMMU	*Avg.*
Qwen2.5-VL-7B	60.47	56.00	52.55	80.15	74.22	65.19	82.52	51.33	65.30
+ TraViS	**65.07**	**56.67**	**54.55**	**83.51**	**78.02**	**68.05**	**88.48**	**52.22**	**68.32**
Δ	**+4.60**	**+0.67**	**+2.00**	**+3.36**	**+3.80**	**+2.86**	**+5.96**	**+0.89**	**+3.02**
Qwen3-VL-4B	61.40	51.33	61.86	83.62	83.53	64.77	89.09	52.11	68.46
+ TraViS	**63.13**	**64.67**	**63.07**	**85.40**	**85.89**	**70.12**	**89.65**	**52.33**	**71.78**
Δ	**+1.73**	**+13.34**	**+1.21**	**+1.78**	**+2.36**	**+5.35**	**+0.56**	**+0.22**	**+3.32**
Qwen3-VL-8B	62.13	70.67	64.23	86.51	84.63	64.34	89.30	54.11	71.99
+ TraViS	**65.93**	**72.67**	**65.49**	**87.11**	**87.58**	**72.08**	**89.62**	**55.78**	**74.53**
Δ	**+3.80**	**+2.00**	**+1.26**	**+0.60**	**+2.95**	**+7.74**	**+0.32**	**+1.67**	**+2.54**
LLaVA-OV-4B	64.43	57.33	50.87	85.22	67.66	60.31	**90.26**	52.44	66.07
+ TraViS	**64.60**	**58.67**	**53.49**	**86.51**	**76.87**	**64.67**	89.34	**54.22**	**68.55**
Δ	**+0.17**	**+1.34**	**+2.62**	**+1.29**	**+9.21**	**+4.36**	-0.92	**+1.78**	**+2.48**

4 EXPERIMENTS

We evaluate TraViS on multiple multimodal benchmarks across different backbones, and further analyze its efficiency and key design choices.

4.1 EXPERIMENTAL SETUP

Unless stated otherwise, we train TraViS on the Visual-CoT 363k dataset (Shao et al., 2024) for one epoch with an effective batch size of 16. TraViS uses eight latent look steps, full language model fine-tuning with the visual encoder and projector frozen, no output head, and a latent alignment loss weight of $\lambda = 0.1$. The visual teacher encodes crops with a window size of $w = 4$ on the merged image token grid at a target resolution of $s = 280$. All controlled ablations share the same backbone and training data. We evaluate TraViS on eight public multimodal benchmarks, including MMStar (Chen et al., 2024), MMVP (Tong et al., 2024b), BLINK (Fu et al., 2024), MMBench (Liu et al., 2024), CV-Bench (Tong et al., 2024a), HallusionBench (Guan et al., 2024), POPE (Li et al., 2023), and MMMU (Yue et al., 2024), following each benchmark's official protocol. Detailed training hyperparameters are in Appendix A.6.

4.2 MAIN RESULTS

Main comparison. As shown in Table 1, TraViS consistently improves the overall performance across all four backbones, with average gains ranging from 2.48% to 3.32%. More importantly, the improvements are not evenly distributed across benchmarks, and the largest gains appear on tasks that require fine-grained visual discrimination, spatial relation understanding, and localized evidence selection. Specifically, TraViS improves MMVP by 13.34% on Qwen3-VL-4B, HallusionBench by 7.74% on Qwen3-VL-8B, and CV-Bench by 9.21% on LLaVA-OV-4B. By comparison, the gains on benchmarks that rely more heavily on global semantics or answer-level recognition are generally smaller. This pattern is consistent with the motivation of TraViS, since the method explicitly guides intermediate latent states to capture task-relevant visual evidence as the reasoning context evolves rather than simply increasing the amount of latent computation. Similar improvements across both Qwen and LLaVA backbones further indicate that the benefit mainly comes from the proposed latent trajectory learning mechanism rather than a specific model architecture.

Comparison with multimodal reasoning methods. Table 2 compares TraViS with representative textual CoT, Think with Image, and latent reasoning methods on Qwen2.5-VL-7B. TraViS with SFT improves the vanilla baseline by 3.02% on average and remains competitive across different reasoning paradigms, showing that latent visual trajectory learning can provide effective reasoning gains without relying on explicit textual reasoning chains or repeated visual operations. Compared with LVR and Laser, the results further indicate that introducing latent computation alone is not sufficient, since the performance of latent reasoning also depends on how intermediate states are

Table 2: Additional comparison with representative multimodal reasoning and perception systems. The **best** and second best values are emphasized.

Method	MMStar	MMVP	BLINK	MMBench	CV-Bench	Hallu.	POPE	MMMU	*Avg.*
Vanilla	60.47	56.00	52.55	80.15	74.22	65.19	82.52	51.33	65.30
Vision-R1	57.10	55.30	47.08	78.61	74.54	63.69	88.33	45.11	63.72
PAPO	62.67	57.33	55.76	83.76	76.48	65.32	83.04	43.89	66.03
PixelReasoner	65.20	56.67	55.23	81.44	79.93	66.41	87.83	52.55	68.16
DeepEyes	64.33	56.00	51.55	83.68	79.37	62.70	86.72	50.67	66.88
LVR	60.20	46.00	50.18	72.79	71.90	64.12	83.65	48.00	62.92
Laser	64.13	50.00	55.81	67.61	**81.53**	64.12	88.27	49.89	65.17
TraViS (SFT)	65.07	56.67	54.55	83.51	78.02	68.05	88.48	52.22	68.32
TraViS (GRPO)	**66.07**	**58.67**	**56.81**	**84.19**	80.08	**69.03**	**89.10**	**53.13**	**69.64**

guided during the reasoning process. TraViS addresses this issue through context-adaptive visual targets and target feedback, allowing task-relevant visual evidence to influence subsequent latent computation as the trajectory evolves. GRPO further improves the average performance by 1.32% over TraViS with SFT and achieves the best result on six of eight benchmarks, which suggests that reinforcement post training provides a complementary benefit by optimizing task-level behavior on top of the learned latent visual trajectory.

4.3 Efficiency Analysis

Different multimodal reasoning paradigms introduce substantially different inference costs. Methods based on explicit textual reasoning or repeated visual interaction generally incur higher latency, whereas latent reasoning provides a more compact alternative. TraViS further reduces this cost and achieves the lowest end-to-end latency of 660.8 ms among the compared methods, while keeping peak memory close to the vanilla model. This efficiency follows from the inference design. The crop teacher and the per-step ViT encodes used to construct intermediate targets are gated to training only, so the inference pass is a single global image encoding followed by an n-step latent rollout and standard autoregressive decoding. The image is encoded once, the target-construction machinery adds no inference-time cost, and the response is correspondingly short since the model no longer needs to verbalize the visual evidence. The results in Table 3 therefore show that TraViS improves reasoning performance without introducing substantial additional inference overhead. Detailed response length and decode latency statistics are provided in Appendix A.2.

Table 3: Efficiency comparison of different multimodal reasoning paradigms on Qwen2.5-VL-7B.

Method	E2E (ms)	Peak (GB)
Baseline		
Base	1372.4	15.77
CoT Reasoning		
Vision-R1	5089.4	15.78
PAPO	1497.6	15.78
Think with Image		
PixelReasoner	790.7	15.91
DeepEyes	8901.8	16.15
Latent Reasoning		
LVR	790.0	15.99
Laser	761.3	16.00
Ours	**660.8**	**15.80**

4.4 Ablation Studies

We analyze the effects of latent trajectory length, learning rate, parameter adaptation, context adaptive targeting, and output heads under controlled settings.

Latent look steps. The number of latent look steps has a clear but non-monotonic effect on performance. A moderate trajectory length performs best across benchmarks, while overly short trajectories limit latent visual evolution and longer trajectories bring little additional benefit. We therefore use eight steps as the default setting. The trajectory is self-reinforcing under the alignment objective: each step trains H_i to approximate its target T_i, the detached $\mathrm{sg}[T_i]$ then conditions the next slot, and the next H_{i+1} is computed from an already-aligned context. The alignment between H_i and T_i therefore tightens as the rollout advances, so early-step errors do not propagate unchecked.

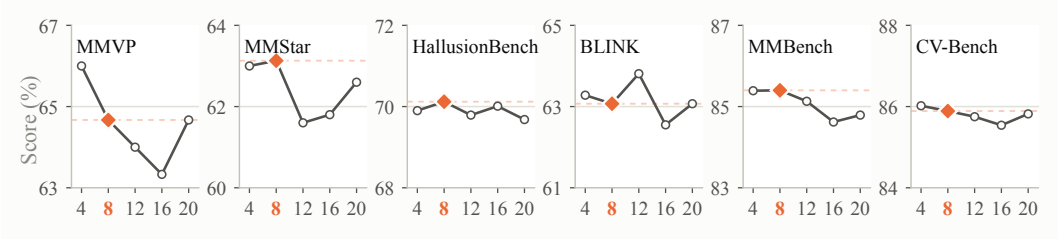

Figure 3: Ablation of latent look steps across six benchmarks. Eight steps are selected as the default configuration based on the overall performance.



Figure 4: Visualization of latent visual trajectories across eight look steps. Each row shows progressive refinement of task-relevant visual evidence. Heatmaps indicate state-conditioned attention, yellow dashed boxes denote selected regions, and red boxes denote reference regions.

Figure 4 shows this in practice: the selected crop targets vary across latent steps despite a fixed crop size, and the rollout progressively refines its visual focus. The pattern suggests that the gain of TraViS comes not from simply increasing latent computation, but from allowing intermediate states to progressively capture task-relevant visual evidence.

Latent SFT learning rate. Latent SFT shows clear sensitivity to the learning rate. The average is essentially flat between 5×10^{-7} and 1×10^{-6}, reaches 72.05% at 1×10^{-6}, and then drops monotonically past that point (Table 7). The degradation concentrates on the fine-grained benchmarks: MMVP, BLINK, and CV-Bench each lose ten to twelve points between 1×10^{-6} and 8×10^{-6}, while MMBench and MMStar decline more gradually. The mechanism is that target construction depends on the running model, so at high LR the latent targets shift faster than the alignment loss can follow and the representation that H_i is supposed to align with becomes unstable. Conservative optimization keeps the target distribution nearly stationary across consecutive updates (Section A.4) and gives the alignment loss a stable anchor. We therefore use 1×10^{-6} as the default learning rate.

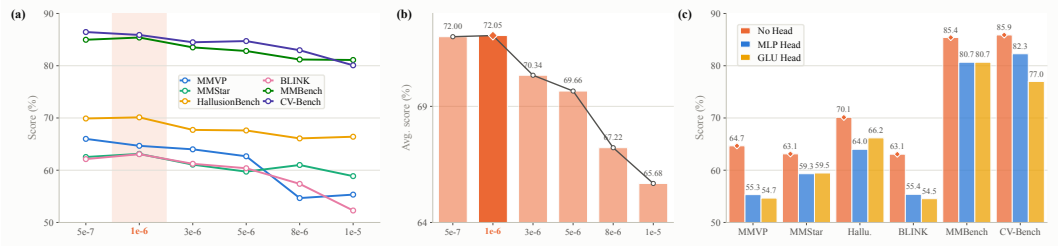

Figure 5: Learning-rate and output-head ablations on six benchmarks. (a) and (b) show benchmark and mean scores across learning rates, while (c) compares the three output-head settings.

Fine-tuning parameterization. Full parameter fine-tuning yields better performance than LoRA across evaluated benchmarks. As shown in Table 4, it provides an average gain of 0.93%, indicating that latent trajectory learning benefits from coordinated adaptation of language model representations rather than updating limited parameters. This is relevant to TraViS, where intermediate latent states must align with adaptive visual targets throughout the trajectory.

Table 4: Comparison of full fine tuning and LoRA for latent trajectory training.

Setting	MMStar	BLINK	MMBench	CV-Bench
LoRA	61.60	62.28	84.88	85.00
Full	63.13	63.07	85.40	85.89
Δ	+1.53	+0.79	+0.52	+0.89

Effect of context-adaptive targeting. Context-adaptive targeting consistently outperforms fixed visual supervision across all six benchmarks. As shown in Table 5, it brings an average improvement of 3.89%, with particularly large gains on CV-Bench, MMVP, and BLINK, where performance improves by 5.82%, 5.34%, and 4.36%, respectively. These benchmarks rely heavily on fine-grained perception and localized evidence, which matches the motivation of TraViS. The results indicate that the benefit comes not simply from introducing intermediate visual supervision, but from allowing the supervision target to adapt to the evolving latent context. This confirms context-adaptive targeting as a key component of latent visual trajectory learning.

Table 5: Effect of context adaptive visual targeting compared with fixed visual supervision.

Setting	MMStar	MMVP	Hallu.	BLINK	MMBench	CV-Bench	Mean
Fixed	61.67	59.33	66.73	58.71	82.47	80.07	68.16
Ours	63.13	64.67	70.12	63.07	85.40	85.89	72.05
Δ	+1.46	+5.34	+3.39	+4.36	+2.93	+5.82	+3.89

Output head. The alignment loss already pulls the latent state into the target space, so a learned projection head is, in principle, redundant. We test this by comparing direct prediction against an MLP head and a GLU head. Table 8 shows that direct prediction wins on every benchmark. The two extra heads trail by five to ten points on average, with largest damage on MMVP and BLINK, and the MLP and GLU heads differ by 0.74, within seed variance. We adopt direct alignment.

5 Conclusion

We introduced **TraViS**, a framework for learning latent visual trajectories with context-adaptive visual targets. The key idea is representation alignment. Each latent state is trained to approximate its teacher-derived target, so the trained state can stand in for the target and the heavy target construction is left at training time only. This design delivers consistent gains of over 2.5% on average across three Qwen backbones, LLaVA-OV-4b, and eight multimodal benchmarks, while running faster than the vanilla model. The results suggest that effective latent visual reasoning depends not only on additional computation, but also on providing intermediate states with visual targets that adapt to the reasoning context.

AI USE STATEMENT

We used generative AI tools to assist with translation and grammar checking of the manuscript. All AI-assisted content was reviewed by the authors for accuracy and consistency with the intended meaning. We take full responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the model architecture, training configuration, datasets, evaluation protocols, and implementation details in the main paper and appendix. All experiments follow the official dataset splits and clearly specified evaluation metrics to facilitate consistent comparison. We will release the source code and pretrained models to further support reproducibility.

REFERENCES

Hongjun An, Wenhan Hu, Sida Huang, Siqi Huang, Ruanjun Li, Yuanzhi Liang, Jiawei Shao, Yiliang Song, Zihan Wang, Cheng Yuan, et al. Ai flow: Perspectives, scenarios, and approaches. *Vicinagearth*, 3(1):1, 2026.

Xiang An, Yin Xie, Kaicheng Yang, Wenkang Zhang, Xiuwei Zhao, Zheng Cheng, Yirui Wang, Songcen Xu, Changrui Chen, Didi Zhu, et al. Llava-onevision-1.5: Fully open framework for democratized multimodal training. *arXiv preprint arXiv:2509.23661*, 2025.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025a.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025b. URL https://arxiv.org/abs/2502.13923.

Shuochen Chang, Tong Bai, Xiaofeng Zhang, Qianli Ma, Qingyang Liu, Zhaohe Liao, Yibo Miao, and Li Niu. Unlocking the black box of latent reasoning: An interpretability-guided approach to intervention. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, pp. 34019–34032, 2026. doi: 10.18653/v1/2026.acl-long.1568.

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

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

Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A. Smith, Wei-Chiu Ma, and Ranjay Krishna. BLINK: Multimodal large language models can see but not perceive. In *European Conference on Computer Vision*, 2024.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, Dinesh Manocha, and Tianyi Zhou. Hallusionbench: An advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14375–14385, 2024.

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. *arXiv preprint arXiv:2412.06769*, 2024. URL https://arxiv.org/abs/2412.06769.

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. *arXiv preprint arXiv:2406.09403*, 2024. URL https://arxiv.org/abs/2406.09403.

Zhengdong Hu, Chao Wang, Fengyun Rao, Jing Lyu, Hehe Fan, and Yi Yang. PointThinker: Point-incentivized parallel thinking for multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26240–26250, 2026.

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

Hongrui Jia, Chaoya Jiang, Shikun Zhang, and Wei Ye. Mitigating visual context degradation in large multimodal models: A training-free decoupled agentic framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Findings*, pp. 9574–9585, 2026.

Bangzheng Li, Ximeng Sun, Jiang Liu, Ze Wang, Jialian Wu, Xiaodong Yu, Emad Barsoum, Muhao Chen, and Zicheng Liu. Latent visual reasoning. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=j84WR5ORsC.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-OneVision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024. URL https://arxiv.org/abs/2408.03326.

Xuelong Li. Positive-incentive noise. *IEEE Transactions on Neural Networks and Learning Systems*, 35(6):8708–8714, 2022.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 292–305, 2023.

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

Hongcheng Liu, Pingjie Wang, Yuhao Wang, Siqu Ou, Yanfeng Wang, and Yu Wang. When seeing is not enough: Revealing the limits of active reasoning in MLLMs. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, pp. 27395–27422, 2026a. doi: 10.18653/v1/2026.acl-long.1264.

Ruixun Liu, Bowen Fu, Jiayi Song, Kaiyu Li, Wanchen Li, Lanxuan Xue, Hui Qiao, Weizhan Zhang, Deyu Meng, and Xiangyong Cao. ZoomEarth: Active perception for ultra-high-resolution geospatial vision-language tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 34877–34888, 2026b.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. MMBench: Is your multi-modal model an all-around player? In *European Conference on Computer Vision*, pp. 216–233, 2024.

Ziyu Liu, Zeyi Sun, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Haodong Duan, Dahua Lin, and Jiaqi Wang. Visual-rft: Visual reinforcement fine-tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2034–2044, 2025. doi: 10.1109/ICCV51701.2025.00197.

Brandon McKinzie, Zhe Gan, Jean-Philippe Fauconnier, Sam Dodge, Bowen Zhang, Philipp Dufter, Dhruti Shah, Xianzhi Du, Futang Peng, Floris Weers, et al. MM1: Methods, analysis and insights from multimodal LLM pre-training. *arXiv preprint arXiv:2403.09611*, 2024. URL https://arxiv.org/abs/2403.09611.

Fanqing Meng, Lingxiao Du, Zongkai Liu, Zhixiang Zhou, Quanfeng Lu, Daocheng Fu, Botian Shi, Wenhai Wang, Junjun He, Kaipeng Zhang, et al. MM-Eureka: Exploring visual aha moment with rule-based large-scale reinforcement learning. *arXiv preprint arXiv:2503.07365*, 2025. URL https://arxiv.org/abs/2503.07365.

Yingqian Min, Kun Zhou, Yifan Li, Yuhuan Wu, Han Peng, Yifan Du, Wayne Xin Zhao, Min Yang, and Ji-Rong Wen. Improving vision-language models with perception-centric process reward models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 33099–33109, 2026.

Xiaozhen Qiao, Jingkai Zhao, Yuqiu Jiang, Xianda Guo, Zhe Sun, Hongyuan Zhang, and Xuelong Li. Class-aware prototype learning with negative contrast for test-time adaptation of vision-language models. *arXiv preprint arXiv:2510.19802*, 2025.

Xiaozhen Qiao, Peng Huang, Jiakang Yuan, Xianda Guo, Bowen Ye, Chaocan Xue, Ye Zheng, Zhe Sun, and Xuelong Li. Bidirectional prototype-reward co-evolution for test-time adaptation of vision-language models. *IEEE Transactions on Multimedia*, 2026.

Han Qiu, Sheng Jin, Zhongrong Zuo, Ziyue Wang, Qi She, Ling Shao, and Shijian Lu. Multimodal reasoning with explicit reasoning patterns and rewards. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Findings*, pp. 9542–9551, 2026.

Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. Visual CoT: Unleashing chain-of-thought reasoning in multi-modal language models. *arXiv preprint arXiv:2403.16999*, 2024. URL https://arxiv.org/abs/2403.16999.

Jiawei Shao and Xuelong Li. Ai flow at the network edge. *IEEE Network*, 40(1):330–336, 2025.

Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, Ruochen Xu, and Tiancheng Zhao. VLM-R1: A stable and generalizable r1-style large vision-language model. *arXiv preprint arXiv:2504.07615*, 2025. URL https://arxiv.org/abs/2504.07615.

Honghao Song, Liang Wang, Xiaozhen Qiao, Yifan Chen, Da Sun, and Zhe Sun. Embodied intelligence for robot manipulation: development and challenges. *Vicinagearth*, 2(1):8, 2025.

Qi Song, Honglin Li, Yingchen Yu, Haoyi Zhou, Lin Yang, Song Bai, Qi She, Zilong Huang, and Yunqing Zhao. CodeDance: A dynamic tool-integrated MLLM for executable visual reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19186–19195, 2026.

Alex Su, Haozhe Wang, Weiming Ren, Fangzhen Lin, and Wenhu Chen. Pixel reasoner: Incentivizing pixel-space reasoning with curiosity-driven reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, Austin Wang, Rob Fergus, Yann LeCun, and Saining Xie. Cambrian-1: A fully open, vision-centric exploration of multimodal LLMs. In *Advances in Neural Information Processing Systems*, 2024a.

Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal LLMs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9568–9578, 2024b.

Yubo Wang, Juntian Zhang, Yichen Wu, Yankai Lin, Nils Lukas, and Yuhan Liu. Forest before trees: Latent superposition for efficient visual reasoning. *arXiv preprint arXiv:2601.06803*, 2026a. URL https://arxiv.org/abs/2601.06803.

Zengbin Wang, Feng Xiong, Liang Lin, Xuecai Hu, Yong Wang, Yanlin Wang, Man Zhang, and Xiangxiang Chu. Visually-guided policy optimization for multimodal reasoning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, pp. 6646–6664, 2026b. doi: 10.18653/v1/2026.acl-long.301.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2022.

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

Jiaer Xia, Yuhang Zang, Peng Gao, Yixuan Li, and Kaiyang Zhou. Visionary-R1: Mitigating shortcuts in visual reasoning with reinforcement learning. *arXiv preprint arXiv:2505.14677*, 2025. URL https://arxiv.org/abs/2505.14677.

Tong Xiao, Xin Xu, Zhenya Huang, Hongyu Gao, Quan Liu, Qi Liu, and Enhong Chen. Advancing multimodal reasoning capabilities of multimodal large language models via visual perception reward. *arXiv preprint arXiv:2506.07218*, 2025. URL https://arxiv.org/abs/2506.07218.

Guowei Xu, Peng Jin, Hao Li, Yibing Song, Lichao Sun, and Li Yuan. LLaVA-CoT: Let vision language models reason step-by-step. *arXiv preprint arXiv:2411.10440*, 2024. URL https://arxiv.org/abs/2411.10440.

Yige Xu, Xu Guo, Zhiwei Zeng, and Chunyan Miao. SoftCoT: Soft chain-of-thought for efficient reasoning with LLMs. *arXiv preprint arXiv:2502.12134*, 2025. URL https://arxiv.org/abs/2502.12134.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert AGI. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9556–9567, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Yue_MMMU_A_Massive_Multi-discipline_Multimodal_Understanding_and_Reasoning_Benchmark_for_CVPR_2024_paper.html.

Eric Zelikman, Georges Harik, Yijia Shao, Varuna Jayasiri, Nick Haber, and Noah D. Goodman. Quiet-STaR: Language models can teach themselves to think before speaking. *arXiv preprint arXiv:2403.09629*, 2024. URL https://arxiv.org/abs/2403.09629.

Hongyuan Zhang, Sida Huang, Yubin Guo, and Xuelong Li. Variational positive-incentive noise: How noise benefits models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47 (9):8313–8320, 2025.

Jinrong Zhang, Zhaoyang Xu, Xusheng He, Xinrui Li, Na Zheng, and Jianlong Wu. Breaking the regional perception bottleneck of multimodal large language models via external reasoning framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 33531–33541, 2026.

Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*, 2023. URL https://arxiv.org/abs/2302.00923.

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. In *International Conference on Learning Representations*, 2026.

A Appendix

A.1 Algorithm Pseudocode

Algorithm 1 summarizes the TraViS procedure in a single loop that branches between training and inference. Most lines are shared; the only structural differences are whether ϕ is called and what fills the slot. Training feeds each z_i the detached teacher target $\mathrm{sg}[T_i]$, while inference feeds the model's own prediction H_i. A few details that the pseudocode leaves implicit are spelled out below.

The input in line 4 is $[I, q, \langle\text{look}\rangle, z_1, \ldots, z_n, \langle/\text{look}\rangle, \emptyset]$, where I is the global image, q the question, and z_i the i-th learnable latent slot bracketed by look tokens; the trailing $\emptyset$ marks where the response o will be generated. $V = \{v_1, \ldots, v_M\}$ is the global visual encoding obtained from I at line 3 and is reused across all n steps. The pre-slot state $H_i = h_{p_i - 1}$ already aggregates the image, the question, and all previous slots, which is what makes it a useful conditioning signal for the adaptive target.

In the training branch, when $i < n$ or no terminal region R is available, lines 8–11 build T_i in four sub-steps. First, attention scores $a_i = q(H_i)K^\top/\sqrt{d_k}$ are computed across the visual tokens. Second, a Top-K selection $\mathcal{S}_i = \mathrm{TopK}(a_i, k)$ is taken. Third, the highest-density $w \times w$ window in the merged grid is selected. Fourth, the image crop B_i is obtained by mapping the window back to image coordinates. The crop is then encoded by ϕ and accumulated into the alignment loss. The detached target $\mathrm{sg}[T_i]$ becomes the slot input, providing teacher feedback for the next step. When R is available at $i = n$, lines 12–14 replace the adaptive selection with the ground-truth region and anchor the last step.

Inference drops both ϕ and the alignment loss; line 15 simply sets $z_i = H_i$ for every step. Because the alignment loss during training pulls H_i into the same space as T_i, the self-feedback rollout at inference stays inside that space and no further visual encoding is required. The response is generated autoregressively after the closing look token.

A few practical points for re-implementation. The crop teacher ϕ is the base model's own visual encoder and projector, kept frozen throughout. The merged token grid $H_g \times W_g$ used for window scoring is the post-merger output of the visual encoder, and any special tokens (such as a newline token) have to be excluded when mapping window indices back to image coordinates. The adaptive selection itself is a non-differentiable pre-processing step on V and does not back-propagate through the language model. We use $n = 8$, $w = 4$, $k = \max(\lfloor M/10 \rfloor, 4)$, $s = 280$, and $\lambda = 0.1$ as defaults throughout the paper.

A.2 Latency and Throughput

Measurements are taken at inference time on the evaluation set used for the main results, with each query processed independently under the same image-resolution budget as the corresponding training stage. We discard a small warmup batch and report both the mean and the 90th percentile (P90) over the remaining requests. End-to-end latency covers the full forward pass from image encoding through autoregressive decoding; decode latency is reported per output token.

Table 6: Efficiency comparison on Qwen3-VL-4B using mean and P90 statistics. E2E latency and decode latency are measured in milliseconds, with decode latency reported per token. P90 denotes the 90th percentile. Bold indicates the lower value in each Base/Latent pair.

Metric	Stat.	Base	Latent
E2E latency	Mean	4,282.82	**592.62**
	P90	6,843.28	**728.71**
Output tokens	Mean	161.28	**5.84**
	P90	263.00	**11.00**
Decode latency	Mean	**23.83**	25.24
	P90	**23.92**	25.49

The Latent configuration cuts the average output length by $27\times$ (5.84 vs. 161.28 tokens), which is the dominant driver of the $7.2\times$ E2E speed-up at P90. Per-token decode latency is comparable in

Algorithm 1 TraViS unified training and inference procedure.

Require: image I, question q, latent steps n, alignment weight λ, window size w, top-k size k, frozen crop teacher ϕ (training only)

Encode the global visual representation $V = \{v_1, \ldots, v_M\}$ once. ▷ ϕ is removed at inference
Initialize the input sequence $[I, q, \langle\text{look}\rangle, z_1, \ldots, z_n, \langle/\text{look}\rangle, \varnothing]$.
for $i = 1, \ldots, n$ **do**
  Compute H_i at position $p_i - 1$ via Transformer self-attention.
  if mode = **train then**
    if $i < n$ **or** region R unavailable **then**
      Compute attention scores $a_i = q(H_i)K^\top/\sqrt{d_k}$ averaged across heads.
      Select $\mathcal{S}_i = \mathrm{TopK}(a_i, k)$ and the densest $w \times w$ window $(r_i^\star, c_i^\star)$.
      Map $(r_i^\star, c_i^\star)$ back to image coordinates and obtain crop B_i.
      Encode the target $T_i = \mathrm{sg}[\phi(B_i)]$.
    else
      Set $B_n = \mathrm{bbox}(R)$, $T_n = \mathrm{sg}[\phi(B_n)]$. ▷ terminal anchor
    end if
    Accumulate alignment $\mathcal{L}_{\text{align}} \mathrel{+}= \|H_i - T_i\|_2^2$.
    Set the slot input $z_i = \mathrm{sg}[T_i]$. ▷ target feedback
  else
    Set the slot input $z_i = H_i$. ▷ self-feedback
  end if
end for
if mode = **train then**
  Generate the response o autoregressively; compute $\mathcal{L}_{\text{CE}}$.
  Update parameters on $\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda \cdot \frac{1}{n}\mathcal{L}_{\text{align}}$ with ϕ frozen.
else
  Generate the response o autoregressively and **return** o.
end if

the two configurations because the autoregressive decoder is unchanged; the gap is entirely on the response side.

A.3 LR and Output-Head Ablation Details

Both tables below report the per-benchmark numbers underlying Figure 5. The first covers the learning-rate ablation ((a) and (b)) and the second covers the output-head ablation ((c)). In both cases the rightmost Avg. column is the mean over the six benchmarks.

Table 7: Learning-rate ablation. Per-benchmark scores across six learning rates; the rightmost column is the mean over the six benchmarks. Values reproduce (a) and (b) of Figure 5. Best per benchmark is **bold**.

LR	MMVP	MMStar	Hallu.	BLINK	MMBench	CV-Bench	Avg.
5×10^{-7}	66.0	63.0	70.0	62.0	85.0	86.5	72.00
1×10^{-6}	65.0	63.0	70.5	63.0	85.5	86.0	**72.05**
3×10^{-6}	64.0	62.5	67.5	61.0	84.0	84.0	70.34
5×10^{-6}	63.0	60.0	68.0	60.0	83.0	85.0	69.66
8×10^{-6}	55.0	61.0	66.5	58.0	81.0	83.0	67.22
1×10^{-5}	55.5	59.0	67.0	52.0	81.0	81.0	65.68

The alignment loss clearly benefits from conservative optimization. The per-step targets depend on the running model and so lag behind when updates are aggressive. We vary the learning rate across two orders of magnitude; the result in Table 7 is a sharp but uneven sensitivity. The average is essentially flat between 5×10^{-7} and 1×10^{-6}, sitting at 72.00% and 72.05% respectively, and then drops monotonically past 1×10^{-6} to reach 65.68% at 1×10^{-5}, a 6.4-point gap. The degradation is uneven across benchmarks. The fine-grained, localized-evidence tasks (MMVP, BLINK, CV-Bench)

Table 8: Output-head ablation. Per-benchmark scores for direct prediction, an MLP head, and a GLU head. The rightmost column is the mean over the six benchmarks and is computed from the per-benchmark scores. Values reproduce (c) of Figure 5. Best per benchmark is **bold**.

Output head	MMVP	MMStar	Hallu.	BLINK	MMBench	CV-Bench	Avg.
MLP Head	55.3	59.3	64.0	55.4	80.7	82.3	66.17
GLU Head	54.7	59.5	66.2	54.5	80.7	77.0	65.43
No Head	**64.7**	**63.1**	**70.1**	**63.1**	**85.4**	**85.9**	**72.05**

lose ten to twelve points between 1×10^{-6} and 8×10^{-6}, while global-recognition tasks (MMBench, MMStar) decline more gradually and HallusionBench stays within three points throughout. The asymmetry mirrors the failure-mode pattern from Section 4. Tasks that depend on precisely grounded visual evidence are the most sensitive to learning rate, with alignment noise growing with the rate at which the latent targets shift.

It is natural to ask whether a learned projection head could refine the alignment, given that the loss already pulls H_i into the target space. Table 8 suggests no. The no-head configuration wins on every one of the six benchmarks, with an average of 72.05%, exactly matching the best learning-rate setting, while the MLP Head averages 66.17% and the GLU Head 65.43%. The two learned-head configurations are separated by less than a point and well within run-to-run noise, so the practical takeaway is that any learned projection on top of the latent state hurts. The damage is largest where the task depends on localized grounding. MMVP and BLINK each lose more than nine points when a head is added, while MMStar and HallusionBench are more tolerant, losing four to six. This is consistent with an alignment loss that has already aligned H_i with T_i. A learned head has no new information to encode and tends to overfit the supervision it is given.

A.4 Target Stationarity under a Frozen Teacher

During training, the context-adaptive visual target T_i enters the alignment loss as a fixed reference signal. Whether that signal stays locally stationary across successive optimization steps is therefore a question about the selection step that constructs B_i, since the target encoder is frozen.

Fix an image I, a question q, and a frozen target encoder ϕ (visual encoder plus projector). For look step $i < n$ at training iteration τ, let $a_i^{(\tau)}$ be the compatibility scores from equation equation 5 and let $s_i^{(\tau)}(r,c)$ be the window scores in equation equation 8. Denote the k-th and $(k+1)$-th ordered token scores by $a_{i,(k)}^{(\tau)}$ and $a_{i,(k+1)}^{(\tau)}$, and the largest and second-largest window scores by $s_{i,(1)}^{(\tau)}$ and $s_{i,(2)}^{(\tau)}$. Define the local margins

$$m_{\text{top}}^{(\tau)} = a_{i,(k)}^{(\tau)} - a_{i,(k+1)}^{(\tau)}, \qquad m_{\text{win}}^{(\tau)} = s_{i,(1)}^{(\tau)} - s_{i,(2)}^{(\tau)}. \tag{16}$$

If the score changes between two successive optimization steps satisfy $\|a_i^{(\tau+1)} - a_i^{(\tau)}\|_\infty < m_{\text{top}}^{(\tau)}/2$ and $\max_{(r,c)\in\Omega_w} |s_i^{(\tau+1)}(r,c) - s_i^{(\tau)}(r,c)| < m_{\text{win}}^{(\tau)}/2$, the top-$k$ token set and the window argmax remain unchanged on that interval. Consequently, with a frozen target encoder,

$$\mathcal{S}_i^{(\tau+1)} = \mathcal{S}_i^{(\tau)}, \qquad (r_i^{\star(\tau+1)}, c_i^{\star(\tau+1)}) = (r_i^{\star(\tau)}, c_i^{\star(\tau)}), \tag{17}$$

$$B_i^{(\tau+1)} = B_i^{(\tau)}, \qquad T_i^{(\tau+1)} = \text{sg}\Big[\phi(B_i^{(\tau+1)})\Big] = \text{sg}\Big[\phi(B_i^{(\tau)})\Big] = T_i^{(\tau)}. \tag{18}$$

The target is therefore piecewise stationary across the corresponding optimization interval. This is a local property of the target construction and does not by itself imply convergence of the latent rollout across look steps.

A.5 Conditional Stabilization of the Self-Feedback Trajectory

At inference the crop teacher is removed and the slot input is fed back from the predicted state, $z_i = H_i$ (equation equation 14). This self-feedback makes the latent trajectory evolve without external supervision, and the question is when such a rollout remains self-consistent.

For a fixed (I, q), define the augmented observable state and its successive change by

$$X_i = \begin{bmatrix} H_i \\ T_i \end{bmatrix}, \qquad D_i = \|X_i - X_{i-1}\|_2. \tag{19}$$

The full Transformer transition also depends on the preceding context, image, question, cache, and position; therefore, we do not assume a fixed-point map of H_i alone. Instead, consider a suffix of a continued rollout for which the selected crop region is unchanged and suppose that its observed joint increments satisfy the local contraction condition

$$D_{i+1} \le \kappa D_i, \qquad 0 \le \kappa < 1. \tag{20}$$

Repeated substitution gives

$$D_{\ell+r} \le \kappa^{r-1} D_{\ell+1}, \qquad r \ge 1. \tag{21}$$

Hence, for any $m > \ell$, the distance between two states along the suffix satisfies the single telescoping bound

$$\|X_m - X_\ell\|_2 = \left\| \sum_{j=\ell+1}^{m} (X_j - X_{j-1}) \right\|_2 \le \sum_{j=\ell+1}^{m} \|X_j - X_{j-1}\|_2 \tag{22}$$

$$= \sum_{j=\ell+1}^{m} D_j \le \sum_{r=1}^{m-\ell} \kappa^{r-1} D_{\ell+1} \tag{23}$$

$$= D_{\ell+1} \frac{1 - \kappa^{m-\ell}}{1 - \kappa} \le \frac{D_{\ell+1}}{1 - \kappa}. \tag{24}$$

For an indefinitely continued rollout satisfying the same contraction condition, Eq. (21) gives $D_{\ell+1} \le \kappa^{\ell - i_0} D_{i_0+1} \to 0$ for any suffix start i_0. The bound in Eq. (24) therefore vanishes as $\ell \to \infty$, so (X_i) is Cauchy in the finite-dimensional Euclidean space and converges to some $X^\star = (H^\star, T^\star)$. If the state-target residual satisfies $R_i = \|H_i - T_i\|_2 \to \varepsilon$, continuity of the norm gives

$$\|H^\star - T^\star\|_2 = \lim_{i \to \infty} \|H_i - T_i\|_2 = \varepsilon. \tag{25}$$

Accordingly, the derivation establishes only a conditional local self-consistency result. In the actual model, the look horizon is finite, so the result is interpreted as an approach-to-stability criterion rather than an unconditional convergence guarantee. When an answer region is available, the terminal target is instead imposed as the boundary condition

$$T_n = \mathrm{sg}[\phi(B_R)], \tag{26}$$

and is not treated as an additional application of the intermediate self-querying operator.

A.6 Implementation Details

TraViS is trained in two stages. The first stage is *latent SFT* on a multimodal instruction corpus, where the model learns to reproduce the visual trajectory conditioned on teacher-derived targets. The second stage is an optional *latent-fidelity GRPO* refinement on top of the SFT checkpoint, which is the configuration reported in Table 2. All stages use DeepSpeed ZeRO-3 sharding, bf16 mixed precision, gradient checkpointing, gradient clipping at 1.0, weight decay 0, and a 3% cosine warmup schedule. The visual encoder and the projector remain frozen throughout; only the language model and the output head are updated, which matches the LoRA-vs-full ablation in Table 4. Inputs are capped at $4{,}014{,}080$ pixels during latent SFT and at $3{,}008{,}064$ pixels during GRPO, with a minimum of $100{,}352$ pixels in both stages. The latent reasoning module inserts $n = 8$ look tokens between the prompt and the response, each filled with a sampled visual patch embedding drawn from a context-adaptive visual target produced by the teacher. The training loss combines the standard cross-entropy on the response with a mean-squared term between the pre-look hidden state and the target patch embedding, weighted by $\lambda = 0.1$, and the latent configuration is serialised to `latent_config.json` at the end of SFT to be reused by downstream stages. Each backbone is trained with its own learning rate and hardware scale.

We observed that DeepSpeed ZeRO-3's parameter all-gather counts become rank-dependent when the latent module issues a variable number of ViT forwards per step, which manifests as a step-level

hang after roughly one thousand iterations. We resolved this by collapsing all per-look crops into a single batched ViT forward and by issuing the q/k projections once via a pre-gathered weight, making the per-step communication pattern rank-symmetric. To avoid spurious termination on slow storage I/O, we additionally raise the NCCL heartbeat and timeout to 1800 s and disable asynchronous error handling.

Evaluation is performed on MMStar, MMBench, MMVP, BLINK, HallusionBench, and POPE under the same image-resolution budget as the corresponding training stage; for benchmarks where a public baseline uses a different budget, we adopt the baseline's budget to keep the comparison fair. Latency and throughput in Table 6 are measured on Qwen3-VL-4B with mean and P90 statistics.

A.7 GRPO Training Details

We report the implementation details of the GRPO variant shown in Table 2. GRPO is initialized from the full-parameter latent SFT checkpoint and updates the language model together with the output head, while the visual tower and merger remain frozen, matching the SFT recipe. We give the Qwen3-VL-4B configuration as a representative example; other backbones follow the same recipe with backbone-specific learning rate and hardware scale.

Training is performed on 8 NVIDIA H20 GPUs with DeepSpeed ZeRO-3, bf16 mixed precision, and gradient clipping at 1.0. The optimizer uses a learning rate of 2×10^{-6} with a 3% cosine warmup, a per-device batch size of 2, and 4 gradient accumulation steps, yielding an effective batch size of 8. For each prompt we sample $G = 8$ completions at temperature 0.9 with a maximum completion length of 384 tokens and a maximum prompt length of 2048 tokens. The KL coefficient is fixed at $\beta = 0.02$ and group-level reward scaling is disabled.

Reward and group-relative advantage. For prompt q and completion o_i, the scalar reward is

$$r_i = r_{\mathrm{acc}}(o_i) + r_{\mathrm{fmt}}(o_i) + r_{\mathrm{ext}}(o_i) + \alpha\, r_{\mathrm{lf}}(o_i), \qquad \hat{A}_i = r_i - \frac{1}{G}\sum_{j=1}^{G} r_j, \tag{27}$$

where r_{acc}, r_{fmt}, and r_{ext} are rule-based scores on the final answer, output format, and answer extractability, and r_{lf} is the latent-fidelity term introduced in the next paragraph. Because group-level reward scaling is disabled, the advantage is centered within the group but not standardized by its standard deviation.

GRPO objective. With the per-token importance ratio $\rho_{i,t} = \pi_\theta(o_{i,t} \mid q, o_{i,<t}) / \pi_{\theta_{\mathrm{old}}}(o_{i,t} \mid q, o_{i,<t})$ and the unbiased KL estimator

$$\hat{\mathbb{D}}_{\mathrm{KL}}(\pi_\theta \| \pi_{\mathrm{ref}}) = \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\left[\frac{\pi_{\mathrm{ref}}(o_{i,t}\,|\,q,o_{i,<t})}{\pi_\theta(o_{i,t}\,|\,q,o_{i,<t})} - \log\frac{\pi_{\mathrm{ref}}(o_{i,t}\,|\,q,o_{i,<t})}{\pi_\theta(o_{i,t}\,|\,q,o_{i,<t})} - 1\right], \tag{28}$$

the full GRPO loss is

$$\mathcal{L}_{\mathrm{GRPO}} = -\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\min\Big(\rho_{i,t}\,\hat{A}_i,\ \mathrm{clip}(\rho_{i,t}, 1-\epsilon, 1+\epsilon)\,\hat{A}_i\Big) + \beta\,\hat{\mathbb{D}}_{\mathrm{KL}}(\pi_\theta\|\pi_{\mathrm{ref}}), \tag{29}$$

with the standard clipping range ϵ around 1. The language model parameters are updated by minimizing $\mathcal{L}_{\mathrm{GRPO}}$ while π_{ref} is held fixed at the SFT reference checkpoint.

The reward is composed of four terms. They are (i) *accuracy* on the final answer, (ii) *format* compliance, (iii) *answer extractability*, and (iv) a *latent-fidelity* bonus. For completion o_i with post-rollout terminal latent state $H_n^{(i)}$ and teacher-derived target T_n, the latent-fidelity term is

$$r_{\mathrm{lf}}(o_i) = \max\left(0,\ 1 - \frac{1}{\tau}\max\Big(0,\ \big\|H_n^{(i)} - T_n\big\|_2 - m\Big)\right), \tag{30}$$

which is a margin-bounded, temperature-scaled similarity that saturates at 1 when $H_n^{(i)}$ lies inside the margin ball around T_n and decays linearly with the L2 residual beyond it. With the default hyperparameters $\alpha = 0.1$, $\tau = 1.0$, and $m = 0.0$, the term simplifies to $\alpha \cdot \max(0, 1 - \|H_n^{(i)} - T_n\|_2)$

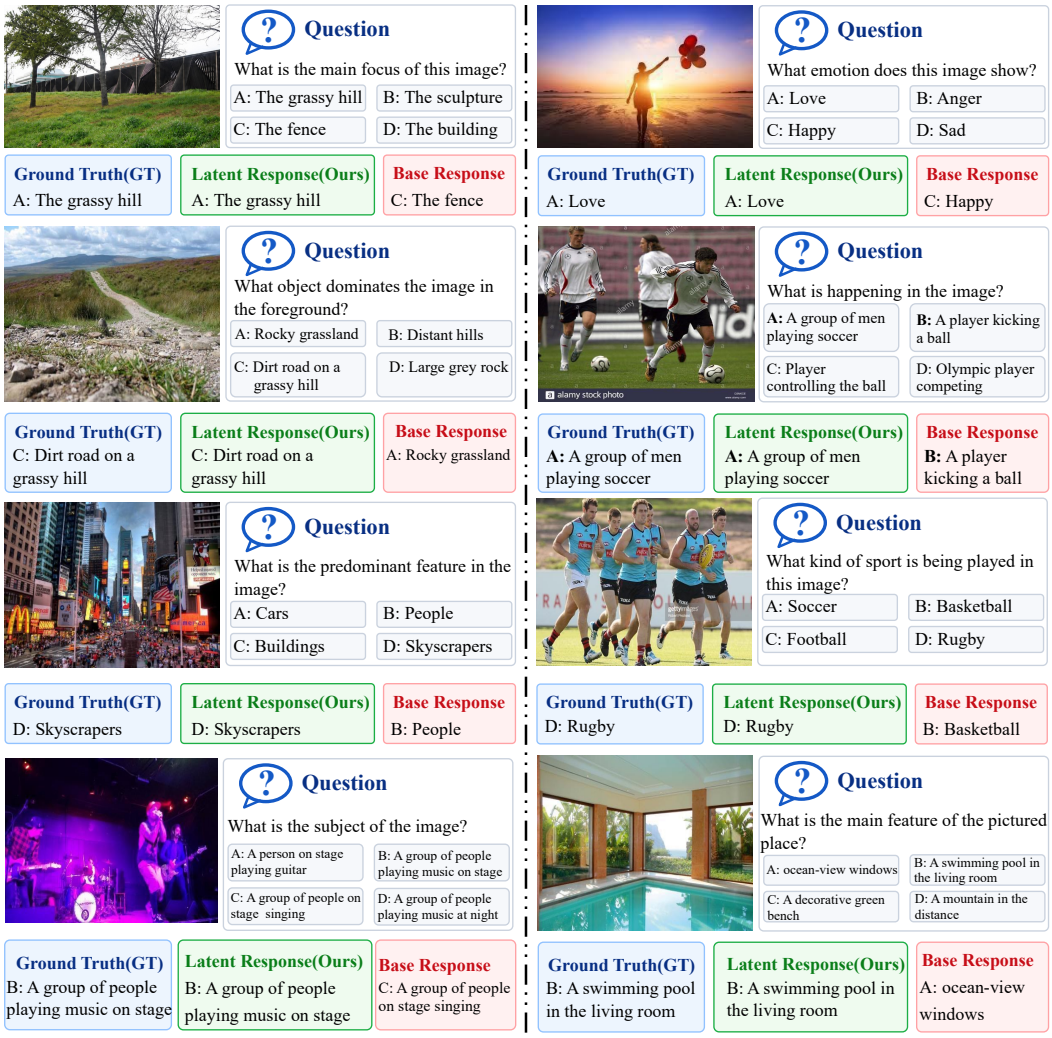

Figure 6: Eight multimodal questions where the base model fails and **TraViS** recovers the ground-truth answer. Errors cluster into two patterns: *saliency bias* (prominent but off-target picks, e.g. fence over hill) and *granularity drift* (wrong description level, e.g. rugby → basketball). **TraViS** answers every one correctly.

and serves as a soft anchor that keeps the policy close to the latent trajectory learned during SFT, preventing the GRPO update from drifting away from the visual scaffold.

GRPO training uses the VIRL39K dataset for one epoch with a maximum of 4000 steps. Checkpoints are saved every 1000 steps and the five most recent are retained.

A.8 Qualitative Analysis

The eight examples in Figure 6 illustrate what kind of mistakes the base model makes and where TraViS recovers. The errors are not random: most cluster into two recurring patterns.

Saliency bias. The base model latches onto whatever is most visually prominent in the frame, even when the question targets a less salient region. It picks the dark fence behind the grassy hill, the ocean-view windows instead of the in-ground swimming pool, the rocky foreground over the dirt road, or the pedestrians over the surrounding skyscrapers.

Granularity drift. The base model commits to a wrong level of description. It conflates adjacent categories (rugby mislabelled as basketball, love as happy), collapses a group activity into a single-actor action ("a player kicking a ball" instead of "a group playing soccer"), or substitutes a related but wrong attribute ("singing" instead of "playing music").

TraViS recovers the ground-truth answer in every one of the eight cases. This matches the broader pattern in the main results (Section 4): the largest gains are on MMVP, CV-Bench, and Hallusion-Bench, which test fine-grained discrimination and localized evidence. Together with the latent crop trajectories in Figure 4, the qualitative evidence supports the central claim of this work, namely that the visual trajectory helps by directing attention to question-relevant content rather than by adding computation.

A.9 LIMITATIONS

TraViS relies on a few design choices that bound its current scope. The number of look steps is fixed at eight across questions and benchmarks; per-question optimal length may be shorter for simple lookups or deeper for multi-hop reasoning. The top-k window selection assumes that question-relevant regions are identifiable by a pre-trained target module, which may not hold when a question's semantics depend on long-range context or scene-level cues. The method also adds latent computation, although it remains cheaper than explicit chain-of-thought decoding. Empirically, the gains concentrate on benchmarks that require fine-grained discrimination (MMVP, CV-Bench, HallusionBench) and are smaller on tasks dominated by global semantics, suggesting that visual trajectories are most useful when the relevant evidence is local rather than scene-level. Finally, self-feedback inference relies on the alignment loss having converged well; if alignment is undertrained, the substituted state drifts from the teacher target and the rollout loses its grounding.