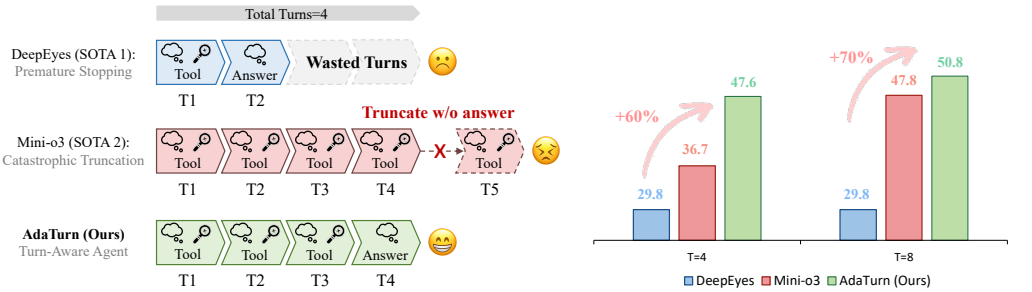
ADATURN: BUDGET-AWARE TEST-TIME SCALING FOR ACTIVE VISUAL PERCEPTION AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Active visual agents solve fine-grained image tasks by interleaving reasoning with image-grounding actions across multiple turns. However, deployment-time turn budgets are rarely fixed: some requests permit long rollouts, while others require the agent to act under a tight turn limit. Existing methods train the policy as if the turn budget were hidden, so when the available budget is smaller than the trajectory the agent prefers, the interaction is often truncated before any valid answer is produced; we term this failure *catastrophic truncation*. To overcome this challenge, we present AdaTurn, a budget-aware framework that conditions the agent on the allowed number of turns and explicitly trains the boundary behavior induced by the budget. Our key component, Forced-Answer DAPO (FA-DAPO), converts the over-budget event from a masked or penalized failure into a trainable final-decision step, teaching the model to synthesize partial evidence when further tool use is no longer possible. We further randomize turn budgets during both training and inference and introduce a load-balanced scheduler that makes such operations practical. AdaTurn substantially improves low-budget accuracy, for example raising VisualProbe-Medium from 36.7% to 47.6% at four turns, while preserving strong scaling at larger budgets and transferring effectively to multiple backbones and general multimodal benchmarks.



(a) Comparison between SoTA approaches and our AdaTurn.

(b) VisualProbe benchmark accuracy (Medium).

Figure 1: Turn-aware reasoning under limited turn budgets. Existing visual agents can stop too early when extra turns remain or suffer catastrophic truncation when the allocated budget is smaller than the rollout they implicitly expect, yielding no valid answer. AdaTurn conditions the agent on the turn budget and teaches it to switch from tool use to answer generation before the budget is exhausted, which leads to much stronger low-budget performance while preserving high-budget gains.

1 INTRODUCTION

Multimodal language models are increasingly used not only as one-shot image predictors (Liu et al., 2023; Guo et al., 2024; Bai et al., 2025b; Chen et al., 2024), but also as active visual agents that interleave reasoning with image-grounding actions across multiple turns (Wu & Xie, 2024; Wang et al., 2025b; Zhang et al., 2025b; Shen et al., 2025b; Zheng et al., 2026; Su et al., 2026; Lai et al.,

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.

2026). Instead of answering from a single encoded image, the model can inspect local regions, gather new observations, and refine its decision over multiple interaction turns.

This shift has made test-time scaling newly relevant for visual reasoning (Snell et al., 2024; OpenAI, 2024; Agarwal et al., 2025; Wang et al., 2025a). Recent systems such as DeepEyes (Zheng et al., 2026), Pixel Reasoner (Su et al., 2026), Chain-of-Focus (Zhang et al., 2025b), and Mini-o3 (Lai et al., 2026) show that "thinking with images" can outperform standard one-pass prediction on high-resolution and visually cluttered tasks. When the agent is allowed more turns, it can search more carefully, reject distractors, and verify a tentative answer before committing.

Yet this success hides an important deployment mismatch. In practical settings, the turn budget is an external resource constraint rather than a fixed property of the model (see Appendix G for a discussion of turn-level versus token-level budgets). One request may permit only four turns because of latency requirements, while another may permit thirty-two. Existing visual agents are typically trained for a fixed rollout regime rather than with the deployment-time budget as an explicit input (Zheng et al., 2026; Su et al., 2026; Lai et al., 2026). As a result, when the available budget is smaller than the policy prefers, the agent may continue requesting tools until the system terminates the rollout without any valid answer. We call this failure mode *catastrophic truncation*. For example, Mini-o3's performance on VisualProbe-Hard benchmark reduces from 48.0% to 26.5% after we limit the turn budget to 4. The same budget-blindness also wastes computation at the opposite extreme: when extra turns are available, the agent may stop as soon as it finds a plausible answer instead of using the remaining budget for verification.

AdaTurn addresses this mismatch by making the turn budget part of the learning problem. The central idea is simple: a capable visual agent should not learn a single rollout style, but a family of behaviors calibrated to the available budget. This perspective yields three design requirements. First, the policy must be conditioned on the turn budget, so that short-budget and long-budget behavior can differ systematically. Second, the model must receive learning signal at the budget boundary, because this is precisely where standard training breaks down. Third, if turn budgets vary during training and inference, the underlying infrastructure must also become budget-aware; otherwise the slowest worker dominates training throughput.

We satisfy these requirements with AdaTurn, an active visual perception agent with dynamic rollout depth. Besides the budget-conditioned visual agent, we also introduce **Forced-Answer DAPO (FA-DAPO)**, a reinforcement learning strategy built on DAPO (Yu et al., 2026), a more robust variant of GRPO (Shao et al., 2024), that converts an over-budget tool request into a trainable final-answer decision. Instead of penalizing or masking the boundary case, AdaTurn teaches the policy how to synthesize the evidence collected so far when no further tool use is possible. We further train with dynamic turn budgets and pair this with a load-balanced scheduler that keeps worker utilization stable despite heterogeneous rollout lengths.

The resulting system is effective across different turn budgets and benchmarks. At four turns, AdaTurn improves VisualProbe-Medium (Lai et al., 2026) from 36.7% to 47.6% and MME-RealWorld (Zhang et al., 2025c) from 50.1% to 64.0% over Mini-o3 (Lai et al., 2026). Figure 6 shows that these gains are concentrated where deployment constraints matter most, namely the low-budget regime, while the model remains strong as the budget increases. We also find that the approach transfers to Qwen3-VL 4B and 8B backbones (Bai et al., 2025a) and preserves broad multimodal ability on OCR and reasoning benchmarks.

In summary, our contributions are:

- We formulate budget-aware active visual perception and identify catastrophic truncation as a central failure mode of deployment-constrained visual agents.
- We propose AdaTurn, a budget-conditioned visual agent, and Forced-Answer DAPO (FA-DAPO), which explicitly trains the final-answer decision at the budget boundary.
- We introduce a load-balanced scheduler that reduces rollout imbalance during both training and inference and improves rollout throughput by $1.34\times$.
- We demonstrate that AdaTurn delivers large gains in the low-budget regime, remains strong at larger budgets, transfers to multiple backbones, and preserves general multimodal capability.

2 Related Work

2.1 High-Resolution Image Understanding

Early vision-language models such as LLaVA (Liu et al., 2023) use fixed low-resolution inputs and lose the fine details required by many visual tasks. High-resolution models, including LLaVA-UHD (Guo et al., 2024), Qwen2.5-VL (Bai et al., 2025b), and InternVL 2.5 (Chen et al., 2024), improve image encoding through dynamic tiling, image slicing, or stronger visual backbones, but still perceive the image in a single pass. Active perception methods, such as SEAL (Wu & Xie, 2024), DC^2 (Wang et al., 2025b), Chain-of-Focus (Zhang et al., 2025b), and ZoomEye (Shen et al., 2025b), instead inspect the image over multiple steps. AdaTurn follows this paradigm but studies how such agents should behave when the turn budget varies across requests.

2.2 Agentic Visual Perception

Recent work learns multi-turn visual search within a language-model action loop. DeepEyes (Zheng et al., 2026) shows that image-grounded tool use can emerge from reinforcement learning alone, and DeepEyesV2 (Hong et al., 2026), VISTA-R1 (Lu et al., 2026), and Agent0-VL (Liu et al., 2025a) extend the tool space or the training environment. Mini-o3 (Lai et al., 2026) further shows that visual agents benefit from longer rollouts and richer interaction patterns. While these methods aim to obtain longer and more effective trajectories, AdaTurn asks how an agent should behave when the permitted trajectory length changes at deployment, and directly optimizes its decision at the last turn.

2.3 Reinforcement Learning and Test-Time Scaling for VLMs

Group Relative Policy Optimization (GRPO) (Shao et al., 2024) has become a standard recipe for training reasoning models, from DeepSeek-R1 (Guo et al., 2025) to multimodal models such as VLM-R1 (Shen et al., 2025a), R1-VL (Zhang et al., 2025a), and Visionary-R1 (Xia et al., 2025), and recent work extends it to multi-turn tool-integrated reasoning (Xue et al., 2026; Zheng et al., 2025). Test-time scaling shows that additional inference-time computation improves reasoning when the model uses it effectively (Snell et al., 2024; OpenAI, 2024; Agarwal et al., 2025; Wang et al., 2025a). Existing budget-aware methods mainly control the token budget of a single generation through budget tokens or confidence-based early stopping (Wen et al., 2025; Nogueira et al., 2025). Closest to our setting, Liu et al. (2025b) propose a training-free Budget Tracker that informs web search agents of their remaining budget through the prompt, which we find insufficient for 7B visual agents (Appendix F). AdaTurn instead learns turn-level budget awareness through reinforcement learning and provides the systems support to train it efficiently.

3 Method

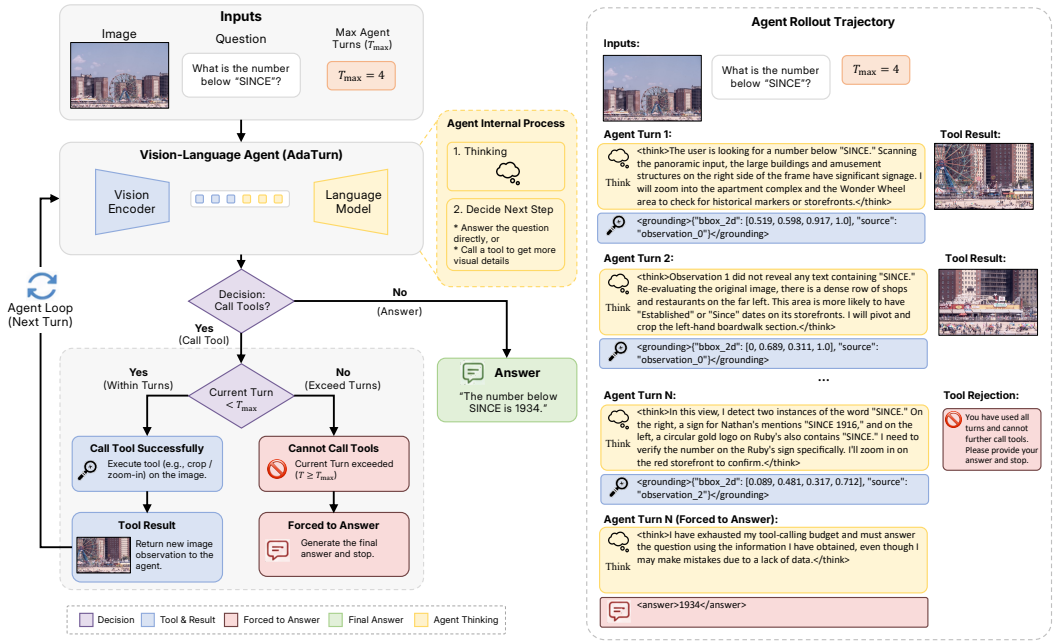
AdaTurn is built around a simple premise: for an active visual agent, the turn budget is not a nuisance variable but part of the task specification. The policy should therefore depend not only on the image and the question, but also on how many interaction turns are available. We first formalize this budget-aware setting in Section 3.1. We then present Forced-Answer DAPO (FA-DAPO) in Section 3.2, which supplies the missing learning signal at the final-turn boundary. Finally, Section 3.3 describes dynamic-budget training and the load-balanced rollout scheduler that makes such training efficient in practice. Figure 2 gives an overview.

3.1 Problem Formulation and AdaTurn Agent Loop

Given a question Q about an image I and a turn budget $T_{\max}$, AdaTurn answers Q through a multi-turn interaction between a vision-language model and the image. The interaction produces an interleaved sequence of inputs, model outputs, and tool observations:

$$(Q, I, T_{\max}, (y_1, o_1), \ldots, (y_{T_{\max}-1}, o_{T_{\max}-1}), y_{T_{\max}}), \tag{1}$$

where y_t is the token sequence generated by the model at turn t and o_t is the observation returned by the tool for y_t. At each turn, the policy π_θ generates $y_t \sim \pi_\theta(\cdot \mid h_t)$, where the history h_t contains all elements of Eq. 1 that precede y_t. Each output y_t first reasons about the evidence collected so



(a) AdaTurn Agent Loop.

(b) AdaTurn Agent Rollout Trajectory.

Figure 2: Overview of AdaTurn. (a) The agent loop. The model receives the image, the question, and the turn budget $T_{\max}$, then decides at each turn whether to call the zoom-in tool or to answer the question. If a tool call would exceed the budget, AdaTurn rejects it and forces the model to answer using the evidence already collected. (b) An example rollout in which the agent gathers several observations and then answers at the last turn.

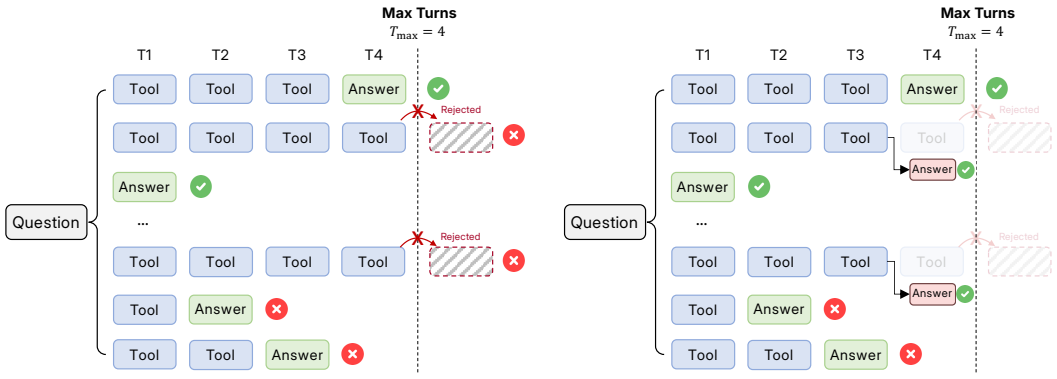
far and then takes one of two actions. It either answers the question, denoted $y_t \in \mathcal{A}$, which ends the interaction, or calls an image zoom-in tool, denoted $y_t \in \mathcal{C}$, which crops a region from the input image or from an earlier observation and returns it as o_t. The interaction therefore ends before turn $T_{\max}$ whenever the model answers early.

The turn budget restricts which action is admissible at each turn:

$$y_t \in \begin{cases} \mathcal{C} \cup \mathcal{A}, & t < T_{\max}, \\ \mathcal{A}, & t = T_{\max}. \end{cases} \tag{2}$$

Before the last turn, both actions are accepted, so the model may either gather more evidence or answer. At the last turn, the model is expected to answer, because no later turn remains to use a new observation. In practice, however, a vision-language model may not follow this requirement. If it still calls the tool at turn $T_{\max}$, the call cannot be executed and the interaction ends without a valid answer, which is the catastrophic truncation discussed in Section 1. To resolve this issue, we propose Forced-Answer DAPO (Section 3.2), which trains the model to answer at the last turn using the collected evidence. After training, AdaTurn adapts to the given budget: under a small $T_{\max}$, it compresses its search and commits to an answer based on partial evidence, while under a large $T_{\max}$, it spends the extra turns verifying a tentative answer.

Figure 2 illustrates the agent loop with an example. As shown in Figure 2(a), the model decides at each turn whether to answer or to call the tool. A tool call is executed only when the current turn is smaller than $T_{\max}$; otherwise, it is rejected and the model is forced to answer, as detailed in Section 3.2. Figure 2(b) shows a rollout with $T_{\max} = 4$ for the question "What is the number below 'SINCE'?". The agent first inspects the right side of the image and finds no relevant text. It then moves to the storefronts on the left and detects two signs containing "SINCE". At the last turn, it requests a further zoom-in on one of the signs, but this call exceeds the budget and is rejected, so the agent answers "1934" based on the observations it has already collected.



(a) Existing vision agent training approach (e.g., DeepEyes, Mini-o3).

(b) Our new agent training approach.

Figure 3: Training behavior at the budget boundary. Prior methods either treat over-budget rollouts as wrong and penalize them, or mask them and discard the learning signal. Forced-Answer DAPO instead discards the over-budget tool call, prompts the model to answer with the evidence collected so far, and trains on the resulting rollout.

3.2 Forced-Answer DAPO

DAPO. Before introducing FA-DAPO, we briefly review the reinforcement learning algorithm it builds on, Decoupled Clip and Dynamic Sampling Policy Optimization (DAPO) (Yu et al., 2026), a more robust variant of GRPO (Shao et al., 2024; Guo et al., 2025; Shen et al., 2025a; Zhang et al., 2025a). Given a training sample $(Q, I, T_{\max})$, DAPO samples a group of G multi-turn rollout trajectories $\{Y^{(i)}\}_{i=1}^{G}$ from the old policy $\pi_{\theta_{\text{old}}}$. Here, $Y^{(i)} = [y_1^{(i)}, \ldots, y_{T_{\max}}^{(i)}]$ is the i-th rollout in the group, and $y_t^{(i)}$ is the token sequence generated at turn t of this rollout. A rollout contains fewer than $T_{\max}$ turns if the model answers early. Each rollout receives a scalar reward $R^{(i)}$, and all of its tokens share the group-normalized advantage

$$\hat{A}^{(i)} = \frac{R^{(i)} - \frac{1}{G}\sum_{j=1}^{G} R^{(j)}}{\text{Std}(\{R^{(j)}\}_{j=1}^{G}) + \delta}, \tag{3}$$

where δ is a small constant for numerical stability. The reward combines accuracy and format:

$$R^{(i)} = \lambda_{\text{acc}}\, r_{\text{acc}}(Y^{(i)}) + \lambda_{\text{fmt}}\, r_{\text{fmt}}(Y^{(i)}), \tag{4}$$

where r_{acc} measures whether the final answer in $Y^{(i)}$ matches the ground truth, r_{fmt} measures whether $Y^{(i)}$ follows the required format, and λ_{acc} and λ_{fmt} are balancing weights. To define the token-level objective, we view $Y^{(i)}$ as the concatenation of its generated tokens, and denote its k-th token by $Y_k^{(i)}$ and its length by $|Y^{(i)}|$. DAPO minimizes the token-normalized clipped objective

$$\mathcal{L}_{\text{DAPO}}(\theta) = -\mathbb{E}\left[\frac{1}{\sum_{i=1}^{G}|Y^{(i)}|}\sum_{i=1}^{G}\sum_{k=1}^{|Y^{(i)}|}\min\left(\rho_k^{(i)}\hat{A}^{(i)},\ \text{clip}\left(\rho_k^{(i)}, 1-\epsilon_{\text{low}}, 1+\epsilon_{\text{high}}\right)\hat{A}^{(i)}\right)\right], \tag{5}$$

where ϵ_{low} and ϵ_{high} are the lower and upper clipping ranges, the expectation is over training samples and their groups, and $\rho_k^{(i)}$ is the token-level importance ratio:

$$\rho_k^{(i)} = \frac{\pi_\theta\left(Y_k^{(i)} \mid Y_{<k}^{(i)}\right)}{\pi_{\theta_{\text{old}}}\left(Y_k^{(i)} \mid Y_{<k}^{(i)}\right)}. \tag{6}$$

For brevity, we omit from the conditioning the inputs $(Q, I, T_{\max})$ and the tool observations that precede $Y_k^{(i)}$. Tool observations are returned by the environment rather than generated by the policy, so they are not part of $Y^{(i)}$ and do not contribute to the loss.

FA-DAPO. Having reviewed DAPO, we now introduce Forced-Answer DAPO (FA-DAPO), which adapts DAPO to the turn budget. As discussed in Sections 1 and 3.1, catastrophic truncation occurs at turn $t = T_{\max}$. Instead of answering the question, the model may still call the tool, although no tool call is allowed at this turn (Eq. 2). The rollout is then truncated without any valid answer. Existing agents handle such truncated rollouts in two ways, as illustrated in Figure 3(a). DeepEyes (Zheng et al., 2026) treats them as wrong rollouts, which penalizes long-horizon exploration and ties the policy to the training budget. Mini-o3 (Lai et al., 2026) instead masks them out of the loss, which avoids the penalty but also removes the learning signal exactly where budget awareness is needed. Both strategies are suboptimal, because neither teaches the model what to do when the budget runs out, so its behavior at the last turn remains untrained.

Instead of penalizing or masking truncated rollouts, FA-DAPO explicitly trains the model to answer at the last turn. When a rollout calls the tool at turn $T_{\max}$, we discard this tool call and roll back to the history before the last turn,

$$h^{(i)}_{T_{\max}} = \left(Q, I, T_{\max}, (y^{(i)}_1, o^{(i)}_1), \ldots, (y^{(i)}_{T_{\max}-1}, o^{(i)}_{T_{\max}-1})\right),$$

which contains the inputs together with all model outputs and tool observations of the first $T_{\max} - 1$ turns. We then append a forced-answer prompt b to this history and let the model regenerate the last turn. The prompt b states that the turn budget has been exhausted and that the model must answer with the evidence collected so far (Appendix I). Formally, we replace each sampled rollout $Y^{(i)}$ with

$$\tilde{Y}^{(i)} = \begin{cases} \left[y^{(i)}_1, \ldots, y^{(i)}_{T_{\max}-1}, \tilde{y}^{(i)}_{T_{\max}}\right], & \text{if } y^{(i)}_{T_{\max}} \in \mathcal{C}, \\ Y^{(i)}, & \text{otherwise}, \end{cases} \tag{7}$$

where $\tilde{y}^{(i)}_{T_{\max}} \sim \pi_{\theta_{\text{old}}}(\cdot \mid h^{(i)}_{T_{\max}}, b)$ is the regenerated last turn. In other words, a truncated rollout keeps its first $T_{\max} - 1$ turns and replaces its last tool call with $\tilde{y}^{(i)}_{T_{\max}}$, while all other rollouts remain unchanged. We then compute the reward (Eq. 4), the advantage (Eq. 3), and the loss (Eq. 5) on $\tilde{Y}^{(i)}$ in place of $Y^{(i)}$. Like the tool observations, the prompt b is excluded from the loss. If the model ignores b and calls the tool again, $\tilde{y}^{(i)}_{T_{\max}}$ contains no answer and receives zero accuracy reward. Figure 3(b) contrasts FA-DAPO with the existing strategies in Figure 3(a): rollouts that would otherwise be penalized or masked now end with a forced answer and can receive a positive reward when this answer is correct.

Since all turns of $\tilde{Y}^{(i)}$ share one advantage, FA-DAPO also rewards the earlier tool calls that collected sufficient evidence for a correct forced answer. The same forced-answer step is applied at inference, so AdaTurn always answers within $T_{\max}$ turns. Appendix C compares FA-DAPO with alternative training variants, including masking prefix turns, training only the forced-answer turn, and adding a format reward.

3.3 Dynamic Turn Budgets and Load-Balanced Scheduling

The turn budget is not fixed during either training or inference. During training, we sample $T_{\max}$ from a predefined set of candidate budgets and inject it into the prompt as part of the agent state. The same task therefore appears under both short and long budgets, forcing the policy to learn a calibrated trade-off between further search and immediate commitment. This same mechanism carries over directly to inference: practitioners can set $T_{\max}$ according to their latency or compute budget without switching models or retraining the policy.

Dynamic budgets, however, create a systems bottleneck. In multi-engine rollout generation, batch latency is determined by the slowest engine. If long-budget samples are assigned unevenly, fast workers idle while one overloaded worker processes a set of overlong trajectories. Figure 4 shows our solution. We assign rollout jobs by sorting requests according to their assigned budget and greedily dispatching each request to the engine with the current lowest accumulated load, which keeps the rollout cost more balanced across workers. Appendix B provides the full pseudocode. Each request r_n carries an assigned turn budget T_n. After sorting requests by T_n in descending order, we greedily assign each request to the engine with the current lowest accumulated load. We quantify rollout imbalance as the gap between the largest and smallest accumulated turn budgets across engines after assignment.

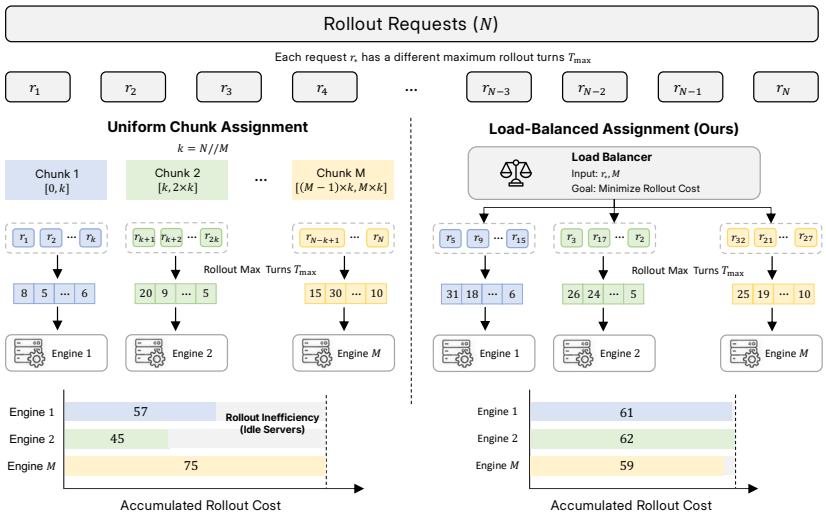


Figure 4: Load-balanced rollout assignment. Dynamic turn budgets create large differences in per-sample rollout cost. Instead of uniformly splitting requests across engines, we sort requests by assigned budget and greedily place each one on the engine with the current lowest accumulated load, which keeps the rollout workers much better balanced.

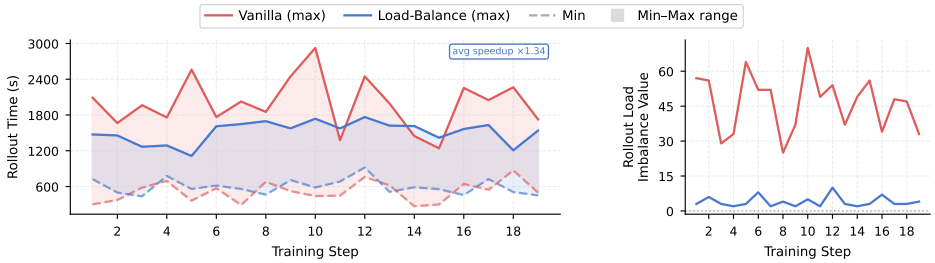


Figure 5: Effect of load-balanced rollout scheduling. The left plot reports the minimum, maximum, and min-max rollout time range across engines, showing that the scheduler reduces maximum engine time and yields an average $1.34\times$ speedup. The right plot reports the gap between the largest and smallest accumulated turn budgets across engines, which is consistently smaller with balancing.

Figure 5 reports the effect on real training runs. The scheduler consistently reduces both the maximum rollout time and the cross-engine load imbalance value calculated as the gap between the largest and smallest accumulated turn budgets across engines. This leads to an average $1.34\times$ speedup in rollout generation, which is critical for dynamic-budget training and inference.

4 Experiments

4.1 Experimental Details

Datasets. We evaluate AdaTurn on VisualProbe (Lai et al., 2026) and V* Bench (Wu & Xie, 2024) for agentic visual search, HR-Bench (Wang et al., 2025b) at 4K and 8K for high-resolution perception, and MME-RealWorld (Zhang et al., 2025c) for real-world multimodal reasoning. To assess whether the resulting policy remains broadly useful beyond the main visual-search setting, we additionally report OCRBench (Liu et al., 2024), ChartQA (Masry et al., 2022), DocVQA (Mathew et al., 2021), MathVista (Lu et al., 2023), ScienceQA (Lu et al., 2022), and CV-Bench (Tong et al., 2024).

Baselines. We compare against closed and open-source multimodal models, including GPT-4o (Hurst et al., 2024), LLaVA-OneVision (Li et al., 2024), Qwen2.5-VL-7B-Instruct (Bai et al., 2025b), DeepEyes (Zheng et al., 2026), Mini-o3 (Lai et al., 2026), Pixel Reasoner (Su et al., 2026), and

Table 1: Quantitative comparison on visual perception benchmarks. AdaTurn performs favorably in the low-budget regime and remains competitive at 32 turns while using the same 7B backbone scale as prior open-source agent baselines. For VisualProbe and V* Bench, we report Avg@32 to reduce variance caused by randomness. We report Avg@8 and Avg@1 for HR-Bench and MME-RealWorld, respectively.

Model	VisualProbe			V*	HR-Bench		MME-RealWorld
	hard	medium	easy		4K	8K	
GPT-4o (Hurst et al., 2024)	11.2	15.4	47.5	65.2	62.0	58.3	45.2
LLaVA-OneVision (Li et al., 2024)	13.4	12.5	36.2	70.9	61.2	54.0	57.4
Qwen2.5-VL-Instruct (Bai et al., 2025b)	23.9	26.0	39.1	75.5	68.2	62.7	57.3
SEAL (Wu & Xie, 2024)	-	-	-	75.4	-	-	-
DyFo (Li et al., 2025)	-	-	-	81.2	-	-	-
Chain-of-Focus (Zhang et al., 2025b)	-	-	-	88.0	-	-	-
Pixel Reasoner (Su et al., 2026)	28.8	29.6	58.4	86.3	74.0	66.9	64.4
DeepEyes (Zheng et al., 2026) (6 turns)	35.1	29.8	60.1	83.3	73.2	69.5	64.0
Mini-o3 (Lai et al., 2026) (4 turns)	26.5	36.7	42.4	79.2	68.9	63.2	50.1
Mini-o3 (Lai et al., 2026) (32 turns)	48.0	50.4	**67.0**	**88.2**	77.5	73.3	65.5
Ours (4 turns)	**39.0**	**47.6**	**61.2**	**85.2**	**75.7**	**71.1**	**64.0**
Ours (32 turns)	**48.1**	**51.0**	66.7	86.9	**77.5**	**73.4**	**66.3**

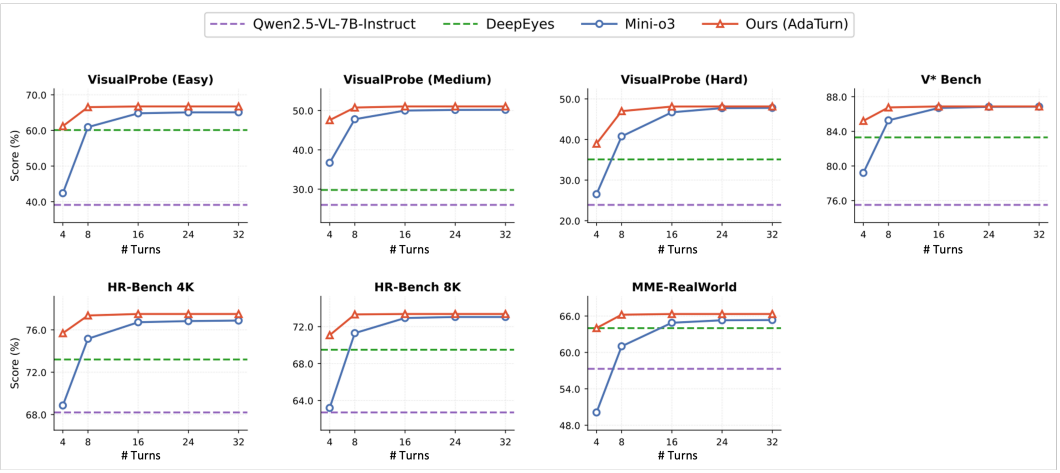


Figure 6: Performance vs. turn budget. AdaTurn provides a more favorable trade-off than Mini-o3 (Lai et al., 2026) in the low-budget regime, particularly at four and eight turns, while remaining competitive as additional turns are available.

Chain-of-Focus (Zhang et al., 2025b). The most relevant baselines are DeepEyes (Zheng et al., 2026) and Mini-o3 (Lai et al., 2026) because they also train multi-turn visual tool use; they therefore isolate the contribution of budget-aware learning more directly than generic VLM baselines.

Metrics. We report benchmark-standard accuracy. Following the corresponding benchmark protocols, VisualProbe and V* Bench are averaged over 32 repeated runs to reduce stochastic variation, while HR-Bench and MME-RealWorld are evaluated with 8 and 1 run, respectively.

Implementation Details. Unless otherwise specified, the main results use a Qwen2.5-VL-7B-Instruct backbone (Bai et al., 2025b) and compare inference under different turn budgets. For more implementation details, including the training hyperparameters and the ablation variants, please refer to Appendix A.

4.2 Quantitative Comparison

Table 1 reports the main quantitative comparison. AdaTurn performs particularly well in the low-budget regime, where catastrophic truncation is most consequential. At four turns, AdaTurn improves

Table 2: Transfer across backbone scales. AdaTurn improves both Qwen3-VL-4B-Instruct and Qwen3-VL-8B-Instruct under tight budgets and remains competitive when the turn budget is increased. For VisualProbe and V* Bench, we report Avg@32 to reduce variance caused by randomness. We report Avg@8 and Avg@1 for HR-Bench and MME-RealWorld, respectively.

Model	VisualProbe			V*	HR-Bench		MME-RealWorld
	hard	medium	easy		4K	8K	
Qwen3-VL-4B-Instruct (Bai et al., 2025a)	-	-	-	88.0	81.3	74.4	-
+ Mini-o3 (Lai et al., 2026) (4 turns)	24.7	44.3	54.9	87.0	77.2	72.4	61.2
+ Mini-o3 (Lai et al., 2026) (32 turns)	46.3	53.9	69.1	90.2	**80.0**	77.3	66.0
+ Ours (4 turns)	**40.0**	**51.7**	**68.4**	**90.0**	**79.2**	**76.6**	**66.6**
+ Ours (32 turns)	**46.4**	**54.3**	**70.6**	**90.7**	79.8	**78.0**	**67.1**
Qwen3-VL-8B-Instruct (Bai et al., 2025a)	-	-	-	90.1	82.3	78.0	-
+ Mini-o3 (Lai et al., 2026) (4 turns)	24.9	44.6	58.7	87.6	78.3	75.8	60.0
+ Mini-o3 (Lai et al., 2026) (32 turns)	55.4	55.6	74.7	91.1	82.8	81.9	69.3
+ Ours (4 turns)	**45.9**	**54.6**	**73.8**	**91.1**	**82.4**	**81.4**	**69.2**
+ Ours (32 turns)	**56.0**	**57.0**	**74.8**	**91.6**	**83.0**	**82.3**	**69.8**

over Mini-o3 (Lai et al., 2026) by $+12.5\%$ on VisualProbe-Hard, $+10.9\%$ on VisualProbe-Medium, $+6.0\%$ on V*, $+6.8\%$ on HR-Bench 4K, $+7.9\%$ on HR-Bench 8K, and $+13.9\%$ on MME-RealWorld. These improvements are consistent across VisualProbe (Lai et al., 2026), HR-Bench (Wang et al., 2025b), and MME-RealWorld (Zhang et al., 2025c), suggesting that the benefit is not confined to a single benchmark family.

At the same time, these gains do not come at the expense of high-budget behavior. At thirty-two turns, AdaTurn matches or exceeds Mini-o3 (Lai et al., 2026) on VisualProbe-Hard, VisualProbe-Medium, HR-Bench 8K, and MME-RealWorld, while tying on HR-Bench 4K. This indicates that FA-DAPO does not simply encourage earlier termination; rather, it teaches the model to answer appropriately when the budget is limited while still making use of longer rollouts when they are available.

Figure 6 presents the same trend from a complementary perspective by plotting performance as a function of the turn budget. AdaTurn shifts the curve upward in the low-budget region, especially at 4 and 8 turns, while preserving favorable scaling as the budget grows. Ablation studies of the FA-DAPO design and an analysis of the behaviors that reinforcement learning instills in AdaTurn are provided in Appendices C and D, while additional rollout visualizations and a failure case analysis are provided in Appendices J and K.

4.3 Applicability

Table 2 shows that AdaTurn transfers cleanly to other backbone scales. On both Qwen3-VL-4B-Instruct and Qwen3-VL-8B-Instruct (Bai et al., 2025a), AdaTurn outperforms Mini-o3 (Lai et al., 2026) at four turns across nearly all visual-search benchmarks, and these improvements largely persist at thirty-two turns. This suggests that AdaTurn is not tied to a single base model, but reflects a more general training principle for visual agents.

General multimodal results are deferred to Appendix E. They show that AdaTurn preserves broad OCR and reasoning ability relative to the underlying Qwen2.5-VL-7B-Instruct backbone (Bai et al., 2025b) while remaining competitive with Mini-o3 (Lai et al., 2026).

5 Conclusion

We presented AdaTurn, a budget-aware framework for active visual agents under constrained test-time interaction. AdaTurn treats the turn budget as part of the problem specification, introduces Forced-Answer DAPO to optimize the final-turn boundary explicitly, and combines dynamic-budget training with load-balanced rollout scheduling. Together, these components address catastrophic truncation while preserving the gains of longer rollouts. Empirically, AdaTurn performs favorably in the low-budget regime, remains competitive at larger budgets, transfers to multiple backbone sizes, and preserves broad multimodal capability.

AI USE STATEMENT

In this work, we used generative AI tools solely to fix English grammatical errors and polish the phrasing of sentences already drafted by the authors. We have not used generative AI tools for any other tasks with required or recommended disclosure, including but not limited to generating synthetic data, developing theoretical models or hypotheses, formulating mathematical claims or proofs, designing methodology or experiments, implementing methods, analyzing or interpreting results, creating or editing figures, drafting original text, summarizing or analyzing literature, or suggesting paper structure, titles, or keywords; these tasks are not applicable to this work. All AI-assisted grammar and wording suggestions were reviewed by the authors, who verified that the edits preserved the intended technical meaning before accepting them. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Reyna Abhyankar, Qi Qi, and Yiying Zhang. Osworld-human: Benchmarking the efficiency of computer-use agents. *Proceedings of Machine Learning and Systems*, 8:482–494, 2026.

Aradhye Agarwal, Ayan Sengupta, and Tanmoy Chakraborty. The art of scaling test-time compute for large language models. *arXiv preprint arXiv:2512.02008*, 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.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024.

Pengfei Gao and Chao Peng. More with less: An empirical study of turn-control strategies for efficient coding agents. In *Proceedings of the 2026 IEEE/ACM 48th International Conference on Software Engineering*, pp. 956–967, 2026.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Zonghao Guo, Ruyi Xu, Yuan Yao, Junbo Cui, Zanlin Ni, Chunjiang Ge, Tat-Seng Chua, Zhiyuan Liu, and Gao Huang. Llava-uhd: an lmm perceiving any aspect ratio and high-resolution images. In *European Conference on Computer Vision*, pp. 390–406. Springer, 2024.

Jack Hong, Chenxiao Zhao, ChengLin Zhu, Weiheng Lu, and Guohai Xu. Deepeyesv2: Toward agentic multimodal model. In *International Conference on Learning Representations*, volume 2026, pp. 114851–114872, 2026.

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

Xin Lai, Junyi Li, Wei Li, Tao Liu, Tianjian Li, and Hengshuang Zhao. Mini-o3: Scaling up reasoning patterns and interaction turns for visual search. In *International Conference on Learning Representations*, volume 2026, pp. 76722–76746, 2026.

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.

Geng Li, Jinglin Xu, Yunzhen Zhao, and Yuxin Peng. Dyfo: A training-free dynamic focus visual search for enhancing lmms in fine-grained visual understanding. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9098–9108. IEEE, 2025.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

Jiaqi Liu, Kaiwen Xiong, Peng Xia, Yiyang Zhou, Haonian Ji, Lu Feng, Siwei Han, Mingyu Ding, and Huaxiu Yao. Agent0-vl: Exploring self-evolving agent for tool-integrated vision-language reasoning. *arXiv preprint arXiv:2511.19900*, 2025a.

Tengxiao Liu, Zifeng Wang, Jin Miao, I Hsu, Jun Yan, Jiefeng Chen, Rujun Han, Fangyuan Xu, Yanfei Chen, Ke Jiang, et al. Budget-aware tool-use enables effective agent scaling. *arXiv preprint arXiv:2511.17006*, 2025b.

Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xu-Cheng Yin, Cheng-Lin Liu, Lianwen Jin, and Xiang Bai. Ocrbench: on the hidden mystery of ocr in large multimodal models. *Science China Information Sciences*, 67(12):220102, 2024.

Meng Lu, Ran Xu, Yi Fang, Wenxuan Zhang, Yue Yu, Gaurav Srivastava, Yuchen Zhuang, Mohamed Elhoseiny, Charles Fleming, Carl Yang, et al. Scaling agentic reinforcement learning for tool-integrated reasoning in vlms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26518–26529, 2026.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in neural information processing systems*, 35:2507–2521, 2022.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating math reasoning in visual contexts with gpt-4v, bard, and other large multimodal models. *arXiv preprint arXiv:2310.02255*, 2023.

Ahmed Masry, Jia Qing Tan, Shafiq Joty, Enamul Hoque, et al. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the association for computational linguistics: ACL 2022*, pp. 2263–2279, 2022.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 2199–2208. IEEE, 2021.

João Paulo Nogueira, Wentao Sun, Alonso Silva, and Laith Zumot. Certainty-guided reasoning in large language models: A dynamic thinking budget approach. *arXiv preprint arXiv:2509.07820*, 2025.

OpenAI. Learning to reason with LLMs. *OpenAI Blog*, 2024. URL `https://openai.com/index/learning-to-reason-with-llms/`.

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.

Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, et al. Vlm-r1: A stable and generalizable r1-style large vision-language model. *arXiv preprint arXiv:2504.07615*, 2025a.

Haozhan Shen, Kangjia Zhao, Tiancheng Zhao, Ruochen Xu, Zilun Zhang, Mingwei Zhu, and Jianwei Yin. Zoomeye: Enhancing multimodal llms with human-like zooming capabilities through tree-based image exploration. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 6613–6629, 2025b.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, 2025.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

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.

Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai C Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *Advances in Neural Information Processing Systems*, 37: 87310–87356, 2024.

Jian Wang, Boyan Zhu, Chak Tou Leong, Yongqi Li, and Wenjie Li. Scaling over scaling: Exploring test-time scaling plateau in large reasoning models. *arXiv preprint arXiv:2505.20522*, 2025a.

Wenbin Wang, Liang Ding, Minyan Zeng, Xiabin Zhou, Li Shen, Yong Luo, Wei Yu, and Dacheng Tao. Divide, conquer and combine: A training-free framework for high-resolution image perception in multimodal large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 7907–7915, 2025b.

Hao Wen, Xinrui Wu, Yi Sun, Feifei Zhang, Liye Chen, Jie Wang, Yunxin Liu, Yunhao Liu, Ya-Qin Zhang, and Yuanchun Li. Budgetthinker: Empowering budget-aware llm reasoning with control tokens. *arXiv preprint arXiv:2508.17196*, 2025.

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)*, pp. 13084–13094. IEEE, 2024.

Jiaer Xia, Yuhang Zang, Peng Gao, Sharon Li, and Kaiyang Zhou. Visionary-r1: Mitigating shortcuts in visual reasoning with reinforcement learning. *arXiv preprint arXiv:2505.14677*, 2025.

Zhenghai Xue, Longtao Zheng, Qian Liu, Yingru Li, Xiaosen Zheng, Zejun Ma, and Bo An. Simpletir: End-to-end reinforcement learning for multi-turn tool-integrated reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 8424–8449, 2026.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026.

Jingyi Zhang, Jiaxing Huang, Huanjin Yao, Shunyu Liu, Xikun Zhang, Shijian Lu, and Dacheng Tao. R1-vl: Learning to reason with multimodal large language models via step-wise group relative policy optimization. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1859–1869. IEEE, 2025a.

Xintong Zhang, Zhi Gao, Bofei Zhang, Pengxiang Li, Xiaowen Zhang, Yang Liu, Tao Yuan, Yuwei Wu, Yunde Jia, Song-Chun Zhu, et al. Adaptive chain-of-focus reasoning via dynamic visual search and zooming for efficient vlms. *arXiv preprint arXiv:2505.15436*, 2025b.

YiFan Zhang, Huanyu Zhang, Haochen Tian, Chaoyou Fu, Shuangqing Zhang, Junfei Wu, Feng Li, Kun Wang, Qingsong Wen, Zhang Zhang, et al. Mme-realworld: Could your multimodal llm challenge high-resolution real-world scenarios that are difficult for humans? In *International Conference on Learning Representations*, volume 2025, pp. 89655–89701, 2025c.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group sequence policy optimization, 2025. URL https://arxiv.org/abs/2507.18071.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyan Luo. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 3: system demonstrations)*, pp. 400–410, 2024.

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, pp. 126775–126798, 2026.

A Detailed Training Setup

Following Mini-o3 (Lai et al., 2026), we first initialize AdaTurn by supervised fine-tuning Qwen2.5-VL-7B-Instruct (Bai et al., 2025b) on the Mini-o3 cold-start dataset[1]. We then perform reinforcement learning on DeepEyes_train_4K[2] and VisualProbe_train[3] with dynamic turn budgets from 4 to 12 turns. We use LLaMA-Factory (Zheng et al., 2024) for supervised fine-tuning and VERL (Sheng et al., 2025) for reinforcement learning, and we follow the released licenses and terms of use of the frameworks, datasets, and pretrained backbones used in our experiments.

Table 3: Training configuration used in our experiments.

Configuration	Supervised fine-tuning	Reinforcement learning
Framework	LLaMA-Factory	VERL
Training data	Mini-o3 cold-start dataset	DeepEyes_train_4K + VisualProbe_train
Batch size	64	64
Group size	–	16
Mini-batch size	–	32
Learning rate	1×10^{-5}	1×10^{-6}
Schedule	10% warmup + cosine decay	none
Epochs	3	1
KL loss	–	none
Reward weights	–	accuracy 1.0, format 0.0
Minimum image pixels	40,000	40,000
Maximum image pixels	2,000,000	2,000,000
Dynamic turn budgets	–	4 to 12
Freeze vision encoder	true	false
Freeze projection layer	true	false
Freeze LLM	false	false
Context length	32,768	32,768
Compute	1 node with 8 A100 GPUs	2 nodes with 8 A100 GPUs per node
Wall-clock time	about 2 hours	about 3 days

B Load-Balanced Rollout Assignment Pseudocode

Algorithm 1 gives the full scheduling routine used by AdaTurn when dynamic turn budgets are enabled. The key idea is to sort requests by assigned turn budget and greedily place each request on the rollout engine with the smallest current accumulated load. This simple strategy reduces the idle time induced by heterogeneous rollout lengths and complements the main-text discussion in Section 3.3.

C Ablation Studies

Figure 7 isolates the contribution of the reinforcement learning design. The main takeaway is that forced-answer training is the primary contributor to the improvement: every AdaTurn variant outperforms Mini-o3 (Lai et al., 2026) at four turns, indicating that explicit supervision of the budget boundary is important for mitigating catastrophic truncation. Among these variants, the default formulation performs best overall, especially once the budget exceeds eight turns.

Masking earlier prefix turns degrades performance, which suggests that the final forced-answer decision is best learned together with the preceding search trajectory rather than in isolation. Restricting reinforcement learning to only the final forced-answer turn performs worse still, indicating that budget-aware behavior is a trajectory-level property rather than a single-step correction.

[1] https://huggingface.co/datasets/Mini-o3/Mini-o3-Coldstart-Dataset
[2] https://huggingface.co/datasets/Mini-o3/DeepEyes_train_4K
[3] https://huggingface.co/datasets/Mini-o3/VisualProbe_train

Algorithm 1: Load-balanced rollout assignment

Input : Batch of prompts P, set of rollout servers $\mathcal{S}$, turn budgets R
Output : Concatenated generation results O

if *LoadBalance is* ***False*** **then**
  Divide P into k equal chunks $P_1, \ldots, P_k$ where $k = |\mathcal{S}|$;
  for $i = 1$ **to** k **do**
    $O_i \leftarrow \text{Generate}(P_i)$;
else
  Initialize server loads $L \leftarrow [0, \ldots, 0]$ for each $s \in \mathcal{S}$;
  Initialize server assignments $A \leftarrow [\emptyset, \ldots, \emptyset]$;
  Sort indices Idx of P by $R[idx]$ in descending order;
  for $idx \in Idx$ **do**
    $s^* \leftarrow \arg\min_s L[s]$;
    Append idx to $A[s^*]$;
    $L[s^*] \leftarrow L[s^*] + R[idx]$;
  for *server* $S_i \in \mathcal{S}$ **do**
    if $A[i]$ *is not empty* **then**
      $O_i \leftarrow \text{Generate}(P[A[i]])$;
Wait for all O_i and reorder results to match original indices of P;
return Concat(O)

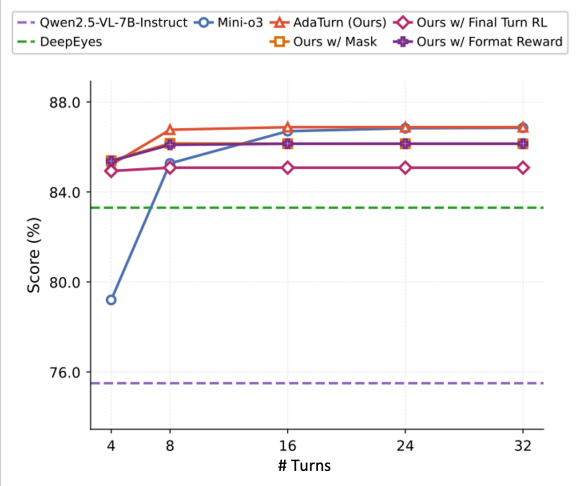


Figure 7: Ablation of the reinforcement learning design. Explicitly training the final-turn answer behavior contributes most of the observed gain. Masking prefix turns, restricting reinforcement learning to only the final forced-answer turn, or adding an extra format reward all underperform the default full-rollout accuracy-only training.

Adding an explicit format reward is also mildly harmful. In our setting, answer extraction already imposes the required structural constraint, so allocating reward mass to formatting does not improve supervision and can distract from the accuracy objective. Taken together, these results support the default full-rollout, accuracy-only FA-DAPO design.

The systems-side ablation in Figure 5 leads to a similar conclusion. The left subfigure reports the minimum and maximum rollout time across engines together with their induced range, while the right subfigure summarizes the gap between the most and least loaded engines. Dynamic turn budgets create substantial worker imbalance under naive assignment, but the proposed scheduler reduces both statistics and yields an average $1.34\times$ rollout speedup, making budget-aware training materially more efficient in practice.

Table 4: General multimodal capability after AdaTurn training. Compared with the base Qwen2.5-VL-7B-Instruct model, AdaTurn mostly preserves or improves performance on broad OCR and reasoning benchmarks, indicating that budget-aware agent training does not come at the cost of general visual-language ability.

	OCR-Related			General	Reasoning	
Model	OCRBench	ChartQA	DocVQA (val)	CV-Bench	MathVista (testmini)	ScienceQA (img)
Qwen2.5-VL-7B-Instruct (Bai et al., 2025b)	81.5	79.6	94.6	73.9	68.2	**89.0**
DeepEyes (Zheng et al., 2026)	-	-	-	-	**70.1**	-
Mini-o3 (Lai et al., 2026)	83.8	77.4	**94.8**	74.4	68.8	84.5
Ours	**85.5**	**80.0**	93.0	**77.6**	68.1	83.7

D How Reinforcement Learning Improves AdaTurn

The ablations above show that training the budget boundary is the main source of improvement. In this section, we analyze in more detail which behaviors FA-DAPO instills in the policy. We identify two aspects: budget-aware instruction following and reasoning with limited evidence.

Budget-Aware Instruction Following. When the policy requests a tool call that would exceed the budget, FA-DAPO rejects the call and injects the budget-exhausted instruction (Appendix I), which requires the model to produce a final answer. Since the accuracy reward can only be obtained from a valid and extractable final answer, a trajectory that ignores this instruction and issues another tool call receives no reward. Reinforcement learning therefore teaches the policy to comply with the injected instruction and to switch from tool use to answer generation at the budget boundary. This behavior is difficult to obtain through prompting alone. As shown in Appendix F, informing Mini-o3 of the budget at inference time reduces, but does not close, the gap to AdaTurn.

Reasoning with Limited Evidence. A valid answer alone is not sufficient, since the reward further requires the answer to be correct, even when the budget is too small to complete the search. To earn this reward, the policy must acquire two complementary behaviors. First, it must choose tool calls with high information gain and avoid unnecessary ones, so that every turn within the limited budget contributes useful evidence. Second, it must reason carefully over the observations it has already collected, because no further evidence can be gathered once the budget is exhausted. Both behaviors involve the whole trajectory rather than only the final turn, which is consistent with the ablation in Appendix C: restricting reinforcement learning to the final forced-answer step performs worse than optimizing the full rollout.

E General Multimodal Capability

Table 4 examines whether AdaTurn preserves general multimodal ability beyond the high-resolution visual-search benchmarks used in the main study. Relative to the base Qwen2.5-VL-7B-Instruct model (Bai et al., 2025b), AdaTurn improves OCRBench (Liu et al., 2024), ChartQA (Masry et al., 2022), and CV-Bench (Tong et al., 2024) while remaining competitive on DocVQA (Mathew et al., 2021), MathVista (Lu et al., 2023), and ScienceQA (Lu et al., 2022). Relative to Mini-o3 (Lai et al., 2026), it performs better on OCRBench (Liu et al., 2024), ChartQA (Masry et al., 2022), and CV-Bench (Tong et al., 2024). These results suggest that budget-aware agent training does not merely specialize the model to a narrow tool-use benchmark, and can be compatible with broad multimodal competence.

F Comparison with Budget-Prompted Mini-o3

In the main comparison (Table 1), Mini-o3 (Lai et al., 2026) is evaluated in its official configuration. Its prompts and training recipe contain no notion of a turn budget, so the budget acts only as a system-side truncation limit: a rollout that still requests a tool at the last turn is terminated without a valid answer. This raises the question of whether the advantage of AdaTurn comes from reinforcement learning with the turn constraint, or simply from exposing the budget to the agent at inference time. To separate these two factors, we evaluate a budget-prompted variant of Mini-o3 that receives explicit budget information at inference time. In each turn, we inform Mini-o3 of the number of remaining turns. When the agent reaches its last tool-calling turn, we instruct it that this is its final tool call and

Table 5: Comparison with budget-prompted Mini-o3 under a four-turn budget. The budget-prompted variant is informed of the remaining turns at each step, told at its last tool-calling turn that no further tool calls are allowed, and then prompted to answer directly. Although budget prompting improves Mini-o3 substantially, AdaTurn still performs better on six of the seven benchmarks.

Model (4 turns)	VisualProbe			V*	HR-Bench		MME-RealWorld
	hard	medium	easy		4K	8K	
Mini-o3 (Lai et al., 2026)	26.5	36.7	42.4	79.2	68.9	63.2	50.1
Mini-o3 (Lai et al., 2026) w/ budget prompt	36.1	**47.7**	58.1	81.9	73.1	67.7	59.7
Ours	**39.0**	47.6	**61.2**	**85.2**	**75.7**	**71.1**	**64.0**

that no further turns are allowed. After this tool call, we prompt it to think and reply directly without any further tool calling, analogous to the forced-answer stage of AdaTurn (Appendix I). All other settings follow the evaluation protocol of Table 1.

Table 5 reports the results under a four-turn budget. Budget prompting substantially improves Mini-o3, for example from 26.5% to 36.1% on VisualProbe-Hard and from 50.1% to 59.7% on MME-RealWorld, which confirms that catastrophic truncation accounts for a large part of its low-budget degradation. Nevertheless, AdaTurn still outperforms the budget-prompted Mini-o3 on six of the seven benchmarks, with gains of $+2.9\%$ on VisualProbe-Hard, $+3.1\%$ on VisualProbe-Easy, $+3.3\%$ on V*, $+2.6\%$ on HR-Bench 4K, $+3.4\%$ on HR-Bench 8K, and $+4.3\%$ on MME-RealWorld, while the two models perform on par on VisualProbe-Medium.

These results indicate that prompting alone is insufficient to make an agent budget-aware. The budget prompt exposes the remaining turns at every step and guarantees that Mini-o3 produces a valid answer, but Mini-o3 has never been trained to act on this information. Its search behavior is still the one learned for long rollouts, so the budget in the prompt has limited influence on how the agent allocates its turns, and the final answer must be synthesized from evidence that was not gathered with the budget in mind. In contrast, FA-DAPO trains the policy under randomized budgets and explicitly optimizes the final-turn decision, so AdaTurn learns to plan its search and aggregate partial evidence within the available turns. The improvement of AdaTurn therefore stems from reinforcement learning with the turn constraint rather than from budget information at inference time alone.

G Turn Budget versus Token Budget

Prior budget-aware reasoning methods mainly control the number of tokens generated within a single response, for example via budget tokens or confidence-based early stopping (Wen et al., 2025; Nogueira et al., 2025). AdaTurn instead measures the inference budget in interaction turns. In this section, we explain this design choice and report the token cost of AdaTurn, showing that the two budgets are complementary rather than competing.

Why Turn Budget. We choose the turn budget as the primary constraint for three reasons. First, the turn budget directly bounds the dominant cost of a multi-turn visual agent. The grounding tool used in this work crops an image region, which is a deterministic in-memory operation with negligible latency. The cost of an extra turn instead lies in the VLM inference it induces. Each tool call interrupts the ongoing rollout, and the returned image tokens must be prefilled before decoding can resume. If the prefix is not cached, for example when the request is rescheduled to a different engine or its KV cache is evicted under memory pressure, a complete prefill over the entire accumulated multimodal context is required. End-to-end latency therefore grows primarily with the number of turns, and this effect becomes even stronger for agents equipped with slower tools such as web search. This observation is consistent with prior measurements of agent efficiency. OSWorld-Human (Abhyankar et al., 2026) benchmarks multimodal computer-use agents and finds that model calls dominate the overall latency, that later steps can take up to $3\times$ longer than the steps at the beginning of a task, and that end-to-end latency reaches tens of minutes as the number of steps grows. In the coding-agent domain, Gao & Peng (2026) show that token cost grows rapidly with the number of turns and that controlling the turn count noticeably reduces cost. Both studies identify the number of turns as the primary driver of agent cost. Second, the turn budget is a natural and interpretable control interface for practitioners, since each turn corresponds to one perception-action cycle. Third, the turn budget is

Table 6: Token usage under a six-turn budget. We report the mean and standard deviation of the total token count (input prompt, tool calls, tool results, and final answer), the number of output tokens, and the number of output tokens per turn. AdaTurn has a similar total token cost to Mini-o3 while generating considerably fewer output tokens.

Token usage (6 turns)	Mini-o3 (Lai et al., 2026)	Ours
Total tokens	**5526.64** $\pm$ 293.25	5713.39 $\pm$ 478.61
Output tokens	698.96 $\pm$ 67.62	**504.24** $\pm$ 35.60
Output tokens per turn	243.88 $\pm$ 15.62	**163.62** $\pm$ 11.67

well suited to reinforcement learning. The budget boundary defines a discrete decision point, namely the forced-answer step of FA-DAPO (Section 3.2), at which the policy can be explicitly trained to commit to an answer. A token budget, in contrast, can expire in the middle of a reasoning segment or a tool call, which leaves the boundary state ill-defined and yields a much noisier training signal.

Token Cost. To verify that the gains of AdaTurn under turn budgets do not come at the price of a higher token cost, we measure the token usage of AdaTurn and Mini-o3 (Lai et al., 2026) under a six-turn budget. The total token count includes the input prompt, the tool-calling text, the tool results (i.e., the image tokens of the cropped regions), and the final answer. As shown in Table 6, the two models exhibit a similar total token cost. The total of AdaTurn is slightly higher (5713.4 vs. 5526.6) because its tool responses carry richer content, such as the tool-calling schema and error feedback (Appendix I). For output tokens, which dominate the decoding cost, AdaTurn is clearly more efficient: it generates 27.9% fewer output tokens in total and 32.9% fewer output tokens per turn than Mini-o3. The improvements of AdaTurn under tight turn budgets are therefore not obtained by spending more tokens. Extending FA-DAPO to joint turn and token budgets is an interesting direction for future work.

H Performance Judge Prompt

To evaluate whether a model prediction matches the ground-truth answer, we use an external judge model. Specifically, we use Qwen3-32B as a binary evaluator that compares the question, the ground-truth answer, and the model prediction, then returns a score of 1 for correct and 0 for incorrect. The exact system prompt and query prompt are shown below.

System Prompt

```
You are an intelligent chatbot designed for evaluating the correctness
    of generative outputs for question-answer pairs.
Your task is to compare the predicted answer with the correct answer
    and determine if they match meaningfully. Here's how you can
    accomplish the task:
------
##INSTRUCTIONS:
- Focus on the meaningful match between the predicted answer and the
    correct answer.
- Consider synonyms or paraphrases as valid matches.
- Evaluate the correctness of the prediction compared to the answer.
```

Query Prompt

```
I will give you a question related to an image and the following text
    as inputs:

1. **Question Related to the Image**: {question}
2. **Ground Truth Answer**: {ground_truth}
3. **Model Predicted Answer**: {prediction}
```

```
Your task is to evaluate the model's predicted answer against the
    ground truth answer, based on the context provided by the question
    related to the image. Consider the following criteria for
    evaluation:
- **Relevance**: Does the predicted answer directly address the
    question posed, considering the information provided by the given
    question?
- **Accuracy**: Compare the predicted answer to the ground truth answer.
     You need to evaluate from the following three perspectives:
(1) If the ground truth answer is open-ended, consider whether the
    prediction accurately reflects the information given in the ground
    truth without introducing factual inaccuracies. If it does, the
    prediction should be considered correct.
(2) If the ground truth answer is a definitive answer, strictly compare
     the model's prediction to the actual answer. Pay attention to unit
     conversions such as length and angle, etc. As long as the results
    are consistent, the model's prediction should be deemed correct.
(3) If the model's prediction is irrelevant string, such as punctuation
    , newlines, tabs, whitespace, or emptry string etc., without a
    meaningful answer, set the score as 0.
**Output Format**:
Your response should include an integer score indicating the
    correctness of the prediction: 1 for correct and 0 for incorrect.
    Note that 1 means the model's prediction strictly aligns with the
    ground truth, while 0 means it does not.
The format should be "Score: 0 or 1"
```

I AdaTurn Agent Prompts

AdaTurn uses a small set of structured prompts to control the agent loop. We provide the exact prompts used for the initial system instruction, tool observations, tool errors, and the forced-answer stage when the turn budget is exhausted.

System Prompt

```
You are a helpful assistant. Answer the user's question based on the
    image provided. Output your thinking process within the <think> and
     </think> tags. Whenever you find anything unclear, you can zoom in
     a specific region in the given image to see more clearly by
    outputing <grounding>{"bbox_2d": [x0, y0, x1, y1], "source": "
    original_image"}</grounding>, where (x0, y0) and (x1, y1) are the
    top-left and bottom-right coordinates of the region that you want
    to zoom in, respectively (suppose the width and height of the image
     are 1.0), and 'source' refers to the image that you zoom in and
    could be either 'original_image' or 'observation_i'. Once the final
     answer is confirmed, put it within <answer> and </answer>.
```

Tool Observation Response

```
After the above Action {action_turn}, here is the the zoom-in image (
    Observation {observation_turn}):
<|vision_start|><|image_pad|><|vision_end|>.
Continue your reasoning process inside <think> and </think>. If needed,
     you can continue to zoom in on the original image or any of the
    observations, by outputting <grounding> and </grounding> as before.
     If the final answer is confirmed, put your final answer inside <
    answer> and </answer>.
```

Tool Error Response

```
Please analyze the error information obtained from the function tool
    and adjust your response. Countinue your reasoning process inside <
    think> and </think>.
```

Force-to-Answer Prompt

```
<think>I have exhausted my tool-calling budget and must answer the
    question using the information I have obtained, even though I may
    make mistakes due to a lack of data.
```

J VISUALIZATION OF ADATURN ROLLOUT TRAJECTORIES

Figures 8 and 9 show representative successful trajectories under a tight four-turn budget, while Figures 10 and 11 show successful trajectories under an eight-turn budget. These examples highlight two qualitative behaviors encouraged by AdaTurn. First, the model does not simply terminate at the budget boundary; instead, it uses the available observations to synthesize a final answer when further grounding is disallowed. Second, the trajectory remains adaptive under different budgets: under $T_{\max} = 4$, the model makes compressed decisions based on limited evidence, whereas under $T_{\max} = 8$, it can continue refining the crop sequence until the target detail becomes identifiable.

The examples also illustrate that AdaTurn learns a useful form of budget-aware evidence aggregation. In Figure 8, the final crop is less informative than an earlier observation, but the model still answers correctly by relying on the sharper evidence collected before the final turn. In Figure 9, the model revisits the original image, shifts attention to a previously overlooked region, and then answers correctly at the forced-answer step. Figures 10 and 11 show that with a larger budget, AdaTurn can sustain longer search trajectories to identify small traffic-light signals and fine-grained scene text, respectively.

K FAILURE ANALYSIS

Figure 12 shows a representative failure case under a four-turn budget. The question asks for the direction faced by the person wearing the plaid shirt, but the available observations do not provide sufficient evidence to isolate the correct individual with confidence. After exhausting the turn budget, the model is forced to answer using incomplete information and instead anchors on a different person in the scene, which leads to a hallucinated final answer.

This failure mode is consistent with the intended trade-off of AdaTurn. Forced-answer training mitigates catastrophic truncation by ensuring that the model produces a valid answer even when the search budget is insufficient, but it cannot guarantee correctness when the collected evidence remains ambiguous. In such cases, the model may still hallucinate by overcommitting to an imperfect hypothesis, especially in crowded scenes where small identity distinctions are difficult to resolve within a short rollout.

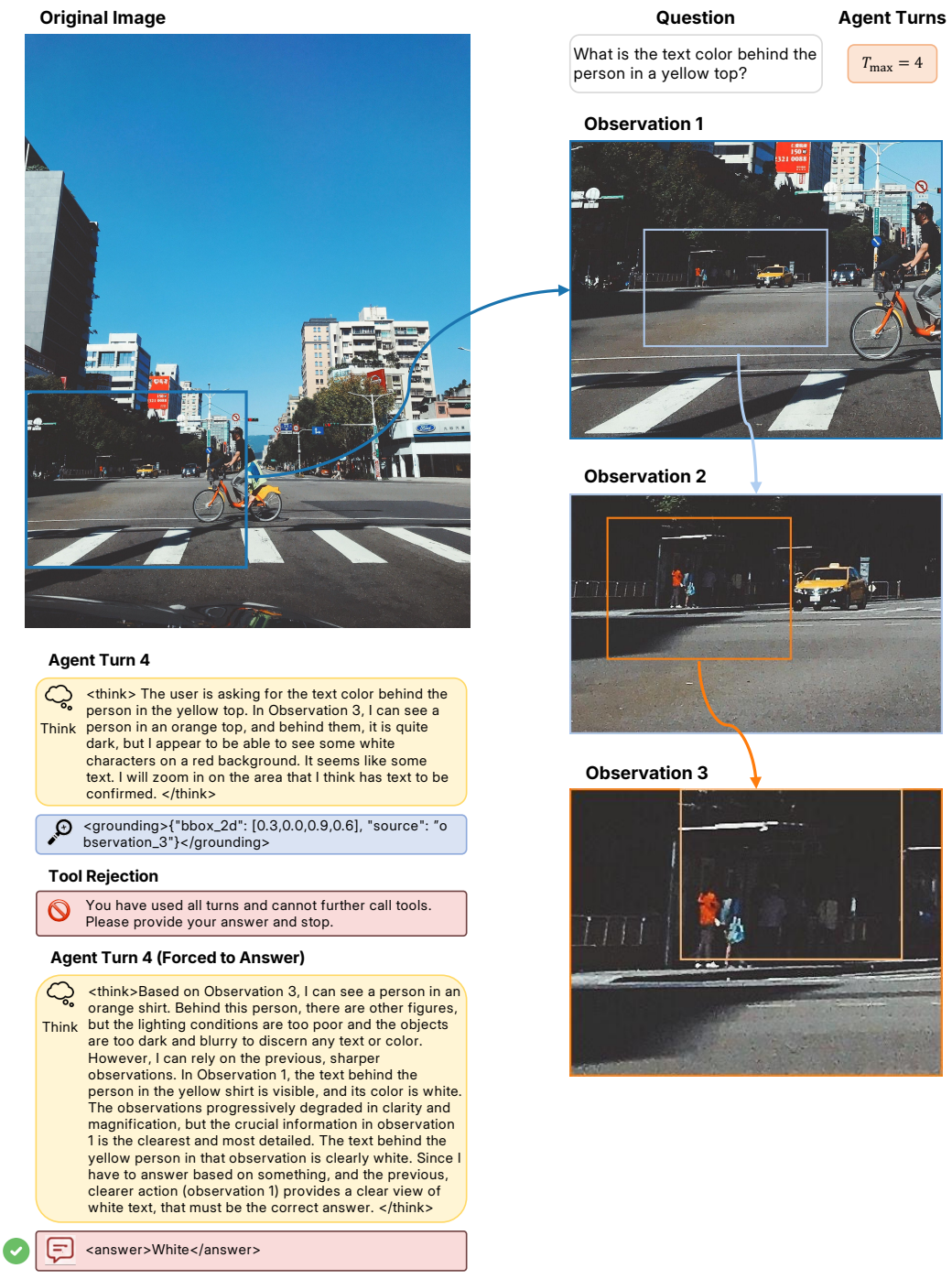


Figure 8: Successful rollout under a four-turn budget. The agent localizes the pedestrian in the yellow top, explores the region behind them, and then answers at the forced-answer step by relying on an earlier, clearer observation that reveals the relevant text color as white. This example illustrates that AdaTurn can retain and use previously collected evidence even when the final crop is less informative.

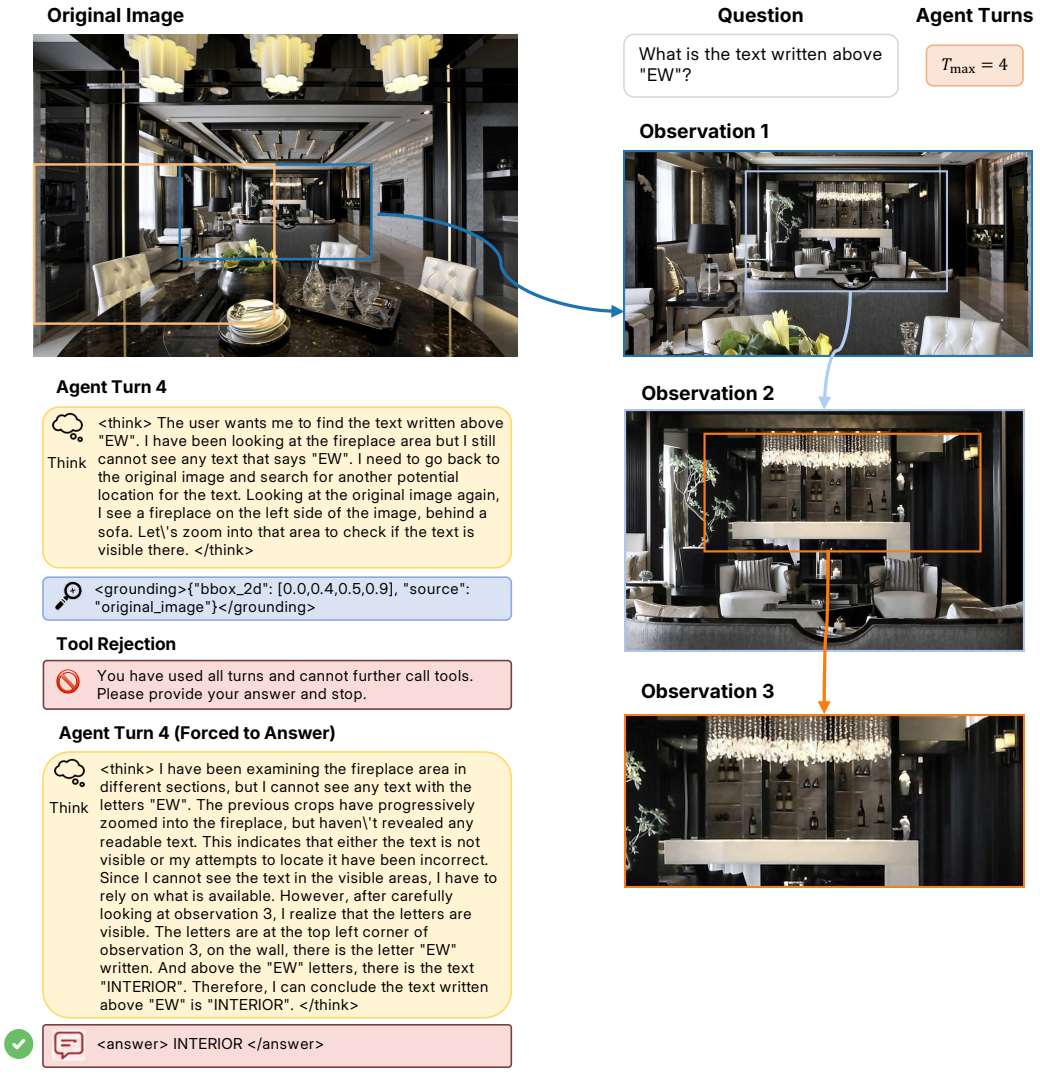


Figure 9: Successful rollout under a four-turn budget. The agent initially searches the fireplace region, fails to confirm the text, then redirects attention to another plausible location and identifies that the text above “EW” is “INTERIOR” before producing the forced-answer response. This example shows that AdaTurn can revise its search plan rather than committing prematurely to its first hypothesis.

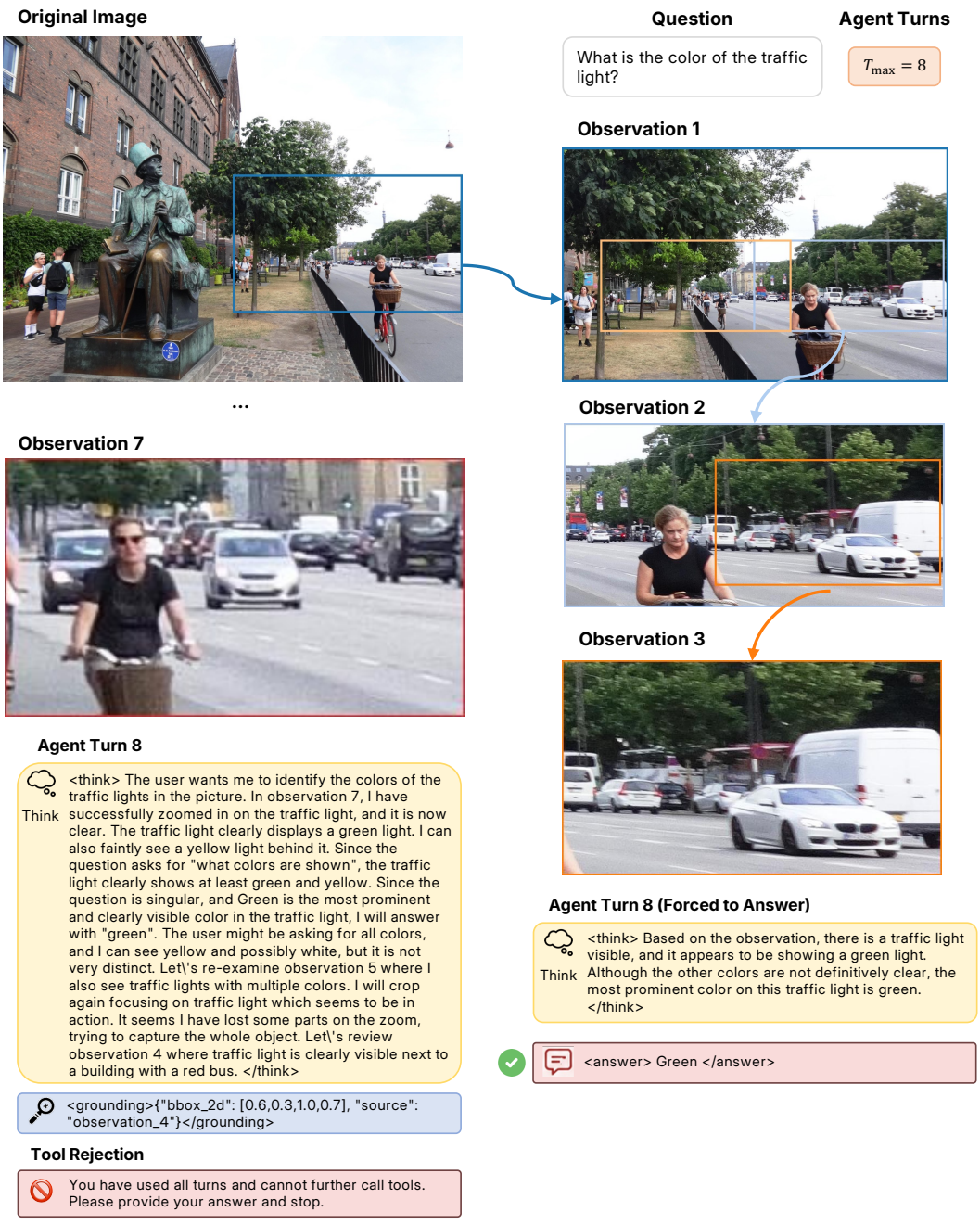


Figure 10: Successful rollout under an eight-turn budget. The agent performs a longer sequence of crops to isolate a distant traffic light and ultimately answers that the light is green. The additional budget enables a deeper search trajectory that would be difficult to complete reliably under a tighter turn limit.

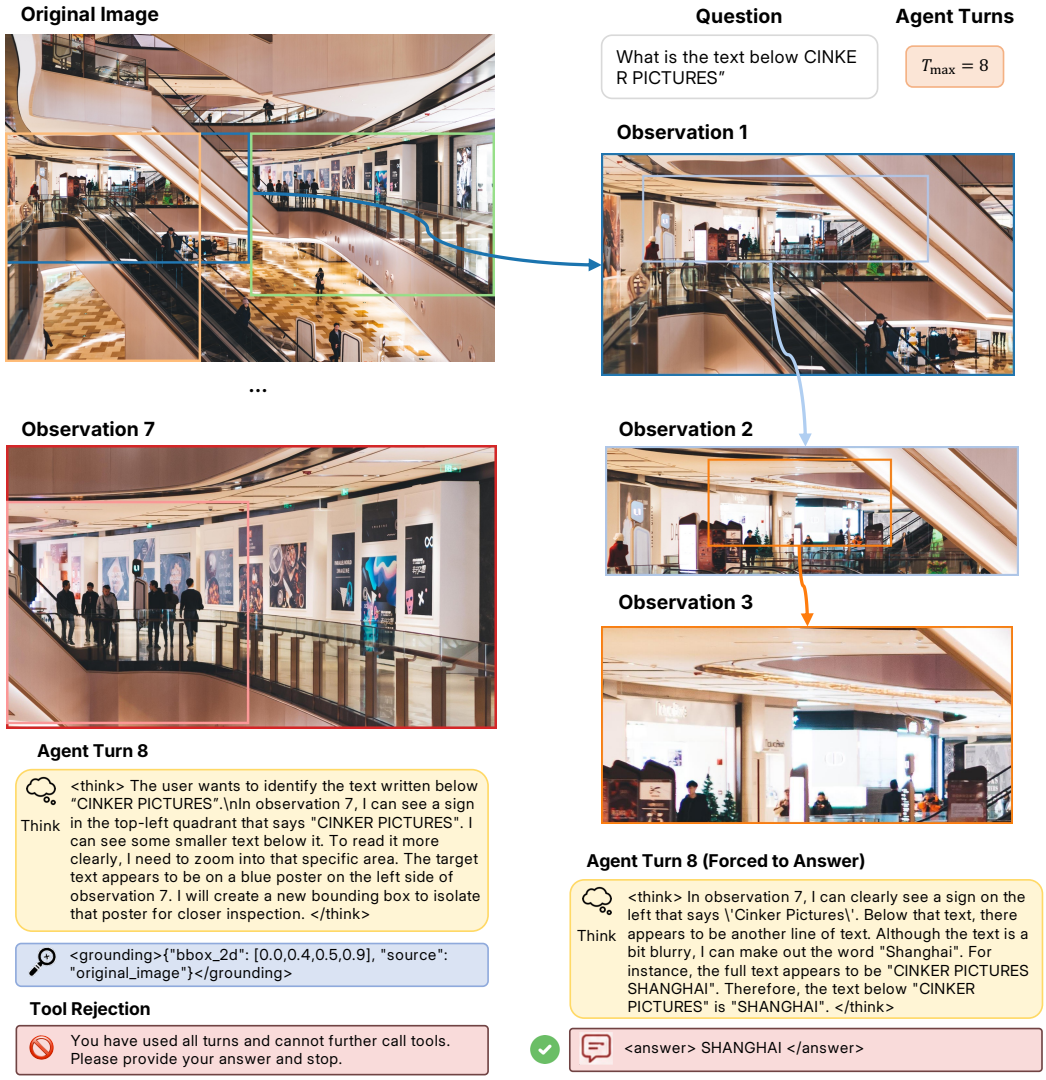


Figure 11: Successful rollout under an eight-turn budget. The agent progressively zooms into a small storefront poster, identifies the sign "CINKER PICTURES," and then uses the final forced-answer step to infer that the text below it is "SHANGHAI." This example illustrates AdaTurn's ability to use longer trajectories for fine-grained text reading while still terminating with a valid answer at the budget boundary.

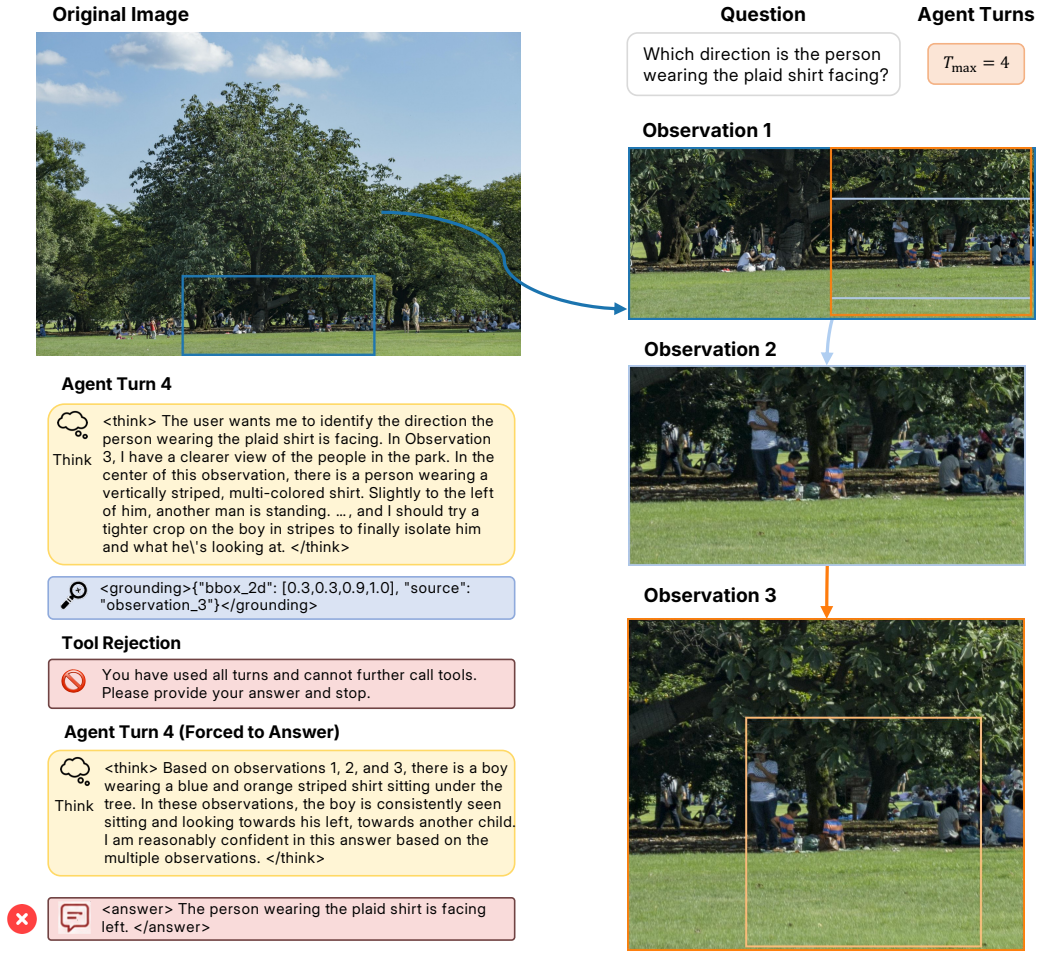


Figure 12: Failure case under a four-turn budget. The agent searches a crowded park scene for the person in a plaid shirt, but the collected crops do not isolate the correct target before the budget is exhausted. At the forced-answer step, the model hallucinates by grounding its decision on a different person and incorrectly answers that the target is facing left.