ATLAS: Agentic or Latent Visual Reasoning? One Word is Enough for Both

Ziyu Guo[1,2] **Rain Liu**[1] **Xinyan Chen**[2] **Pheng-Ann Heng**[2]
[1]Meta AI [2]The Chinese University of Hong Kong
guoziyu86@gmail.com
https://atlas-oneword.github.io

Abstract

Visual reasoning, often interleaved with intermediate visual states, has emerged as a promising direction in the field. A straightforward approach is to directly generate images via unified models during reasoning, but this is computationally expensive and architecturally non-trivial. Recent alternatives include agentic reasoning through code or tool calls, and latent reasoning with learnable hidden embeddings. However, agentic methods incur context-switching latency from external execution, while latent methods lack task generalization and are difficult to train with autoregressive parallelization. To combine their strengths while mitigating their limitations, we propose **ATLAS**, a framework in which a single discrete "word", termed as a functional token, serves both as an agentic operation and a latent visual reasoning unit. Each functional token is associated with an internalized visual operation, yet requires no visual supervision and remains a standard token in the tokenizer vocabulary, which can be generated via next-token prediction. This design avoids verbose intermediate visual content generation, while preserving compatibility with the vanilla scalable SFT and RL training, without architectural or methodological modifications. To further address the sparsity of functional tokens during RL, we introduce Latent-Anchored GRPO (LA-GRPO), which stabilizes the training by anchoring functional tokens with a statically weighted auxiliary objective, providing stronger gradient updates. Extensive experiments and analyses demonstrate that ATLAS achieves superior performance on challenging benchmarks while maintaining clear interpretability. We hope ATLAS offers a new paradigm inspiring future visual reasoning research.

1 Introduction

The rapid evolution of Vision-Language Models (VLMs) [2, 4, 5, 20, 28, 46, 54, 64, 67] has advanced multimodal intelligence from perception toward reasoning [1, 16, 24, 62]. In these tasks, purely textual reasoning is often insufficient, as problem solving frequently requires intermediate visual analysis [10, 17, 47, 72]. This capability, commonly studied as interleaved visual reasoning, involves generating, perceiving, and using intermediate visual states to guide subsequent inference [9, 33, 44, 51, 68, 73]. For instance, game solving may require updating the board state after each operation, while geometry solving may require constructing auxiliary lines to reveal hidden relations [22, 69, 70]. Despite strong progress in direct visual understanding, current VLMs still remain limited in this dynamic visual reasoning process.

Unified models [8, 12, 15, 26, 29, 31, 38, 42, 50, 59, 61, 72, 74] provide a straightforward solution by explicitly generating pixel-level images, as illustrated in Fig. 1.I. This paradigm is intuitive: the model externalizes intermediate visual representations in the same modality as the input. However,

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

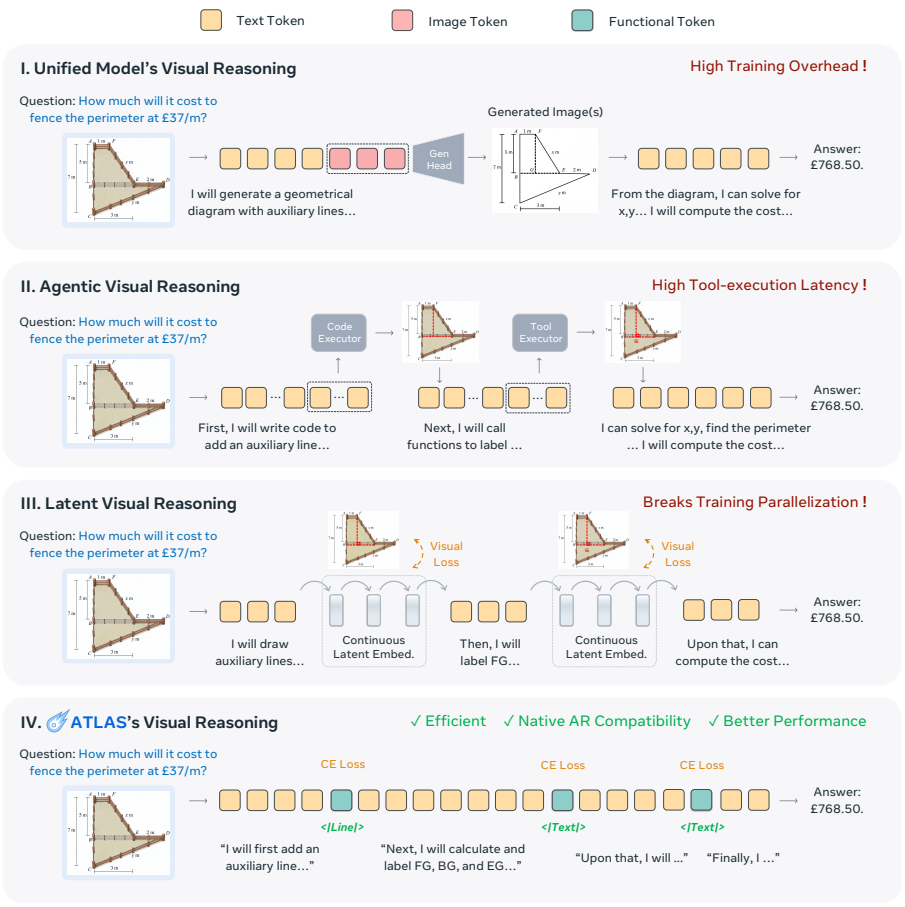


Figure 1: **Comparison of Visual Reasoning Paradigms. I**: Unified models generate intermediate pixel-level images. **II**: Agentic methods rely on external code or tool execution. **III**: Latent methods conduct intermediate reasoning through latent embeddings. **IV**: ATLAS bridges agentic and latent visual reasoning through discrete vocabulary functional tokens within the standard autoregressive generation loop, more efficient and effective.

generating new images introduces substantial inference cost and training difficulty. The model must allocate significant capacity to image decoding and re-encoding, and requires non-trivial framework-level architectural designs, which often necessitates pre-training from scratch.

To better preserve the standard VLM architecture, existing methods explore two alternative routes. First, ***agentic visual reasoning*** [19, 22, 25, 30, 32, 52, 53] in Fig. 1.II, treats the VLM as a high-level controller that generates code or tool calls to manipulate the visual input through external modules. Although its computational overhead is lower than that of generating full intermediate images, it still often requires verbose code or tool-call formulations even for simple visual operations, increasing output length and inference latency. Second, ***latent reasoning*** [27, 45, 57] in Fig. 1.III, performs intermediate reasoning in hidden representations rather than generating images or long textual operations. However, the supervision signals for latent embeddings are derived from a specific range of tasks, limiting their generalization to broader domains. More critically, they introduce recurrent latent dependencies [21], which break the compatibility with standard parallel training and substantially increase training cost.

In this paper, we propose **ATLAS**, a framework in which only a single functional "word" serves as both an agentic operation and a latent reasoning unit, as illustrated in Fig. 1.IV. The key idea of ATLAS is to represent each visual operation as a standard discrete token in the tokenizer vocabulary, such as zooming into a region, constructing auxiliary lines, drawing shapes, adding arrows, or inserting textual labels. These tokens are generated through ordinary next-token prediction within the same sequence as natural language tokens, rather than being modeled as continuous latent states outside the autoregressive sequence.

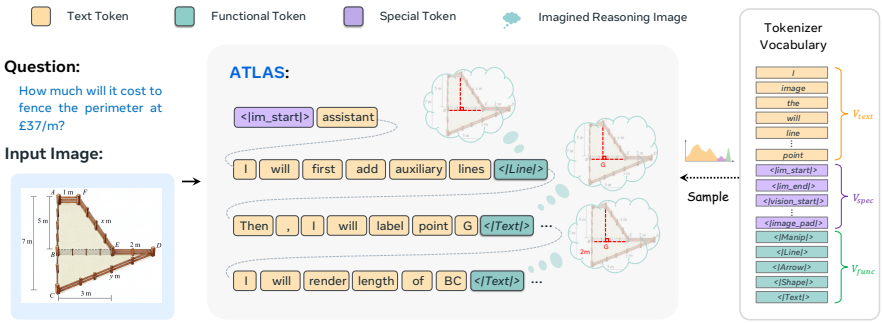


Figure 2: **Overall Pipeline of ATLAS.** ATLAS represents visual operations as functional tokens within the standard autoregressive sequence, enabling the model to perform visual reasoning without generating intermediate images or invoking external tools.

Compared with agentic methods, ATLAS provides a compact and efficient interface that internalizes complex code generation, tool calling, and external execution into a single token. Compared with latent methods, ATLAS maintains a standard autoregressive generation loop without any visual supervision, preserving compatibility with existing supervised fine-tuning (SFT) and reinforcement learning (RL) frameworks, enabling efficient parallel training with scalability to larger-size models and data. It is also worth noting that these functional tokens do not require image-level supervision. Instead, they are optimized with the standard cross-entropy (CE) objective over token sequences, allowing the model to learn from the reasoning context by itself when and how to invoke them as effective visual operations.

We adopt a two-stage training recipe for ATLAS. First, to provide a reliable cold start for using functional tokens, we curate a new dataset, ATLAS-178K, covering over 40 visual reasoning tasks collected and reformulated from existing efforts [44]. Each example is annotated with functional-token trajectories that specify the desired visual operations, enabling the model to learn when and how to invoke functional tokens within standard autoregressive generation. On top of this, we apply RL to enhance visual reasoning through outcome-driven optimization. Thanks to our designs that functional tokens are represented as ordinary vocabulary tokens, ATLAS can be optimized directly with standard GRPO [48, 56], without introducing customized training modifications [18, 23, 35, 63]. We leverage a diverse reward ensemble that jointly encourages answer correctness, valid functional-token usage, and coherent reasoning behavior, which already yields improvements over the SFT model.

However, during RL training, we observe a critical "gradient dilution" issue: the sparse functional tokens responsible for visual reasoning are overwhelmed by the much larger number of ordinary text tokens, leading to insufficient optimization. To mitigate this, we introduce Latent-Anchored GRPO (LA-GRPO), which augments the standard GRPO objective with a statically weighted token-level auxiliary loss anchored on the functional-token vocabulary. This auxiliary objective provides a persistent learning signal for functional tokens, yielding consistent performance gains across reasoning tasks.

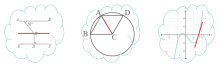
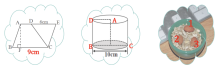
Our contributions are summarized as follows:

- We propose ATLAS, a visual reasoning framework that represents visual operations as discrete functional tokens in the standard vocabulary, avoiding verbose intermediate visual states, while preserving compatibility with scalable autoregressive training.
- We identify gradient dilution for sparse functional tokens during training and propose LA-GRPO, a token-anchored objective that strengthens functional-token optimization.
- We show that ATLAS enables compact single-token visual reasoning, achieving strong performance on challenging benchmarks with substantially reduced overhead.

2 ATLAS

In this section, we present ATLAS, a framework that bridges agentic and latent visual reasoning through discrete functional tokens. We first introduce the overall model architecture in Sec. 2.1, including the design of functional tokens within the autoregressive sequence. We then describe the training paradigm in Sec. 2.2, which consists of an SFT on the curated ATLAS-178K dataset

Table 1: **Taxonomy of Functional Tokens.** The imagined reasoning images illustrate the internal visual reasoning states, which will not be generated or supervised in the sequence.

Functional Token	Description	Imagined Reasoning Image
<|Manip|>	Enhances image through denoising, sharpening, filtering, cropping, or zoom-in based inspection.	
<|Shape|>	Marks regions for visual grounding, highlighting, or area reasoning.	
<|Line|>	Adds lines for geometric construction, visual separation, structural cues, or emphasis.	
<|Arrow|>	Indicates direction, motion, force, flow, causal relation, or highlights key visual elements.	
<|Text|>	Adds symbolic labels, numerical values, or textual notes to support multi-step reasoning.	

followed by a standard RL with GRPO [48]. Finally, in Sec. 2.3, we present the proposed LA-GRPO objective for enhanced functional-token optimization.

2.1 Model Architecture

Building upon standard autoregressive architectures [4, 5, 34], ATLAS formulates visual reasoning as next-token prediction by representing visual operations as discrete learnable functional tokens in the tokenizer vocabulary [36]. We instantiate ATLAS with Qwen2.5-VL [5] and add five functional tokens, each corresponding to an internalized operation. Generated like ordinary words within the same autoregressive sequence, these tokens provide a compact and interpretable interface for active perception and visual construction, while avoiding external tool execution, pixel-level intermediate supervision, and recurrent latent dependencies. This preserves compatibility with existing VLM pipelines and supports efficient parallel training.

Taxonomy of Functional Tokens. To internalize visual operations into the reasoning process, we expand the standard vocabulary $\mathcal{V}$ with a compact set of functional tokens. Formally, the full vocabulary is defined as $\mathcal{V} = \mathcal{V}_{\text{text}} \cup \mathcal{V}_{\text{spec}} \cup \mathcal{V}_{\text{func}}$, where $\mathcal{V}_{\text{text}}$ denotes natural language tokens, $\mathcal{V}_{\text{spec}}$ denotes the original special tokens of the VLM (*e.g.*, <im_start>, <image_pad>), and

$$\mathcal{V}_{\text{func}} = \{\texttt{<|Manip|>}, \texttt{<|Shape|>}, \texttt{<|Line|>}, \texttt{<|Arrow|>}, \texttt{<|Text|>}\}$$

denotes the five proposed functional tokens. We intentionally keep $\mathcal{V}_{\text{func}}$ compact to avoid excessive perturbation to the original token distribution of the base model. Instead of introducing many task-specific tokens, we abstract common visual operations into a small set of general categories. For instance, image restoration and enhancement operations are unified under <|Manip|>, while bounding boxes, masks, cropping, and zooming can all be represented by the generalized region-based token <|Shape|>. As summarized in Tab. 1, each functional token corresponds to a high-level visual operation that can support multi-step reasoning. This taxonomy is not intended to be exhaustive. Rather, it provides a simple and effective template for internalizing visual operations as discrete tokens. Future work can naturally extend the functional-token vocabulary to cover more diverse operations and scenarios.

Unified Sequence Modeling. Unlike agentic approaches that pause generation to call external modules, or latent methods that produce continuous hidden embeddings, ATLAS keeps the entire reasoning process within a single discrete autoregressive sequence. Given a multimodal input context $\mathbf{x}$, the model predicts an output sequence $\mathbf{y} = \{y_1, \ldots, y_T\}$, where $y_t \in \mathcal{V} = \mathcal{V}_{\text{text}} \cup \mathcal{V}_{\text{spec}} \cup \mathcal{V}_{\text{func}}$. When a functional token $y_t \in \mathcal{V}_{\text{func}}$ is predicted, it is treated as an ordinary sequence token while serving as an internal reasoning unit that specifies the type of visual operation needed at the current step. For example, <|Line|> indicates that the model should reason with an auxiliary line, while <|Text|> indicates that symbolic labels or numerical annotations may be useful for the subsequent derivation. This formulation preserves the explicitness and interpretability of agentic reasoning, while avoiding the latency of tool execution and the cost of pixel-level image generation. Importantly, functional tokens do not require any image-level supervision. Instead, they are optimized with the

same cross-entropy (CE) objective as ordinary text tokens:

$$\mathcal{L}_{\text{func}} = -\sum_{y_t \in \mathcal{V}_{\text{func}}} \log p_\theta(y_t \mid \mathbf{x}, y_{<t}).$$

Through token-level supervision, the model learns from the surrounding reasoning context when and how to invoke functional tokens as effective visual operations. For example, when the reasoning context states, *"Now I will add an auxiliary height to ..."*, the next functional token can be `<|Line|>`, encouraging the model to associate such geometric construction intent with the corresponding functional token. Since all reasoning units remain within the autoregressive sequence, ATLAS is fully compatible with scalable next-token training and inference pipelines.

2.2 Two-stage Training Recipe

We train ATLAS in two stages. First, we curate ATLAS-178K, an SFT dataset tailored to our visual reasoning paradigm with functional tokens. This provides a cold start for functional-token invocation and improved interleaved visual reasoning. Second, we apply standard GRPO for RL, further enhancing reasoning performance through reward-guided optimization.

Stage 1: SFT with ATLAS-178K. We construct ATLAS-178K to provide supervised reasoning trajectories for the SFT stage. Specifically, it is constructed through the following three steps:

1. ***Source Data and Token Extraction:*** We start from the publicly released preview subset of `V-Interaction-400K` [44], which provides image-construction code paired with visual reasoning problems, making it suitable for deriving functional-token supervision. We parse the original code and extract visual operations that can be naturally mapped to our functional-token space, including line drawing, text annotation, shape drawing, visual refinement, cropping, and other visually grounded transformations. We then filter the extracted samples and retain 138K high-quality examples covering over 40 tasks for functional-token trajectory construction.
2. ***Trajectory Construction and Polishing:*** After extracting the mapped operations, we convert them into reasoning trajectories with functional tokens. For each functional step, we insert a predefined transition template so that the functional token appears as an explicit part of the reasoning process. Since directly templated trajectories can be overly rigid, we further use Gemini-2.5-Pro [11] to polish them into more natural reasoning text while preserving the original semantics and functional-token order.
3. ***Perception Preservation:*** To preserve the model's low-level perceptual ability, we also include `V-Perception-40K` [44] during SFT. This part of data does not contain functional tokens, but provides complementary supervision for fine-grained visual understanding and helps reduce catastrophic forgetting during fine-tuning.

With this dataset, we train the model using the vanilla CE loss [37], updating all tokens in the sequence and enabling the model to learn valid functional-token invocation from context.

Stage 2: Standard RL with GRPO. While SFT provides a cold start for functional-token usage, complex multi-step reasoning further requires the model to decide when such operations are useful for reaching the correct answer. Thanks to the compatibility of ATLAS with standard autoregressive generation, we can directly adopt GRPO without introducing customized training procedures. Given a query q, the policy π_θ samples a group of G outputs $\{o_1, \ldots, o_G\}$. We define a composite reward $r(o)$ that encourages answer correctness, effective functional-token usage, and valid formatting, while penalizing overly long responses and excessive token invocation:

$$r(o) = \lambda_{\text{acc}} r_{\text{acc}} + \lambda_{\text{func}} r_{\text{func}} + \lambda_{\text{fmt}} r_{\text{fmt}} - \lambda_{\text{len}} p_{\text{len}} - \lambda_{\text{spam}} p_{\text{spam}},$$

where each λ controls the corresponding reward or penalty term defined as follows:

- **Answer Accuracy (r_{acc}):** Evaluates whether the final answer is correct. We use exact string matching and mathematical equivalence checking when applicable. The reward is 1 for a correct answer and 0 otherwise.
- **Functional Token Usage (r_{func}):** To encourage the meaningful usage of functional tokens and prevent reward hacking, we implement a strict conditional reward mechanism. The functional-token

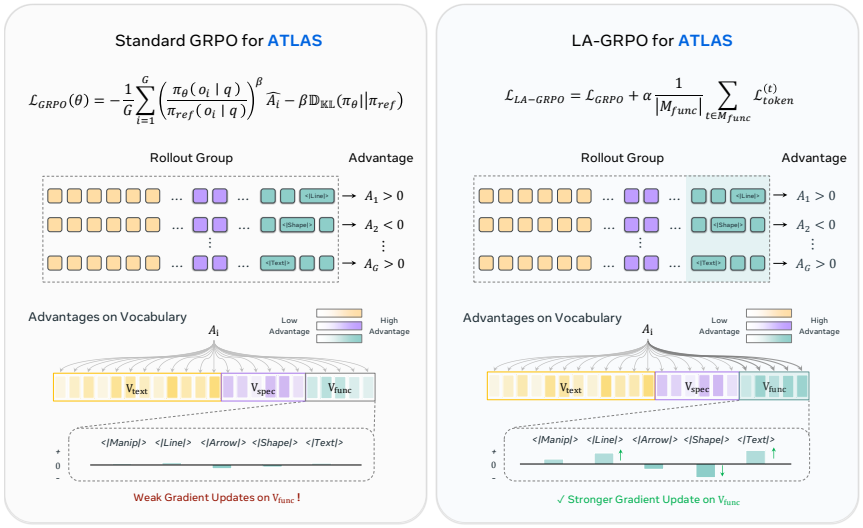


Figure 3: **Latent-Anchored GRPO.** Standard GRPO provides sequence-level advantages to all generated tokens, which can dilute the learning signal for sparse functional tokens. LA-GRPO adds a token-level auxiliary objective on $\mathcal{V}_{\text{func}}$ to stabilize functional-token optimization.

reward is granted only if the model invokes at least one functional token and successfully gets the correct final answer.

- **Format Adherence (r_{fmt}):** Ensures that the final answer follows the required format for reliable parsing. It yields 1 if the required format is satisfied and 0 otherwise.
- **Length Penalty (p_{len}):** Discourages overly verbose responses. If the output length $L(o)$ exceeds a predefined threshold $L_{\max}$, we apply a linear penalty within a fixed buffer range, capped by a maximum penalty value.
- **Token Overuse Penalty (p_{spam}):** Prevents the model from repeatedly generating functional tokens only to exploit the usage reward. Let N_{func} denote the number of functional tokens in the output. If N_{func} exceeds a threshold τ_{spam}, we apply a bounded linear penalty for excessive usage.

This reward design reflects a simple principle: functional tokens should be encouraged only when they support effective reasoning. The standard GRPO objective then optimizes the policy according to the relative advantage within the sampled group:

$$\mathcal{L}_{\text{GRPO}}(\theta) = -\frac{1}{G}\sum_{i=1}^{G}\left(\frac{\pi_\theta(o_i \mid q)}{\pi_{\text{ref}}(o_i \mid q)}\right)^{\beta}\hat{A}_i - \beta\mathbb{D}_{\text{KL}}(\pi_\theta||\pi_{\text{ref}}),$$

where $\hat{A}_i$ is the advantage computed from the group rewards $r(o_i)$, and β controls the KL penalty.

2.3 LA-GRPO

Directly applying standard GRPO to ATLAS suffers from a "gradient dilution" issue. As illustrated in Fig. 3, standard GRPO assigns a sequence-level advantage to each rollout response and propagates this signal to all generated tokens. However, functional tokens occupy only a very small portion of the sequence. In ATLAS trajectories, an average response contains 203.7 generated tokens, but only 4.8 of them are functional tokens, corresponding to a ratio of 2.3%. As a result, the learning signal for these sparse but important visual-operation tokens is easily diluted by the much larger number of ordinary text tokens. This weakens updates on $\mathcal{V}_{\text{func}}$ and may cause the model to underuse functional tokens or learn unstable behaviors such as token spamming.

To address this issue, we propose Latent-Anchored GRPO (LA-GRPO). The key idea is to keep the original sequence-level GRPO objective unchanged, while adding a functional-token anchor that explicitly strengthens optimization on $\mathcal{V}_{\text{func}}$. Concretely, for each sampled rollout, we identify the positions where functional tokens appear and apply an additional token-level auxiliary objective only to these positions. This auxiliary term reuses the rollout advantage from GRPO, but concentrates the update on functional tokens such as `<|Line|>`, `<|Shape|>`, and `<|Text|>`. In this way, LA-GRPO preserves the global reward-driven optimization of standard GRPO while providing a stronger and more persistent learning signal for the tokens responsible for internalized visual operations.

Specifically, for each rollout $o_i = \{y_1, \ldots, y_T\}$, we collect the positions of functional tokens $M_{\text{func}}(o_i) = \{t \mid y_t \in \mathcal{V}_{\text{func}}\}$. For each $t \in M_{\text{func}}(o_i)$, we define a token-level clipped surrogate loss:

$$\mathcal{L}_{\text{token}}^{(t)} = -\min\left(\rho_{i,t}\hat{A}_i,\ \text{clip}(\rho_{i,t}, 1-\epsilon, 1+\epsilon)\hat{A}_i\right), \quad \rho_{i,t} = \frac{\pi_\theta(y_t \mid q, y_{<t})}{\pi_{\theta_{\text{old}}}(y_t \mid q, y_{<t})}.$$

This objective anchors the group-level advantage directly to functional-token positions, producing stronger updates on sparse visual-operation tokens. Then, the final objective is:

$$\mathcal{L}_{\text{LA-GRPO}} = \mathcal{L}_{\text{GRPO}} + \alpha \frac{1}{|M_{\text{func}}|} \sum_{t \in M_{\text{func}}} \mathcal{L}_{\text{token}}^{(t)},$$

where α controls the anchor strength and the normalization stabilizes the auxiliary loss scale. Unlike standard GRPO, which spreads the advantage signal across the whole vocabulary, LA-GRPO explicitly reinforces $\mathcal{V}_{\text{func}}$ while retaining sequence-level reward optimization. This yields stronger gradients for functional tokens, stabilizes their invocation, and improves visual reasoning without modifying the autoregressive training pipeline.

3 Experiments

3.1 Implementation Details

Training Details. We adopt Qwen2.5-VL-7B [5] as our base model. During the SFT stage, we freeze the vision encoder and solely update the visual projector and the language model on ATLAS-178K (detailed code-to-token mappings and templates are provided in Appendix A). For the RL stage, we continue to freeze the vision encoder while optimizing the aligner and language model for 1 epoch on *We-Math 2.0* [43], *MMK12* [39], and *ThinkLite* [58], which cover diverse visual reasoning scenarios and provide diverse supervision for reinforcement learning.

Evaluation Datasets and Metrics. We evaluate ATLAS across a suite of benchmarks, including V* [60], BLINK [13], and WeMath [43]. We adopt a rigorous automated judging pipeline, where we first employ rule-based scripts to parse the answer from the model outputs and then utilize Qwen3-VL-235B-A22B-Instruct [4] as a deterministic LLM-as-a-judge to verify the correctness of the answer and the format.

3.2 Quantitative Analysis

The performance comparison across visual reasoning benchmarks is summarized in Tab. 2. Compared with the base model Qwen2.5-VL [5], ATLAS brings clear improvements across all benchmarks. The improvement is notable on BLINK, where Qwen2.5-VL achieves an average accuracy of 22.8%, while ATLAS$_{\textit{LA-GRPO}}$ reaches 51.3%. This shows that discrete vocabulary functional tokens can effectively enhance structured visual reasoning.

Among our variants, ATLAS$_{\textit{SFT}}$ already improves the BLINK average from 22.8% to 46.0%, demonstrating that supervised fine-tuning provides a strong reasoning capability. ATLAS$_{\textit{GRPO}}$ further improves the overall performance, reaching 50.5% on BLINK and 40.3% on WeMath. It brings large gains on several subsets, such as Jigsaw and Spatial Relation, but the improvements are not uniform. For example, it drops on IQ and multi-view reasoning compared with ATLAS$_{\textit{SFT}}$. This suggests that sequence-level preference optimization can improve final-answer accuracy, but may also introduce unstable functional-token behavior on structured reasoning tasks. With LA-GRPO, ATLAS achieves the best performance among our variants on WeMath and BLINK average, reaching 45.0% and 51.3%, respectively. It also improves over standard GRPO on several subsets, including Art Style, Counting, Forensic Detection, and Multi-view Reasoning, with multi-view reasoning increasing from 43.6% to 53.4%. Compared with existing visual reasoning methods, ATLAS$_{\textit{LA-GRPO}}$ obtains consistently competitive results. Although standard GRPO performs better on a few metrics such as V*, Jigsaw, and Spatial Relation, LA-GRPO gives a more balanced result across the benchmark. These results indicate that anchoring the optimization on functional tokens helps stabilize training and improves the overall effectiveness of ATLAS on complex visual reasoning tasks.

Table 2: **Performance Comparison on Visual Reasoning.** We compare five groups of VLMs with our ATLAS models on three challenging benchmarks.

Method	V*	WeMath	BLINK							
			Avg.	Art.	Count.	Forensic.	IQ	Jigsaw	M-view.	Spatial
Closed-source Models										
GPT-4o [40]	62.8	50.6	61.0	82.9	49.2	79.5	31.3	55.3	59.4	69.2
Claude-4-Sonnet [3]	15.2	63.0	49.9	61.5	59.2	35.6	30.0	53.3	47.4	62.2
Gemini-2.0-Flash [41]	73.3	47.4	45.3	56.4	55.0	30.3	25.3	48.7	43.6	58.0
Gemini-2.5-Pro [11]	79.1	71.3	74.6	85.5	78.3	89.4	43.3	85.3	50.4	90.2
Standard VLMs										
Qwen2.5-VL [5]	70.2	36.2	22.8	29.9	58.3	0.8	18.7	31.3	0.0	20.3
LLaVA-OneVision-7B [28]	75.4	23.1	36.6	47.0	43.3	25.0	20.7	38.7	33.8	47.6
MiniGPT-v2 [7]	35.6	11.0	32.8	43.6	13.3	24.2	20.3	34.7	48.9	44.8
Gemma-3-27B [55]	62.3	31.7	32.1	42.7	37.5	21.2	16.0	33.3	31.6	42.7
Unified Models										
Anole [10]	25.4	24.7	16.4	31.6	25.0	11.7	14.3	2.0	3.0	27.3
Bagel [12]	55.5	39.4	51.1	63.2	60.8	37.1	**30.0**	57.3	39.8	69.2
Agentic Visual Models										
Visual CoT [47]	44.5	28.6	44.4	47.0	57.5	25.0	20.7	52.7	44.4	63.6
V-Thinker [44]	41.4	32.5	35.0	26.9	43.3	19.7	18.7	42.0	51.1	43.4
VTS-V [6]	74.9	42.8	51.2	62.4	61.7	32.9	28.7	56.1	49.4	67.2
Latent Visual Models										
LVR [27]	77.5	41.2	49.4	59.0	60.0	35.6	25.3	52.7	48.1	65.0
MCOT [71]	76.4	39.6	47.4	55.6	57.5	33.3	26.7	50.7	45.9	62.2
CoVT [45]	72.8	38.1	47.9	59.0	60.0	36.4	24.0	41.3	49.6	65.0
Monet [57]	77.8	36.9	41.8	41.0	56.7	22.7	28.0	45.3	40.5	58.7
Ours										
ATLAS$_{SFT}$	77.5	28.9	46.0	50.4	59.2	26.5	26.0	54.7	48.1	57.3
ATLAS$_{GRPO}$	**77.9**	40.3	50.5	57.3	61.7	34.1	26.0	**57.7**	43.6	**70.6**
ATLAS$_{LA\text{-}GRPO}$	75.4	**45.0**	**51.3**	**65.0**	**62.5**	**37.9**	26.3	51.3	**53.4**	62.9

3.3 Qualitative Analysis

Qualitative Examples. Fig. 4 (and Fig. 6 in Appendix) shows representative visual reasoning trajectories from ATLAS. As shown, the model invokes functional tokens at meaningful steps: <|Shape|> is used to localize relevant regions, <|Arrow|> guides attention to additional evidence, and <|Text|> supports counting and labeling. These examples show that functional tokens are not used as isolated markers, but are naturally integrated into the reasoning process to support visual reasoning.

Attention Scores. We further visualize the image attention around functional tokens in Fig. 5 (and Fig. 7 in Appendix). For each functional token, we average the attention scores between image tokens and the nearby 10 tokens around it. The resulting maps show that different functional tokens tend to focus on relevant visual regions, *e.g.*, in Fig. 5, <|Line|> attends to the height segment in the geometry problem, and <|Shape|> highlights the cat region for spatial comparison. These patterns suggest that functional tokens are associated with meaningful visual evidence rather than being used only as textual markers.

3.4 Efficiency Analysis

We further evaluate the inference efficiency of ATLAS on BLINK-Jigsaw [13] and compare it with V-Thinker [44], which relies on explicit agentic reasoning. As shown in Tab. 3, ATLAS greatly reduces both output length and operation-formulation overhead. This is because each visual operation is represented by a single functional token, rather than a long textual description or tool-call formulation. This compact formulation directly improves inference efficiency. ATLAS reduces the average latency from 18.83*s* to 3.80*s* and lowers peak memory usage from 2.55GB to 1.43GB, while maintaining competitive accuracy. These results show that representing visual operations as compact functional tokens can reduce inference cost while preserving effective visual reasoning.



Figure 4: **Qualitative Examples of ATLAS.** The cloud-shaped regions in the middle illustrate the model's imagined reasoning states for better understanding.

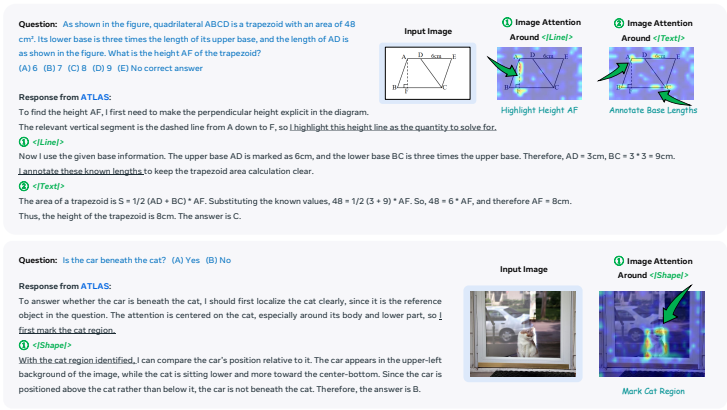


Figure 5: **Attention Analysis of Functional Tokens.** The highlighted regions show that functional tokens tend to attend to task-relevant visual evidence like geometric lines and target objects.

3.5 Ablation Study of Negative Reward Penalties

We ablate the format reward (r_{fmt}), length penalty (p_{len}), and token spam penalty (p_{spam}) to verify the necessity of each reward constraint. As shown in Tab. 4, removing any of these components leads to a performance drop on BLINK, indicating that the negative reward design is important for stable functional-token alignment. The full LA-GRPO objective achieves the best BLINK average accuracy of 51.3. When the format reward is removed, the model becomes less reliable in producing parseable final answers, leading to a drop of 1.3. Removing the length penalty causes responses to become unnecessarily verbose, which increases generation cost and hurts reasoning accuracy. The largest degradation occurs when removing the token spam penalty, where the BLINK average drops from 51.3 to 47.0. In this setting, we observe severe reward hacking: the model generates up to 18.7 functional tokens per sequence merely to accumulate r_{func}, rather than using them to support effective reasoning. Similarly, removing p_{len} increases the average sequence length by 43.8%, resulting in higher compute cost without accuracy gains. These results show that the format, length, and token-spam constraints are complementary: they encourage ATLAS to invoke functional tokens only when useful, while keeping the reasoning trajectory concise, well formatted, and robust.

Table 3: **Efficiency Comparison.** Efficiency metrics are reported as per-query averages. ATLAS substantially reduces generation overhead and latency while improving accuracy.

Method	All Tokens ↓	Code Tokens ↓	Func. Tokens ↓	Latency ↓	Peak Mem. ↓	Acc. ↑
V-Thinker [44]	489.57	350.35	-	18.83s	2.55GB	42.0
ATLAS	99.85	-	0.81	3.80s	1.43GB	57.7
Relative Gain	*4.90×*		*434.3×*	*4.96×*	*1.78×*	*+15.7*

Table 4: **Ablation Study of Negative Reward Penalties in LA-GRPO on BLINK.**

r_{fmt}	p_{len}	p_{spam}	Acc.
✓	✓	✓	51.3
-	✓	✓	50.0
✓	-	✓	49.2
✓	✓	-	47.0

4 Related Work

Agentic Visual Reasoning. Another line of research equips language or multi-modal models with the ability to act through external tools. Representative examples include program-based and code-based systems such as VISPROG [19] and ViperGPT [52], where the model generates executable programs to invoke specialized vision modules. More recent agentic frameworks broaden this idea by integrating richer tool ecosystems, hierarchical planning, and mixed-modality execution [17, 25, 30, 32, 33, 36, 44, 47, 51, 53, 65, 73, 75]. Related work also explores visual workspaces such as sketchpads, where the model produces intermediate drawings, marks, or auxiliary constructions to support subsequent reasoning [22]. These methods are effective because they allow the model to actively manipulate visual inputs rather than passively observe them once. However, the underlying execution typically happens through external programs, APIs, or auxiliary environments. This creates a disjoint reasoning loop in which visual action is performed outside the standard autoregressive computation graph. Consequently, such approaches are usually non-differentiable end-to-end, incur latency from context switching and tool execution, and often require verbose code or program generation even for relatively simple visual operations.

Latent Visual Reasoning. Latent reasoning offers a promising alternative by moving intermediate computation from explicit text into compact hidden representations [21, 71]. In language modeling, recent work explores this direction from several perspectives, including self-generated latent rationales [66], continuous thought representations [21], and recurrent-depth architectures that scale test-time computation without emitting long reasoning traces [14]. In multi-modal settings, methods such as Heima [49] compress explicit reasoning into hidden thinking tokens, while more recent approaches introduce latent visual tokens or latent visual trajectories to support multi-modal reasoning without full image generation [27, 45, 57]. These studies show that latent reasoning can improve efficiency and, in some cases, performance. Nevertheless, existing methods still face important limitations. Many rely on auxiliary supervision, reconstruction, or distillation targets for latent states [45, 57], which restricts flexibility and may limit generalization beyond the training setup. More importantly, several approaches introduce recurrent or non-standard computation patterns, which deviate from the standard next-token prediction pipeline and reduce compatibility with highly optimized autoregressive parallel training systems. In contrast, our method formulates agentic visual actions as discrete functional tokens within the normal vocabulary space, keeping the entire reasoning process strictly inside the standard autoregressive loop. This preserves end-to-end differentiability while avoiding the recurrent dependencies and external execution overhead that limit prior approaches.

5 Conclusion

In this paper, we introduced **ATLAS**, a visual reasoning framework that represents visual operations as discrete functional tokens within the standard autoregressive vocabulary. By internalizing visual reasoning into compact tokens, ATLAS avoids intermediate image generation, external tool execution, and verbose operation formulations, while preserving interpretability and compatibility with parallel autoregressive training. We further identified gradient dilution for sparse functional tokens during GRPO training and proposed Latent-Anchored GRPO to stabilize their optimization. Extensive experiments show that ATLAS establishes a distinct visual reasoning paradigm, achieving strong performance on complex benchmarks with reduced inference latency and memory usage.

Acknowledgments and Disclosure of Funding

The work described in this paper was supported by the Research Grants Council of the Hong Kong Special Administrative Region, China, under Project No. 14200824 and Project No. 14202125. The authors declare no competing interests.

References

[1] Ruichuan An, Sihan Yang, Ziyu Guo, Wei Dai, Zijun Shen, Haodong Li, Renrui Zhang, Xinyu Wei, Guopeng Li, Wenshan Wu, et al. Genius: Generative fluid intelligence evaluation suite. *arXiv preprint arXiv:2602.11144*, 2026.

[2] 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.

[3] Anthropic. System card: Claude Opus 4 & Claude Sonnet 4. Technical report, Anthropic, May 2025. Available at https://www.anthropic.com/claude-4-system-card.

[4] 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*, 2025.

[5] 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*, 2025.

[6] Tianyi Bai, Zengjie Hu, Fupeng Sun, Qiu Jiantao, Yizhen Jiang, Guangxin He, Bohan Zeng, Conghui He, Binhang Yuan, and Wentao Zhang. Multi-step visual reasoning with visual tokens scaling and verification. *Advances in Neural Information Processing Systems*, 38:74554–74592, 2026.

[7] Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *arXiv preprint arXiv:2310.09478*, 2023.

[8] Xinyan Chen, Ziyu Guo, Renrui Zhang, Dongzhi Jiang, and Hongsheng Li. Opencof: Learning to reason through video generation. *arXiv preprint arXiv:2607.08763*, 2026.

[9] Xinyan Chen, Renrui Zhang, Dongzhi Jiang, Aojun Zhou, Shilin Yan, Weifeng Lin, and Hongsheng Li. Mint-cot: Enabling interleaved visual tokens in mathematical chain-of-thought reasoning. *Advances in neural information processing systems*, 38:69110–69139, 2026.

[10] Ethan Chern, Jiadi Su, Yan Ma, and Pengfei Liu. Anole: An open, autoregressive, native large multimodal models for interleaved image-text generation. *arXiv preprint arXiv:2407.06135*, 2024.

[11] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

[12] Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, et al. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025.

[13] 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*, pages 148–166. Springer, 2024.

[14] Jonas Geiping, Sean McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, and Tom Goldstein. Scaling up test-time compute with latent reasoning: A recurrent depth approach. *Advances in Neural Information Processing Systems*, 38:41340–41391, 2026.

[15] Ziyu Guo, Xinyan Chen, Renrui Zhang, Ruichuan An, Yu Qi, Dongzhi Jiang, Xiangtai Li, Manyuan Zhang, Hongsheng Li, and Pheng-Ann Heng. Are video models ready as zero-shot reasoners? an empirical study with the mme-cof benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9175–9184, 2026.

[16] Ziyu Guo, Renrui Zhang, Hao Chen, Jialin Gao, Dongzhi Jiang, Jiaze Wang, and Pheng-Ann Heng. Sciverse: Unveiling the knowledge comprehension and visual reasoning of lmms on multi-modal scientific problems. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 19683–19704, 2025.

[17] Ziyu Guo, Renrui Zhang, Hongyu Li, Manyuan Zhang, Xinyan Chen, Sifan Wang, Yan Feng, Peng Pei, and Pheng-Ann Heng. Thinking-while-generating: Interleaving textual reasoning throughout visual generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26295–26305, 2026.

[18] Ziyu Guo, Renrui Zhang, Chengzhuo Tong, Zhizheng Zhao, Rui Huang, Haoquan Zhang, Manyuan Zhang, Jiaming Liu, Shanghang Zhang, Peng Gao, et al. Can we generate images with cot? let's verify and reinforce image generation step by step. *arXiv preprint arXiv:2501.13926*, 2025.

[19] Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14953–14962. IEEE, 2023.

[20] Jiaming Han, Renrui Zhang, Wenqi Shao, Peng Gao, Peng Xu, Han Xiao, Kaipeng Zhang, Chris Liu, Song Wen, Ziyu Guo, et al. Imagebind-llm: Multi-modality instruction tuning. *arXiv preprint arXiv:2309.03905*, 2023.

[21] 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.

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

[23] Dongzhi Jiang, Ziyu Guo, Renrui Zhang, Zhuofan Zong, Hao Li, Le Zhuo, Shilin Yan, Pheng-Ann Heng, and Hongsheng Li. T2i-r1: Reinforcing image generation with collaborative semantic-level and token-level cot. *Advances in Neural Information Processing Systems*, 38:39856–39890, 2026.

[24] Dongzhi Jiang, Renrui Zhang, Ziyu Guo, Yanwei Li, Yu Qi, Xinyan Chen, Liuhui Wang, Jianhan Jin, Claire Guo, Shen Yan, et al. Mme-cot: Benchmarking chain-of-thought in large multimodal models for reasoning quality, robustness, and efficiency. *arXiv preprint arXiv:2502.09621*, 2025.

[25] Dongzhi Jiang, Renrui Zhang, Ziyu Guo, Yanmin Wu, Jiayi Lei, Pengshuo Qiu, Pan Lu, Zehui Chen, Chaoyou Fu, Guanglu Song, et al. Mmsearch: Benchmarking the potential of large models as multi-modal search engines. *arXiv preprint arXiv:2409.12959*, 2024.

[26] Dongzhi Jiang, Renrui Zhang, Haodong Li, Zhuofan Zong, Ziyu Guo, Jun He, Claire Guo, Junyan Ye, Rongyao Fang, Weijia Li, et al. Draco: Draft as cot for text-to-image preview and rare concept generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5903–5914, 2026.

[27] 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*, volume 2026, pages 148076–148090, 2026.

[28] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

[29] Chengzu Li, Wenshan Wu, Huanyu Zhang, Yan Xia, Shaoguang Mao, Li Dong, Ivan Vulić, and Furu Wei. Imagine while reasoning in space: Multimodal visualization-of-thought. *arXiv preprint arXiv:2501.07542*, 2025.

[30] Haodong Li, Tianfei Ren, Xiaoxiao Ma, Chunmei Qing, Zhen Fang, Sipeng He, Ziyu Guo, Haoyu Wu, Juanxi Tian, Yihang Zou, et al. Videococo: Code-as-cot for physically-consistent video generation via an agentic dual-engine system. *arXiv preprint arXiv:2607.27380*, 2026.

[31] Haodong Li, Jingwei Wu, Quan Sun, Guopeng Li, Juanxi Tian, Huanyu Zhang, Yanlin Lai, Ruichuan An, Hongbo Peng, Yuhong Dai, et al. Gebench: Benchmarking image generation models as gui environments. *arXiv preprint arXiv:2602.09007*, 2026.

[32] Hongyu Li, Manyuan Zhang, Kaituo Feng, Shu Chen, Dian Zheng, Hao Li, Hao Yu, Zhangquan Chen, Zoey Guo, Ray Zhang, et al. Onesearch-vl: Unified multimodal deep research agent for image and video. *arXiv preprint arXiv:2610.12419*, 2026.

[33] Hongyu Li, Manyuan Zhang, Dian Zheng, Ziyu Guo, Yimeng Jia, Kaituo Feng, Hao Yu, Yexin Liu, Yan Feng, Peng Pei, et al. Editthinker: Unlocking iterative reasoning for any image editor. *arXiv preprint arXiv:2512.05965*, 2025.

[34] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024.

[35] Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl. *Advances in neural information processing systems*, 38:40783–40818, 2026.

[36] Jinxiu Liu, Jianru Li, Tanqing Kuang, Xuanming Liu, Kangfu Mei, Yandong Wen, and Weiyang Liu. Symbomni: Evolving agentic omni models via symbolic concept learning. *arXiv preprint arXiv:2607.12042*, 2026.

[37] Anqi Mao, Mehryar Mohri, and Yutao Zhong. Cross-entropy loss functions: Theoretical analysis and applications. In *International conference on Machine learning*, pages 23803–23828. pmlr, 2023.

[38] Kangfu Mei, Mo Zhou, and Vishal Patel. Field-dit: Diffusion transformer on unified video, 3d, and game field generation. In *International Conference on Learning Representations*, volume 2025, pages 95975–95988, 2025.

[39] Fanqing Meng, Lingxiao Du, Zongkai Liu, Zhixiang Zhou, Quanfeng Lu, Daocheng Fu, Tiancheng Han, Botian Shi, Wenhai Wang, Junjun He, et al. Mm-eureka: Exploring the frontiers of multimodal reasoning with rule-based reinforcement learning. *arXiv preprint arXiv:2503.07365*, 2025.

[40] OpenAI. GPT-4o. https://openai.com/index/hello-gpt-4o/, 2024.

[41] Sundar Pichai, Demis Hassabis, and Koray Kavukcuoglu. Introducing Gemini 2.0: our new AI model for the agentic era. Google Blog, 2024. Accessed: 2024-12-11.

[42] Yu Qi, Xinyi Xu, Ziyu Guo, Siyuan Ma, Renrui Zhang, Xinyan Chen, Ruichuan An, Ruofan Xing, Jiayi Zhang, Haojie Huang, et al. Mme-cof-pro: Evaluating reasoning coherence in video generative models with text and visual hints. *arXiv preprint arXiv:2603.20194*, 2026.

[43] Runqi Qiao, Qiuna Tan, Guanting Dong, MinhuiWu, Chong Sun, Xiaoshuai Song, Jiapeng Wang, Zhuoma Gongque, Shanglin Lei, Yifan Zhang, et al. We-math: Does your large multimodal model achieve human-like mathematical reasoning? In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 20023–20070, 2025.

[44] Runqi Qiao, Qiuna Tan, Minghan Yang, Guanting Dong, Peiqing Yang, Shiqiang Lang, Enhui Wan, Xiaowan Wang, Yida Xu, Lan Yang, et al. V-thinker: Interactive thinking with images. *arXiv preprint arXiv:2511.04460*, 2025.

[45] Yiming Qin, Bomin Wei, Jiaxin Ge, Konstantinos Kallidromitis, Stephanie Fu, Trevor Darrell, and XuDong Wang. Chain-of-visual-thought: Teaching vlms to see and think better with continuous visual tokens. *arXiv preprint arXiv:2511.19418*, 2025.

[46] Bytedance Seed. Seed1. 8 model card: Towards generalized real-world agency. *arXiv preprint arXiv:2603.20633*, 2026.

[47] Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. Visual cot: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning. *Advances in Neural Information Processing Systems*, 37:8612–8642, 2024.

[48] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[49] Xuan Shen, Yizhou Wang, Yufa Zhou, Xiangxi Shi, Pu Zhao, Yanzhi Wang, and Jiuxiang Gu. Efficient reasoning with hidden thinking. *arXiv preprint arXiv:2501.19201*, 2025.

[50] Zijun Shen, Sihan Yang, Ruichuan An, Ziyu Guo, Hao Liang, Ming Lu, Renrui Zhang, and Wentao Zhang. Uni-synergy: Bridging understanding and generation for personalized reasoning via co-operative reinforcement learning. *arXiv preprint arXiv:2605.10445*, 2026.

[51] Alex Su, Haozhe Wang, Weiming Ren, Fangzhen Lin, and Wenhu Chen. Pixel reasoner: Incentivizing pixel space reasoning via curiosity-driven reinforcement learning. *Advances in Neural Information Processing Systems*, 38:8222–8251, 2026.

[52] Dídac Surís, Sachit Menon, and Carl Vondrick. Vipergpt: Visual inference via python execution for reasoning. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11854–11864. IEEE, 2023.

[53] Weidong Tang, Ruiheng Wang, Jingxuan Zhang, Hexin Wang, Haoyuan Chen, Zhengfeiyang Zhang, Yixing Ma, Zichen Jiang, Ziyu Guo, Haodong Li, et al. The trinity: Capability-driven recursive self-improvement for multimodal agents. *arXiv preprint arXiv:2610.36077*, 2026.

[54] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

[55] Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

[56] Chengzhuo Tong, Ziyu Guo, Renrui Zhang, Wenyu Shan, Xinyu Wei, Zhenghao Xing, Hongsheng Li, and Pheng-Ann Heng. Delving into rl for image generation with cot: A study on dpo vs. grpo. *Advances in Neural Information Processing Systems*, 38:115632–115655, 2026.

[57] Qixun Wang, Yang Shi, Yifei Wang, Yuanxing Zhang, Pengfei Wan, Kun Gai, Xianghua Ying, and Yisen Wang. Monet: Reasoning in latent visual space beyond image and language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12030–12040, 2026.

[58] Xiyao Wang, Zhengyuan Yang, Chao Feng, Hongjin Lu, Linjie Li, Chung-Ching Lin, Kevin Lin, Furong Huang, and Lijuan Wang. Sota with less: Mcts-guided sample selection for data-efficient visual reasoning self-improvement. *Advances in Neural Information Processing Systems*, 38:118818–118850, 2026.

[59] Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, et al. Janus: Decoupling visual encoding for unified multimodal understanding and generation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12966–12977. IEEE, 2025.

[60] Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal llms. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13084–13094. IEEE, 2024.

[61] Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. In *International Conference on Learning Representations*, volume 2025, pages 28240–28264, 2025.

[62] Junxiang Xu, Ruisi Wang, Fanyi Pu, Maijunxian Wang, Ran Ji, Tongxi Zhou, Chenyang Gu, Jing Zuo, Hongcan Xiao, Yimeng Geng, et al. Vbvr-pro: A scalable and verifiable suite for native visual reasoning. *arXiv preprint arXiv:2608.26105*, 2026.

[63] Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, et al. Dancegrpo: Unleashing grpo on visual generation. *arXiv preprint arXiv:2505.07818*, 2025.

[64] Shilin Yan, Jiaming Han, Joey Tsai, Hongwei Xue, Rongyao Fang, Lingyi Hong, Ziyu Guo, and Ray Zhang. Crosslmm: Decoupling long video sequences from lmms via dual cross-attention mechanisms. *arXiv preprint arXiv:2505.17020*, 2025.

[65] Xiaoran Yang, Yuyang Du, Kexin Chen, Soung Chang Liew, Jiamin Lu, Ziyu Guo, Xiaoyan Liu, Qun Yang, Shiqi Xu, Xingyu Fan, et al. Indusgcc: A data benchmark and evaluation framework for gui-based general computer control in industrial automation. *arXiv preprint arXiv:2509.01199*, 2025.

[66] 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.

[67] Fan Zhang, Zebang Cheng, Chong Deng, Haoxuan Li, Zheng Lian, Qian Chen, Huadai Liu, Wen Wang, YiFan Zhang, Renrui Zhang, et al. Mme-emotion: A holistic evaluation benchmark for emotional intelligence in multimodal large language models. In *International Conference on Learning Representations*, volume 2026, pages 48767–48807, 2026.

[68] Haotian Zhang, Fengyuan Yu, Dezhi Luo, Haoran Sun, Zehong Zhao, Qingying Gao, Yihan Li, Siyuan An, Huayi Qin, Yilan Zhang, et al. Training object permanence in world models. *arXiv preprint arXiv:2609.28654*, 2026.

[69] Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, et al. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In *European Conference on Computer Vision*, pages 169–186. Springer, 2024.

[70] Renrui Zhang, Xinyu Wei, Dongzhi Jiang, Ziyu Guo, Yichi Zhang, Chengzhuo Tong, Jiaming Liu, Aojun Zhou, Shanghang Zhang, Gao Peng, et al. Mavis: Mathematical visual instruction tuning with an automatic data engine. In *International Conference on Learning Representations*, volume 2025, pages 87955–87989, 2025.

[71] 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.

[72] Shanshan Zhao, Xinjie Zhang, Jintao Guo, Jiakui Hu, Lunhao Duan, Minghao Fu, Yong Xien Chng, Guo-Hua Wang, Qing-Guo Chen, Zhao Xu, et al. Unified multimodal understanding and generation models: Advances, challenges, and opportunities. *arXiv preprint arXiv:2505.02567*, 2025.

[73] Dian Zheng, Harry Lee, Manyuan Zhang, Kaituo Feng, Zoey Guo, Ray Zhang, and Hongsheng Li. Interleavethinker: Reinforcing agentic interleaved generation. *arXiv preprint arXiv:2606.13679*, 2026.

[74] Dian Zheng, Manyuan Zhang, Hongyu Li, Kai Zou, Hongbo Liu, Ziyu Guo, Kaituo Feng, Yexin Liu, Ying Luo, and Hongsheng Li. Aia: Rethinking architecture decoupling strategy in unified multimodal model. *arXiv preprint arXiv:2511.22663*, 2025.

[75] Ziwei Zheng, Minghao Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, and Chao Shen. Deepeyes: Incentivizing" thinking with images" via reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pages 126775–126798, 2026.

A Additional Implementation and Data Details

Additional Implementation Details. All experiments are conducted using DeepSpeed ZeRO-3 optimization and `bfloat16` precision on 64 NVIDIA A100 GPUs. During SFT training, the model is trained for 2 epochs using a cosine learning rate scheduler with a peak learning rate of 1×10^{-5}, a warmup ratio of 0.05, and no weight decay. We set the per-device batch size to 4 with 4 gradient accumulation steps. The maximum sequence length is limited to 8,192, and the input image resolution is dynamically bounded between 200,704 and 6,272,000 pixels. The maximum gradient norm is clipped at 1.0.

For LA-GRPO training, we employ the AdamW optimizer ($\beta_1 = 0.9, \beta_2 = 0.95, \epsilon = 10^{-8}$) with a peak learning rate of 1×10^{-6}, a weight decay of 0.1, and a warmup ratio of 0.05. The maximum gradient norm is set to 0.5. For each query, we sample $G = 8$ responses with temperature 0.8 and top-p 0.95. The KL divergence penalty coefficient (β) is set to 0.04. We allow up to 5 resampling attempts when needed. The reward weights and thresholds are provided in Tab. 7. Specifically, $w_{\text{base}} = 0.0$ denotes the unconditional base reward for functional-token invocation (ensuring that r_{func} is granted strictly when the final answer is correct), while $p_{\text{len}}^{\max}$ and $p_{\text{spam}}^{\max}$ denote the upper bounds capping the linear length penalty p_{len} (when output length $L(o) > L_{\max}$) and the linear token-overuse penalty p_{spam} (when functional-token count $N_{\text{func}} > \tau_{\text{spam}}$), respectively.

Training Data Details for SFT. As presented in the main paper, we parse the original code and extract visual actions that can be naturally mapped to our functional-token space. The detailed correspondences between code actions and functional tokens are summarized in Tab. 5. Based on that, for each functional step, we insert a predefined transition template so that the functional token appears as an explicit part of the reasoning process. The full set of templates is listed in Tab. 6. After enhancing by Gemini-2.5-Pro [11], the resulting SFT data contains functional-token reasoning trajectories with logically consistent steps, natural transitions, and correct final answers.

Table 5: **Mapping from Code Operations to Functional Tokens.** We parse visual operations from image-construction code and map them into a compact functional-token space.

Functional Token	Representative Code Patterns

<|Manip|>	`np.pad(...)`
	`cv2.blur(...)`
	`cv2.GaussianBlur(...)`
	`scipy.signal.convolve(...)`
	`cv2.filter2D(...)`

<|Line|>	`plt.plot([x1, x2], [y1, y2], ...)`
	`ax.plot([x1, x2], [y1, y2], ...)`
	`cv2.line(img, pt1, pt2, ...)`

<|Arrow|>	`plt.arrow(x, y, dx, dy, ...)`
	`ax.arrow(x, y, dx, dy, ...)`
	`cv2.arrowedLine(img, pt1, pt2, ...)`

<|Shape|>	`plt.fill(x, y, ...)`
	`ax.add_patch(Circle(...))`
	`ax.add_patch(Rectangle(...))`
	`cv2.rectangle(img, pt1, pt2, ...)`
	`cv2.polylines(img, pts, ...)`
	`img[y1:y2, x1:x2]`
	`PIL.Image.crop((x1, y1, x2, y2))`
	`cv2.resize(crop, ...)`
	`torchvision.transforms.Resize(...)`

<|Text|>	`plt.text(x, y, s, ...)`
	`ax.text(x, y, s, ...)`
	`cv2.putText(img, text, org, ...)`

Table 6: **Transition Templates for Functional Token Trajectories.** Each functional token is paired with several candidate templates. During data construction, one template is selected for each functional step, and the trajectory is further polished with Gemini-2.5-Pro [11] for fluency.

Functional Token	Candidate Templates
<|Manip|>	*1) To see the key visual details more clearly, I will first refine the relevant region.* *2) I will first enhance the visual evidence to reduce ambiguity in the current image.* *3) Before continuing, I improve the perceptual clarity of the image for more reliable reasoning.* *4) I will first apply a visual refinement to make the target evidence easier to inspect.*
<|Line|>	*1) To make the geometric relation explicit, I will first add an auxiliary line.* *2) I will first connect the relevant points with a line to support the next reasoning step.* *3) Before proceeding, I construct a line that helps reveal the underlying structure.* *4) I will first trace the key segment so that the following derivation is clearer.*
<|Arrow|>	*1) To clarify the direction or dependency, I will first add an arrow marker.* *2) I will first indicate the relevant direction so that the following reasoning becomes clearer.* *3) Before continuing, I use an arrow to make the directional relation explicit.* *4) I will first point to the visual evidence that should be checked next.*
<|Shape|>	*1) To better localize the relevant region, I will first mark it with a supporting shape.* *2) I will first outline the important area to make the spatial relation clearer.* *3) Before proceeding, I crop the target region for closer inspection.* *4) I will first zoom in on the relevant region to inspect the visual details more carefully.* *5) I will first highlight the object or region involved in the reasoning.*
<|Text|>	*1) To keep track of the key variables, I will first add a textual annotation.* *2) I will first label the relevant element so that the subsequent reasoning is easier to follow.* *3) Before continuing, I insert a short text label to make the symbolic information explicit.* *4) I will first annotate the known value or object to support the next reasoning step.*

Table 7: **Reward Hyperparameters for LA-GRPO.** We report the weights and thresholds used in the composite reward function.

Component	Symbol	Value
Answer accuracy weight	λ_{acc}	1.0
Functional-token usage weight	λ_{func}	0.2
Format adherence weight	λ_{fmt}	0.1
Length penalty weight	λ_{len}	3.0
Token overuse penalty weight	λ_{spam}	0.05
Base usage reward	w_{base}	0.0
Maximum sequence length	$L_{\max}$	1024
Token overuse threshold	τ_{spam}	20
Maximum length penalty	$p_{\text{len}}^{\max}$	3.0
Maximum token overuse penalty	$p_{\text{spam}}^{\max}$	1.0
Functional-token anchor weight	α	0.2
KL coefficient	β	0.04

B Additional Qualitative Analysis

Fig. 6 provides additional qualitative examples of ATLAS on diverse visual reasoning tasks, and Fig. 7 provides additional attention visualizations around functional tokens.

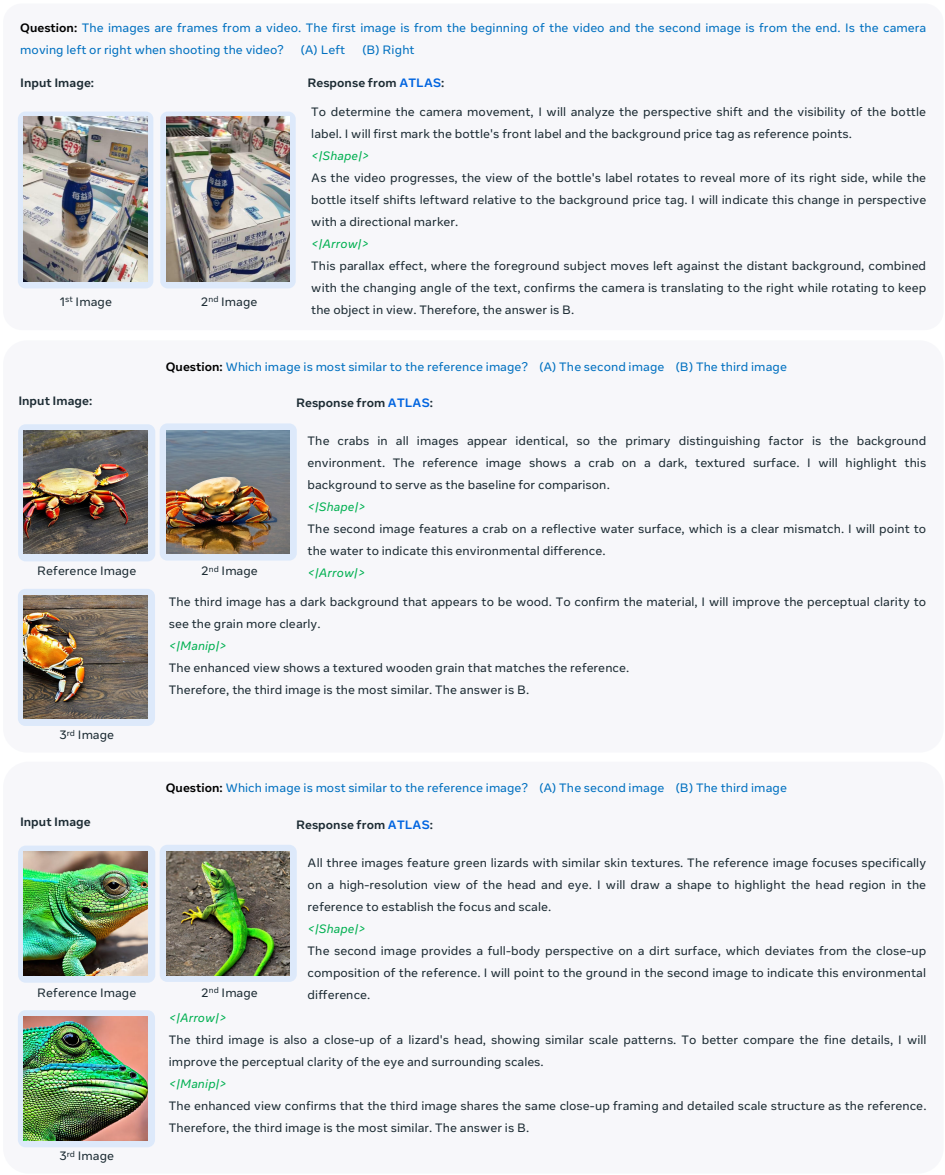


Figure 6: **Additional Qualitative Examples of ATLAS.** We show more visual reasoning examples where functional tokens help localize relevant regions, indicate directions, refine visual evidence, and support multi-step reasoning.

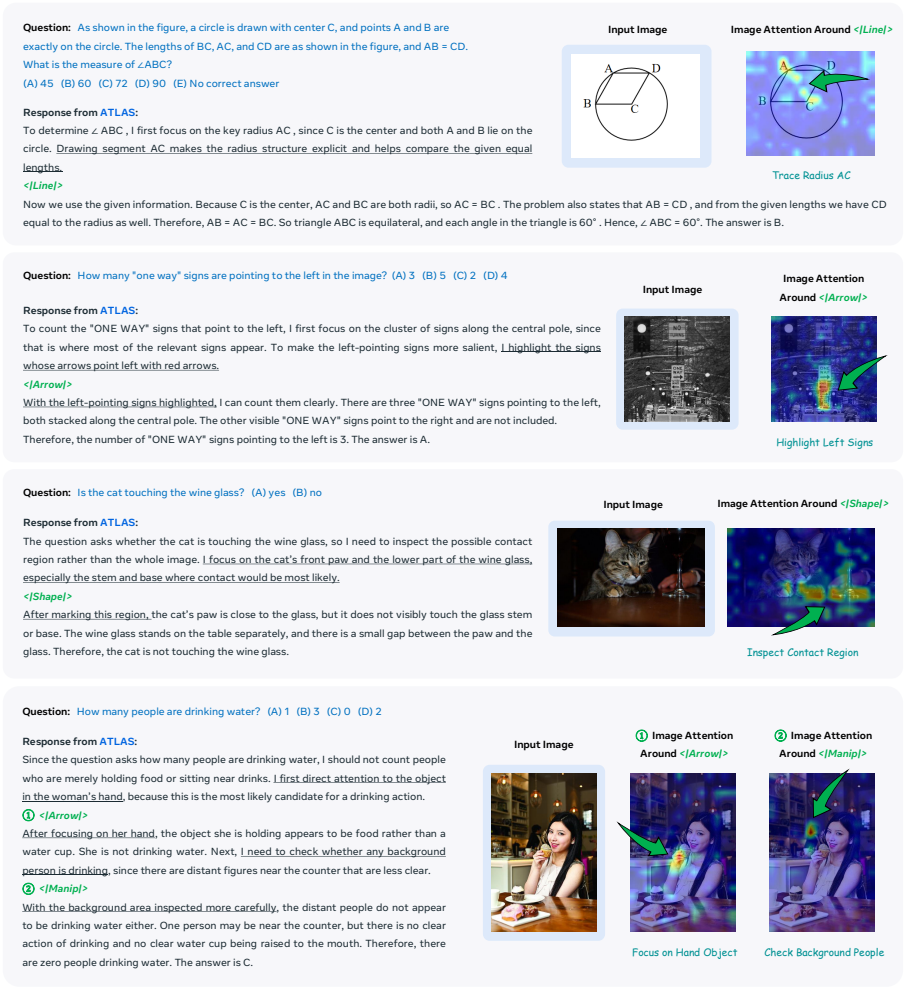


Figure 7: **Additional Attention Visualizations around Functional Tokens.** The attention maps show that functional tokens tend to focus on task-relevant visual evidence across different examples.

C Limitations

Although ATLAS improves visual reasoning efficiency and stability, it still has limitations. First, the functional-token set is compact and may not cover all possible visual operations. Extending the token taxonomy to more fine-grained operations could further improve flexibility. Second, current experiments are mainly conducted on a 7B-scale VLM, and larger-scale training may further verify the scalability.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The main conclusions and the scope of our paper are detailed in the introduction and summarized in the abstract.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The paper includes a dedicated Limitations section in Appendix C.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: All the assumptions and theorems are clearly stated.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The architecture, two-stage training recipe, and implementation details are provided in Section 2 and the Experiments section for reproducing the experimental results.

   Guidelines:

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

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

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

Answer: [No]

Justification: The code and data will be released at https://atlas-oneword.github.io.

Guidelines:

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

6. **Experimental setting/details**

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

Answer: [Yes]

Justification: The training datasets, evaluation benchmarks, and hyperparameters are provided in Section 2, the Experiments section, and Appendix A.

Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [No]

Justification: The paper reports benchmark results, ablations, and efficiency comparisons, but does not report error bars, confidence intervals, or statistical significance tests.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The compute resources (experiments conducted on 64 NVIDIA A100 GPUs) and inference latency/memory overhead are provided in the Experiments section and Appendix A.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: Our research conforms with the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: There is no societal impact of the work performed.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We cite the original papers or websites that produced the code package or dataset.

    Guidelines:

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

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: We detail the curated dataset (ATLAS-178K) and model training pipeline in Section 2 and the Experiments section.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [Yes]

Justification: The paper uses Gemini-2.5-Pro to polish reasoning trajectories during SFT data construction, and Qwen3-VL-235B-A22B-Instruct as an automated judge during evaluation.

Guidelines:

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