

Object-Centric Monte Carlo Tree Search for Hallucination-Robust Visual Question Answering

Jie Bao^a, Tianqiqi Zhu^b, Pingjiao Li^a, Yaohui Jin^c, Junjie Yang^a, Zhicheng Sui^{a,*}

^a*School of Electronic Information Engineering, Shanghai DianJi University, No. 300, Shuihua Road, Pudong New Area District, Shanghai, 201306, China*

^b*WLSA Shanghai Academy, No. 299, Tongji Road, Baoshan District, Shanghai, 200433, China*

^c*Shanghai Jiao Tong University, Shanghai, 200240, China*

Abstract

Reliable visual question answering requires models to recognize visual patterns and reject semantically plausible but visually absent objects. However, large vision-language models typically rely on passive one-pass image encoding, which can lead to incomplete visual exploration, weak visual grounding, and object hallucination. This paper presents MAMCTS, a training-free multi-action Monte Carlo tree search framework that formulates hallucination-robust visual question answering as an object-centric decision search problem. Given an image-question pair, MAMCTS builds a search tree over joint visual-textual states and explores actions including key-object focusing, adaptive cropping, context recovery, and query re-evaluation. A frozen vision-language model is used as a confidence-aware evaluator rather than being fine-tuned, while a dual-verification reward and dynamic pruning strategy guide the search toward visually supported object hypotheses. Final answers are aggregated from high-confidence reasoning paths to reduce the instability of single-pass inference. Experiments on POPE-derived MSCOCO real-world and synthetic benchmarks, together with evaluations on GQA, A-OKVQA, and POPE-COCO, show that MAMCTS consistently reduces false-positive object predictions across multiple vision-language backbones. On MSCOCO real-world splits, it improves LLaVA-v1.5 accuracy by up to 5.13 per-

*Corresponding author.

Email addresses: 18105647651@163.com (Jie Bao), jerryfoothill@gmail.com (Tianqiqi Zhu), 1550324430@qq.com (Pingjiao Li), jinyh@sjtu.edu.cn (Yaohui Jin), yangjj@sdju.edu.cn (Junjie Yang), suizc@sdju.edu.cn (Zhicheng Sui)

centage points and reduces the false positive rate by up to 20.40 points. On synthetic splits, it improves accuracy by up to 9.79 points and reduces the false positive rate by up to 28.27 points. These results demonstrate that object-centric tree search is an effective inference-time mechanism for robust visual pattern recognition in pretrained multimodal models. Code and data are made publicly available on GitHub.

Keywords: Visual question answering, Monte Carlo Tree Search, Object hallucination, Visual grounding, Large vision-language models, Pattern recognition

1. Introduction

Visual Question Answering (VQA) requires models to recognize visual content and combine it with cross-modal reasoning to produce precise answers [1, 2]. Large Vision-Language Models (LVLMs), including LLaVA-1.5 [3, 4], InstructBLIP [5], Qwen2-VL [6], and InternVL-2 [7], have substantially improved zero-shot perception and reasoning. For the pattern recognition community, however, reliable VQA presents a more specific challenge: the model must discriminate visually supported object patterns from semantically plausible but absent hypotheses under incomplete or misleading evidence. Existing LVLMs remain constrained by passive, one-pass visual processing [8], rather than actively revisiting informative regions as human observers do. This limitation creates three deficiencies:

1. Inefficient exploration of relevant regions: The lack of an active "perception-action" loop leads to suboptimal decision paths, particularly in tasks requiring precise visual grounding or multi-step reasoning [9].
2. Susceptibility to object hallucination: Without a mechanism for explicit verification, models are prone to hallucinating non-existent objects, especially in adversarial scenarios [10, 11].
3. Reliance on retraining or distillation: Addressing these issues often requires extensive supervised fine-tuning or distillation [12], which incurs high computational costs and hinders the flexible deployment of models in varied contexts.

Recent approaches use reinforcement learning to elicit visual tool use [13], while Pattern Recognition research has reduced VQA bias through modal information interac-

tion and question transformation [14]. These methods nevertheless require training or additional reasoning machinery. Few frameworks provide adaptive visual verification as a fully training-free inference procedure.

To address these challenges, this paper introduces MAMCTS, a novel Object-Centric Multi-Action Tree Search framework. Drawing inspiration from cognitive decision-making [15, 16], MAMCTS reframes VQA as a sequential decision process [17]. Operating as a training-free inference wrapper on top of frozen LVLMs, it intelligently navigates the joint state space of image regions and textual queries. By emulating human-like iterative visual localization, MAMCTS reframes visual reasoning as a systematic decision optimization process. It maintains an explicit belief state of observed and missing key objects, utilizing a rigorous search methodology to balance exploration with semantic confirmation, thereby ensuring computational efficiency without sacrificing reasoning depth.

In summary, the main contributions of this work are as follows:

- **A Novel Adaptive Reasoning Framework:** We propose MAMCTS, a training-free inference framework that integrates Monte Carlo Tree Search (MCTS) with object-centric region proposals and can be readily integrated with frozen LVLMs. This framework extends VQA from passive single-pass perception to active, iterative visual reasoning.
- **Object-Centric Action Space for Verification:** We construct a specialized action space that shifts reasoning from pixel-level searching to semantic-level verification. This design enables the model to explicitly propose, verify, and localize objects, thereby directly reducing hallucination caused by passive visual processing.
- **Synergistic Inference Pipeline with Ablation Validation:** We integrate LVLm-based evaluation with a training-free MCTS architecture. Through comprehensive ablation studies, we demonstrate that the synergy of object-centric actions, hallucination-aware rewards, and ensemble decision-making is critical to achieving robust adaptive reasoning.



Figure 1: Comparison of reasoning paradigms for complex VQA.

- **Superior Performance and Robustness:** Extensive experiments on MSCOCO Real-World and abstract datasets demonstrate that MAMCTS outperforms state-of-the-art baselines. Compared with LLaVA-v1.5, MAMCTS yields the largest gains in adversarial and synthetic settings, with accuracy improvements of up to 9.79% and FPR reductions of up to 28.27%, demonstrating its effectiveness for mitigating hallucinations in multimodal reasoning.

The remainder of this paper is organized as follows. Section 2 reviews related work on visual reasoning paradigms. Section 3 presents the object-centric sequential decision formulation of MAMCTS, including the state definition, action space, search policy, reward design, and pruning strategy. Section 4 reports the principal benchmark results and cross-model generalization analysis, and summarizes the extended efficiency, sensitivity, ROC, ablation, qualitative, and comparative evaluations detailed in Supplementary Sections S1–S4. Finally, Section 5 discusses the strengths, limitations, implications, and future directions of the proposed framework.

2. Related Work

Visual reasoning in VQA has evolved from static pattern matching to dynamic, knowledge-driven decision-making. We organize the most relevant literature into five complementary lines, progressing from passive perception and visual grounding to active search and explicit tree-based reasoning.

Passive Holistic Reasoning in LVLMs. As illustrated in Figure 1(a), LLaVA-1.5

[4] and InstructBLIP [5] answer from a single global representation. Scene Graphs [18, 19] and Graph Neural Networks [20, 21] add structured visual dependencies [22], but remain largely fixed after construction. Ke et al. [23] instead strengthen visual grounding through language–image consistency augmentation and distillation. This improves cross-modal evidence alignment, but does not provide query-adaptive exploration of uncertain regions.

Active Visual Search (LVLM + Visual Search). Active-perception methods iteratively crop query-relevant regions (Figure 1(b)). Mohamud et al. [24] formulate VQA as an encoder–decoder perception–action cycle. DyFo [25] and multi-scale attention networks [26] also refine evidence through dynamic focusing. These methods improve fine-grained perception, but lack an explicit mechanism that jointly balances exploration, verification, and exploitation.

Collaborative Expert Reasoning (LVLM + Visual Expert). The third paradigm, illustrated in (Figure 1(c)), introduces specialized modules or auxiliary agents to verify and refine the reasoning process. As exemplified by DeepEyes [13] and Agent-of-Thoughts [12], such methods use reinforcement learning or external tools [27, 28] to ground LVLM outputs in visual evidence. More advanced systems, such as Woodpecker [29] and MoE-LLaVA [30], further improve reliability through post-hoc correction modules or sparse mixture-of-experts architectures. Although these approaches can improve accuracy, they often introduce substantial computational overhead and architectural complexity, which limits their deployment practicality in resource-constrained settings.

MCTS-Driven Reasoning (Figure 1(d)). Jegham et al. [31] analyze visual-reasoning consistency and robustness in multimodal LLMs. Explicit search complements such evaluation by revisiting uncertain evidence before prediction. It explores candidate regions more actively than passive LVLMs and balances exploration and exploitation more systematically than heuristic visual search. MAMCTS combines this search principle with object-centric state tracking in a training-free framework [8, 32].

Search-Based Reasoning with ToT and LATS. Recent tree-search-style reasoning frameworks, such as Tree-of-Thoughts (ToT) [17] and Language Agent Tree Search

Table 1: Comparison between MAMCTS and representative search-based reasoning frameworks.

Method	Search State	Visual Grounding	Object Verification	Training-Free	External Tools	Hallucination Mitigation
ToT	Textual thoughts	No	No	Yes	No	Limited
LATS	Language-agent trajectories	Limited	Limited	Yes	Optional	Limited
MAMCTS (ours)	Joint visual-textual states	Yes	Yes	Yes	No	Explicit

(LATS) [32], extend large-model inference by explicitly exploring multiple candidate reasoning trajectories instead of relying on a single forward pass. Their core advantage lies in structured deliberation over alternative thought paths, which has shown promise for complex multi-step decision problems. However, these methods primarily operate over textual reasoning states or language-agent trajectories, and they do not explicitly support the combination of fine-grained visual grounding and object-level verification required for robust VQA. Table 1 summarizes the main conceptual differences between ToT, LATS, and MAMCTS. In contrast, MAMCTS performs search over joint visual-textual states, where each node encodes both reasoning history and the current grounded visual context. This object-centric formulation makes the search process better aligned with VQA settings that require explicit grounding, adaptive visual exploration, and robustness against object hallucination.

3. Method

This section first presents the overall architecture and execution workflow of MAMCTS. It then formulates the object-centric state representation and hierarchical action space. Next, it describes the tree-search procedure, including PUCT-based selection, Dirichlet-noise exploration, parallel simulation, and backpropagation. Finally, it introduces the hallucination-aware reward function and dynamic pruning strategy used to improve reasoning robustness.

3.1. Framework Overview

We propose MAMCTS, an object-centric, training-free reasoning framework that reformulates VQA as an active sequential decision-making process. As depicted in Fig-

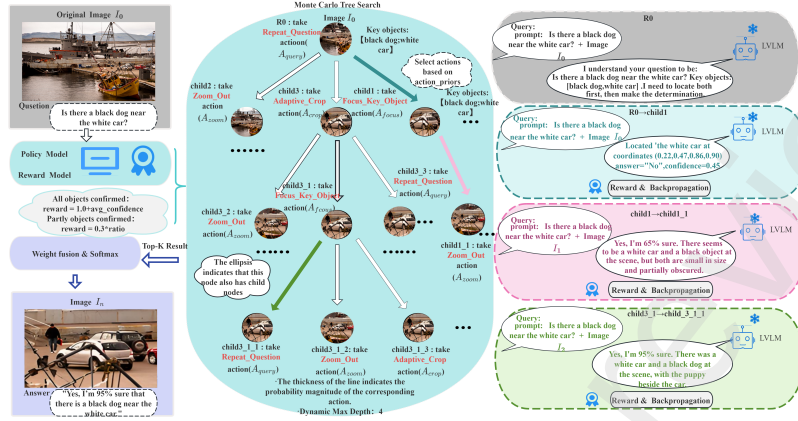


Figure 2: Overview of the MAMCTS framework. The policy model selects actions via PUCT, the search tree expands candidate visual states, the LVLM performs simulation, and the resulting rewards are backpropagated for final decision aggregation.

ure 2, MAMCTS acts as an inference-time wrapper over frozen Large Vision-Language Models (LVLMs), bridging global visual perception and fine-grained visual grounding without parameter updates.

The framework architecture consists of three synergistic stages:

(1) **Object-Centric Initialization:** Given an image I and a complex query Q , the framework first uses the frozen LVLM to parse the query and extract a set of semantically relevant key objects $\mathcal{K}$. This process is illustrated in the root interaction panel (R0) on the right side of Figure 2, where the model identifies targets such as “black dog” and “white car”. This step establishes the initial action priors, grounding the subsequent search in the semantics of the query rather than in random image patches.

(2) **Adaptive Multi-Action Tree Search:** This phase constitutes the core reasoning engine of the framework. In contrast to traditional paradigms that rely on passive, single-pass image processing, MAMCTS dynamically constructs a search tree where each node encapsulates a specific “visual state,” a process visualized by the central “Monte Carlo Tree Search” structure in Figure 2. The system iteratively explores this state space through four search operators: `Focus_Key_Object`, `Adaptive_Crop`, `Zoom_Out`, and `Repeat_Question`. These operators are explicitly depicted as the di-

rectional edges connecting parent and child nodes in the diagram (e.g., the transition labeled "child3: take Zoom_Out action" or "child1: take Focus_Key_Object action").

As illustrated in the Policy Model component (Figure 2, left panel), the selection of actions is governed by a modified prior-guided UCB (PUCT) policy designed to balance the trade-off between exploration (probing new regions) and exploitation (refining promising views). The optimization is guided by a reward signal derived from the LVLM's verification outcomes and confidence estimates for the current view. This evaluation mechanism is demonstrated in the right-side interaction panels (e.g., sequences like "R0 $\rightarrow$ child1"), where the LVLM's probabilistic output (e.g., "95% sure") is integrated into the "Reward & Backpropagation" blocks to iteratively update the tree statistics. Finally, to ensure computational tractability, a dynamic pruning mechanism eliminates low-value branches, thereby concentrating resources on high-probability reasoning paths.

(3) Ensemble Decision Aggregation: After the search simulations are completed, the framework selects the top-scoring visited nodes for final decision aggregation, a transition designated by the "Top-K Result" indicator exiting the main tree structure in Figure 2. These high-value visited nodes are subsequently channeled into the "Weight Fusion & Softmax" module (located in the bottom-left purple panel). Within this component, an ensemble mechanism aggregates the local answers derived from the optimized visual contexts based on their path scores. By synthesizing these multiple high-confidence paths, the system produces a final, robust response—exemplified by the definitive output, for example, "Yes, I'm 95% sure that there is a black dog near the white car." This aggregation step helps reduce the stochasticity associated with single-pass inference.

Algorithm 1: The Execution Pipeline of MAMCTS.

Input: Image I , Question Q , Frozen LVLM $\mathcal{M}$, Hyperparameters ($n_{iter} = 24$, $d_{max} = 4$).

Output: Final Answer A^* .

1. Phase I: Initialization;
2. Parse Q using $\mathcal{M}$ to extract the key-object set $\mathcal{K} = \{k_1, k_2, \dots\}$;
3. Initialize the root state $s_0 = (I_{raw}, [0, 0, 1, 1], \emptyset, \mathcal{K}, \emptyset, \emptyset, 0)$;
4. Create root node n_0 and compute initial action priors $P(a \mid s_0)$;
5. Phase II: Adaptive Tree Search Loop;
6. While total simulations $N < n_{iter}$;
7. Selection: Traverse from n_0 to a leaf node n_L using Eq. (8);
8. Expansion: If n_L is not terminal ($d < d_{max}$): (1) Compute action priors for node n_L . If $n_L = n_0$, perturb the prior with Dirichlet noise using Eq. (14); otherwise use the unperturbed prior $P(a \mid s_{n_L})$. (2) Sample and execute action a to generate child state s' (e.g., crop image). Add child node;
9. Simulation:
 - (1) Spontaneous Check: Prompt $\mathcal{M}$ to list visible objects L_{spon} ;
 - (2) Consistency Verification;
 - (3) Tiered Reward; see Eq. (17);
10. Backpropagation: Update visit counts N , cumulative rewards W , mean values Q for all ancestor nodes up to the root;
11. Dynamic Pruning: If $N \pmod{5} = 0$, prune low-value nodes satisfying the criterion in Eq. (19);
12. End
13. Phase III: Decision Aggregation
14. For each visited node n , define its leaf-level simulation reward as $r_{leaf}(n) := R(s_n)$, where $R(s_n)$ is the immediate reward obtained in simulation, and compute the final ranking score as $S(n) = 0.65 Q(n) + 0.35 r_{leaf}(n)$;
15. Select top- k nodes $\mathcal{T} \subset \mathcal{N}$;
16. Generate local answers $\{a_n\}$ and confidences $\{c_n\}$ for each $n \in \mathcal{T}$ using $\mathcal{M}$;
17. Compute weighted votes: $w_n = 0.7S(n) + 0.3c_n$;
18. Return the final answer A^* by weighted soft voting over the candidate answers from nodes in $\mathcal{T}$.

The complete algorithmic procedure is formally detailed in Algorithm 1. This workflow demonstrates how MAMCTS integrates explicit visual search actions with probabilistic tree search to achieve interpretable and robust multimodal reasoning.

3.2. Object-Centric State Representation and Action Space

We model the VQA reasoning process as a partially observable sequential decision problem over an object-centric state space [33]. Unlike pixel-level exploration, our framework operates on an object-centric representation characterized by a state space $\mathcal{S}$, the action space $\mathcal{A}$, and the transition mechanism f_{trans} . For clarity, this subsection focuses on the definitions of states and actions, while the reward formulation is introduced in Section 3.4.

3.2.1. Mathematical Formulation of Multimodal States

Each node n in the search tree corresponds to a state $s_t \in \mathcal{S}$ at decision step t . We define the state at decision step t as

$$s_t = (\mathbf{v}_t, \mathbf{r}_t, \mathcal{H}_t, \mathcal{K}_t^{\text{miss}}, \mathcal{K}_t^{\text{focus}}, \mathcal{K}_t^{\text{obs}}, d_t) \in \mathcal{S}, \quad (1)$$

where $\mathbf{v}_t \in \mathbb{R}^{H \times W \times C}$ denotes the current visual observation, $\mathbf{r}_t = [x_1, y_1, x_2, y_2] \in [0, 1]^4$ is the normalized spatial region in the original image, $\mathcal{H}_t = (a_0, \dots, a_{t-1})$ is the ordered action history, $\mathcal{K}_t^{\text{miss}} \subseteq \mathcal{K}$ denotes the set of key objects that remain unverified, $\mathcal{K}_t^{\text{focus}} \subseteq \mathcal{K}$ denotes the set of key objects selected for focused reasoning but not yet verified, $\mathcal{K}_t^{\text{obs}} \subseteq \mathcal{K}$ denotes the set of verified key objects, and d_t is the current search depth. The cropped observation is determined by the region, i.e., $\mathbf{v}_t = \text{Crop}(I_{\text{raw}}, \mathbf{r}_t)$. Initially, $\mathcal{K} = \text{Extract}(Q)$, $\mathbf{v}_0 = I_{\text{raw}}$, $\mathbf{r}_0 = [0, 0, 1, 1]$, $\mathcal{H}_0 = \emptyset$, $\mathcal{K}_0^{\text{miss}} = \mathcal{K}$, $\mathcal{K}_0^{\text{focus}} = \emptyset$, $\mathcal{K}_0^{\text{obs}} = \emptyset$, and $d_0 = 0$.

3.2.2. Hierarchical Action Space and Transitions

The action space $\mathcal{A}$ is a discrete, semantically structured set. Let $f_{\text{trans}} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ denote the transition mechanism. For any action $a_t \in \mathcal{A}$, the transition updates the action history and search depth as $H_{t+1} = H_t \oplus a_t$ and $d_{t+1} = d_t + 1$. The next observation is obtained from the updated region, i.e., $v_{t+1} = \text{Crop}(I_{\text{raw}}, r_{t+1})$, unless otherwise noted. We define the primary action operators as follows:

1. Focus_Key_Object ($\mathcal{A}_{\text{focus}}$): The Focus_Key_Object action selects a target object $k \in \mathcal{K}_t^{\text{miss}} \setminus \mathcal{K}_t^{\text{focus}}$ for object-centric reasoning without immediately declaring it as verified. In the current implementation, this action records which key object should be attended to in subsequent reasoning steps while preserving the current visual region:

$$\begin{aligned} r_{t+1} &= r_t, & v_{t+1} &= v_t, \\ H_{t+1} &= H_t \oplus a_t, & d_{t+1} &= d_t + 1, \\ \mathcal{K}_{t+1}^{\text{focus}} &= \mathcal{K}_t^{\text{focus}} \cup \{k\}, & \mathcal{K}_{t+1}^{\text{miss}} &= \mathcal{K}_t^{\text{miss}}, \\ \mathcal{K}_{t+1}^{\text{obs}} &= \mathcal{K}_t^{\text{obs}}. \end{aligned} \quad (2)$$

Successful verification in the subsequent simulation phase updates the object-state variables by removing k from the missing-object set and adding it to the

observed-object set.

2. **Adaptive_Crop** ($\mathcal{A}_{crop}$): The **Adaptive_Crop** action aggregates a subset of query-relevant objects $\mathcal{O} \subseteq \mathcal{K}_t^{miss}$ in order to preserve their spatial relationships during reasoning. Instead of taking the raw set-theoretic union of object regions, the action first computes the minimal enclosing rectangle covering all selected objects and then applies an adaptive spatial margin to retain sufficient contextual information. Formally, let

$$\tilde{r}_{t+1} = \text{Enclose}(\{\text{BoundingBox}(v_t, k) \mid k \in \mathcal{O}_{valid}\}), \quad (3)$$

where $\mathcal{O}_{valid} \subseteq \mathcal{O}$ denotes the subset of selected objects for which localization is available in the current view. The final crop region is then defined as

$$r_{t+1} = \text{Clip}(\text{Expand}(\tilde{r}_{t+1}, \delta_t)), \quad (4)$$

The adaptive margin δ_t is determined by the spatial scale and density of the localized objects, and $\text{Clip}(\cdot)$ constrains the expanded region within the image boundary. This action preserves the spatial configuration needed for relational reasoning while reducing irrelevant background context.

3. **Zoom_Out** ($\mathcal{A}_{zoom}$): The **Zoom_Out** action expands the current visual region around its center to recover broader context. Let $r_t = [x_1, y_1, x_2, y_2]$ with center (c_x, c_y) , width $w = x_2 - x_1$, and height $h = y_2 - y_1$. The next region is obtained by scaling the current region by a factor $\beta > 1$ (default $\beta = 1.5$) and clipping it to image boundaries:

$$r_{t+1} = \text{Clip}\left(c_x - \frac{\beta w}{2}, c_y - \frac{\beta h}{2}, c_x + \frac{\beta w}{2}, c_y + \frac{\beta h}{2}\right). \quad (5)$$

4. **Repeat_Question** ($\mathcal{A}_{query}$): The **Repeat_Question** action re-evaluates the query Q on the current visual view without changing the spatial region. Unlike the dual-verification mechanism introduced in Section 3.3.3, this action does not itself determine object validity; instead, it triggers an additional reasoning step on the same observation. Formally, $r_{t+1} = r_t$, $v_{t+1} = v_t$. Under the general transition rule defined above, the action history and search depth are updated as

$H_{t+1} = H_t \oplus a_t$ and $d_{t+1} = d_t + 1$. The object-state variables remain unchanged, i.e.,

$$\mathcal{K}_{t+1}^{\text{miss}} = \mathcal{K}_t^{\text{miss}}, \quad \mathcal{K}_{t+1}^{\text{focus}} = \mathcal{K}_t^{\text{focus}}, \quad \mathcal{K}_{t+1}^{\text{obs}} = \mathcal{K}_t^{\text{obs}}. \quad (6)$$

This operator preserves the current visual state while allowing the model to refine its reasoning on the same crop.

To improve reproducibility, we further specify object localization in Adaptive_Crop as an explicit localization-and-validation procedure, rather than a free-form prompting step. For each queried object k , the procedure first queries the frozen LVLM to obtain a candidate bounding-box proposal, denoted as $\text{BoundingBox}(v_t, k)$ using a structured instruction: "Task: Locate the coordinates of [Queried Object] in the image (format: (x_1, y_1, x_2, y_2) , range 0–1). Return the result in JSON format." The returned proposal $b_k = [x_1, y_1, x_2, y_2]$ is accepted only if the output can be parsed as valid JSON, contains all four coordinates, satisfies $0 \leq x_1 < x_2 \leq 1$ and $0 \leq y_1 < y_2 \leq 1$, and yields a crop satisfying the area-ratio threshold. Otherwise, the proposal is marked invalid and excluded from $\mathcal{O}_{\text{valid}}$. The enclosing crop in Eq. (3) is then constructed only from the subset $\mathcal{O}_{\text{valid}} \subseteq \mathcal{O}$ of objects whose proposals pass these checks. If $\mathcal{O}_{\text{valid}} = \emptyset$, the current Adaptive_Crop attempt is regarded as unsuccessful, and the framework falls back to the Zoom_Out action to recover broader visual context rather than terminating the search branch.

To further assess the robustness of this component, we measured how often the fallback mechanism was activated during inference. Across 6,000 test samples of the MSCOCO Real-World Adversarial split, the fallback strategy was triggered at least once in 71 samples, corresponding to a sample-level fallback frequency of 1.18%. This low trigger rate suggests that JSON-based coordinate extraction is stable in practice, while the fallback mechanism mainly serves as a safeguard for occasional malformed outputs or unusable spatial proposals.

3.2.3. Semantic Preference for Object-Targeted Actions

To keep the search aligned with query-relevant objects, we define a semantic preference indicator for object-targeted actions. This signal is used to prioritize object-

targeted actions during search rather than to impose a separate hard admissibility constraint. For a focus action instantiated on object k , we define

$$\Phi(s_t, a_{\text{focus}}(k)) = \mathbb{I}(k \in \mathcal{K}_t^{\text{miss}} \setminus \mathcal{K}_t^{\text{focus}}). \quad (7)$$

where $a_{\text{focus}}(k)$ denotes the Focus_Key_Object action instantiated for target object k , and $\mathbb{I}(\cdot)$ is the indicator function. This preference encourages the search to prioritize key objects that remain unverified and have not yet been explicitly focused on in the current reasoning path. In implementation, this preference is reflected through state-aware action priors and focus bookkeeping, rather than through a standalone confidence-thresholding gate. Consequently, the search remains semantically grounded while preserving the flexibility of stochastic exploration.

3.3. Adaptive Multi-Action Tree Search Process

The core inference engine of MAMCTS adheres to the MCTS paradigm but is fundamentally re-engineered for the high-dimensional visual state space. Unlike standard MCTS, our framework combines prior-guided stochastic action expansion with parallel simulations to accelerate convergence in the high-dimensional visual state space. The reasoning process iteratively executes four phases—Selection, Expansion, Simulation, and Backpropagation—thereby constructing a decision tree in which each path represents a candidate reasoning chain.

3.3.1. Selection via a PUCT-Style Policy

The search begins at the root node n_0 and recursively traverses down the tree to identify the most promising leaf node n_L . To balance the trade-off between exploiting high-value visual regions and exploring uncertain areas, we employ a modified prior-guided UCB (PUCT) policy tailored for multimodal reasoning. The policy selects the action a (equivalently, the corresponding child node) that maximizes the following composite objective (visually represented as the "Policy Model" in the left panel of Figure 2, which guides the branching direction). We formulate the node selection policy as a maximization problem of a composite objective function. The optimal action

a^* is selected to maximize:

$$a^* = \arg \max_{a \in \mathcal{A}(s)} (Q(s, a) + C \cdot U(s, a) + D(s, a) - \Psi(d)), \quad (8)$$

where $\mathcal{A}(s)$ denotes the set of admissible actions available at state s , and each term addresses a specific aspect of the decision-making trade-off:

1. $Q(s, a)$ (Exploitation Term): where $Q(s, a)$ is the mean action value, defined as

$$Q(s, a) = \frac{W(s, a)}{N(s, a)}, \quad (9)$$

where $W(s, a)$ denotes the cumulative reward collected by taking action a from state s , and $N(s, a)$ is the corresponding visit count. In implementation, $Q(s, a)$ is realized through the statistics of the child node reached by taking action a from state s , and is therefore estimated from node-level visit and reward counts.

2. $U(s, a)$ (Exploration Term): Following the PUCT algorithm, this term encourages the exploration of less-visited nodes. It is defined as:

$$U(s, a) = P(a|s) \cdot \frac{\sqrt{N(s)}}{1 + N(s, a)} \quad (10)$$

where $P(a|s)$ is the prior probability given by the semantic action priors (Section 3.3.2). In Eq. (8), C is the exploration constant (set to 1.5), which balances the trade-off between prior knowledge and visit frequency.

3. $D(s, a)$ (Diversity Term): To discourage the agent from falling into repetitive loops, we formulate $D(s, a)$ as a composite score of global uniqueness and local consistency:

$$D(s, a) = \lambda_1 \frac{|\mathcal{U}(\mathcal{H}_t \oplus a)|}{|\mathcal{H}_t \oplus a|} + \mathcal{S}_{\text{local}}(\mathcal{H}_t \oplus a). \quad (11)$$

where $\mathcal{H}_t$ denotes the action history up to step t and $\frac{|\mathcal{U}(\mathcal{H}_t \oplus a)|}{|\mathcal{H}_t \oplus a|}$ represents the global ratio of unique actions in the updated action sequence after taking candidate action a , weighted by $\lambda_1 = 0.15$. The term $\mathcal{S}_{\text{local}}$ imposes constraints on immediate transitions:

$$\begin{aligned} \mathcal{S}_{\text{local}}(\mathcal{H}_t \oplus a) = & \lambda_2 \mathbb{I}(a_t \neq a_{t-1}) - \lambda_3 \mathbb{I}(a_t = a_{t-1}) \\ & - \lambda_4 \mathbb{I}\left(\max_x \text{count}(x, \text{tail}_3(\mathcal{H}_t \oplus a)) \geq 2\right). \end{aligned} \quad (12)$$

Here, $\text{tail}_3(\mathcal{H}_t \oplus a)$ denotes the last three actions in the updated action sequence, and $\text{count}(x, \cdot)$ returns the number of occurrences of action x in that subsequence. Here, $\lambda_2 = 0.05$, $\lambda_3 = 0.10$, and $\lambda_4 = 0.15$. This structure rewards action switching while penalizing immediate repetitions and short-term loops.

4. $\Psi(d)$ (Depth Penalty): To prevent inefficient deep searches and encourage concise reasoning, we introduce a regularization term that penalizes depth d :

$$\Psi(d) = \gamma \cdot (1 - \rho^d) \quad (13)$$

where $\gamma = 0.05$ is the base penalty and $\rho = 0.92$ is the decay rate. This term grows as the tree depth increases, softly guiding the agent to find solutions at shallower levels.

3.3.2. Stochastic Single-Action Expansion with Semantic Priors

Unlike standard MCTS, our framework combines prior-guided stochastic action expansion with parallel simulation to accelerate convergence in the high-dimensional visual state space. Upon reaching a leaf node n_L , if the node is not terminal (i.e., depth $d < d_{\max}$), the algorithm initiates stochastic action expansion. To overcome the sparse-reward problem in visual search, we combine semantic priors with controlled stochasticity:

- **Action Prior Computation:** For the current state s , we compute a prior distribution $P(a | s)$ over the action space. This prior favors actions that are semantically aligned with the current reasoning state, such as targeting unverified objects or refining informative regions.
- **Dirichlet Noise Injection [34]:** To encourage exploration without perturbing the entire search process, Dirichlet noise is injected only at the root node when initializing or expanding from the root prior. Specifically, for the root state s_0 , the prior is perturbed as

$$P'(a | s_0) = (1 - \epsilon)P(a | s_0) + \epsilon\eta_a, \quad \eta \sim \text{Dir}(\alpha). \quad (14)$$

where $\epsilon = 0.12$ and $\alpha = 0.2$. This mechanism ensures a diverse set of visual

proposals are considered. For all non-root nodes, the original prior $P(a | s)$ is used without additional noise injection.

- **Single-Action Expansion:** A single action is then sampled from the node-specific prior distribution and executed to generate one child state s' . Specifically, root expansion samples from the perturbed prior $a \sim P'(a | s_0)$, whereas non-root expansion samples from the original prior $P(a | s)$. Parallelism is introduced at the simulation level by evaluating multiple search trajectories concurrently, rather than by expanding multiple children from a single leaf node in one step.

3.3.3. Simulation via Dual-Verification

In this phase, the newly generated child nodes are evaluated to estimate their potential value. A critical limitation in standard VQA reasoning is the "Self-Confirmation Bias," where models tend to output high confidence scores for non-existent objects when explicitly queried. To counteract this, we introduce a Dual-Verification Mechanism that enforces consistency between the model's unconditioned perception and its directed reasoning. The process consists of two steps:

1. **Spontaneous Perception:** First, we prompt the frozen LVLM to list all salient visible objects in the current cropped view without providing any specific target information (Prompt: "*List the main objects clearly visible...*"). Let L_{spn} denote the set of objects identified in this unconditioned generation. This step tests the model's unbiased visual grounding.
2. **Directed Confirmation:** Second, for each key object k under consideration, we explicitly query its presence using a standard Yes/No question to obtain a specific detection confidence score, denoted as $C_{\text{det}}(k)$. The resulting confidence score $C_{\text{det}}(k)$ is normalized to the interval $[0, 1]$.

An object k is considered valid if and only if it satisfies the consistency criterion in Eq. (15), where $Z(k) \in \{0, 1\}$ denotes the object-validity indicator.

$$Z(k) = \mathbb{I}(C_{\text{det}}(k) > \tau_{\text{conf}}) \cdot \mathbb{I}(k \in L_{\text{spn}}), \quad (15)$$

where τ_{conf} is the confidence threshold (set to 0.6) and $\mathbb{I}(\cdot)$ is the indicator function. By requiring the target to be both "spontaneously seen" and "confidently confirmed,"

this mechanism effectively filters out hallucinated responses induced by query biases. The validity indicators $Z(k)$ and confidence scores $C_{det}(k)$ are subsequently passed to the reward formulation in Section 3.4.1.

3.3.4. Backpropagation

Once the simulation reward r is obtained for the expanded child node, it is back-propagated along the selected path to update the statistics of all ancestor nodes. For each node n on the path, we update

$$N(n) \leftarrow N(n) + 1, \quad W(n) \leftarrow W(n) + r, \quad Q(n) \leftarrow \frac{W(n)}{N(n)}. \quad (16)$$

Here, $W(n)$ denotes the cumulative reward of node n , while $Q(n)$ denotes its mean value. When parallel simulations are used, these updates are applied independently to each completed trajectory. These updates refine the exploitation term in Eq. (8) and guide subsequent simulations toward regions with consistently high reward.

3.4. Optimization: Reward Design and Dynamic Pruning

To ensure the MAMCTS framework converges efficiently toward semantically grounded regions while mitigating the risk of hallucinations inherent in stochastic search, we introduce two critical optimization mechanisms: a Hallucination-Aware Reward Function and a Dynamic Pruning Strategy.

3.4.1. Dual-Verification Reward Mechanism

The reward function serves as the objective signal guiding the MCTS policy. Instead of relying solely on raw detection confidence, which is prone to hallucinations, we construct a tiered reward based on the Dual-Verification results derived in the Simulation phase (Section 3.3.3).

Utilizing the validity indicator $Z(k)$ and detection confidence $C_{det}(k)$ obtained from Eq. (15), we formulate the reward $R(s)$ to strictly penalize partial matches or hallucinations. The final reward is structured as follows:

$$R(s) = \begin{cases} 1.0 + \bar{C}_{det}, & \text{if } \sum_{k \in \mathcal{K}} Z(k) = |\mathcal{K}|, \\ 0.3 \cdot \frac{1}{|\mathcal{K}|} \sum_{k \in \mathcal{K}} Z(k), & \text{otherwise.} \end{cases} \quad (17)$$

where

$$\bar{C}_{\text{det}} = \frac{1}{|\mathcal{K}|} \sum_{k \in \mathcal{K}} C_{\text{det}}(k) \quad (18)$$

denotes the average detection confidence across all queried key objects. In the actual implementation, crop-size regularization is not modeled as an additive soft penalty. Instead, it is enforced as a hard geometric validity check before reward evaluation. Specifically, during Adaptive_Crop, a proposed crop is accepted only if it satisfies the minimum geometric constraint; otherwise, the crop is rejected and the framework falls back to the Zoom_Out action to recover broader visual context.

Here, $|\mathcal{K}|$ denotes the total number of key objects. If all objects are verified (Full Consistency), the node receives a high reward (> 1.0), boosted by the average confidence and computed only for geometrically valid crops, while invalid proposals are rejected before reward evaluation. Conversely, if even a single object fails the consistency check (indicating potential hallucination or occlusion), the reward collapses to a significantly lower tier (scaled by 0.3). This strict tiered reward makes ambiguous branches substantially less competitive and biases the search toward regions supported by cross-verified visual evidence. Subsequent dynamic pruning further removes persistently low-value branches.

To mitigate the inherent stochasticity of LVLM generation, a robust fallback strategy is implemented: should the model fail to yield valid JSON coordinates or if the proposed region is negligible in size, the system automatically triggers $\mathcal{A}_{\text{zoom}}$. This mechanism functions as an exception handler, preventing the search process from collapsing into invalid states and facilitating context recovery through a broadened field of view.

3.4.2. Dynamic Pruning Strategy

Given the high branching factor of the visual action space, the search tree can grow exponentially. To maintain computational tractability and filter out low-value branches (i.e., paths focusing on background noise), we implement a Dynamic Pruning mechanism.

We define a pruning criterion based on the relative value of sibling nodes. Let $\mathcal{S}_{\text{parent}}$ be the set of all child nodes generated from a common parent. A node $n_i \in$

S_{parent} is pruned if its estimated value $Q(n_i)$ falls significantly below the group average:

$$\text{Prune}(n_i) \iff Q(n_i) < \epsilon_{\text{prune}} \cdot \frac{1}{|S_{\text{parent}}|} \sum_{n_j \in S_{\text{parent}}} Q(n_j). \quad (19)$$

where ϵ_{prune} is the pruning sensitivity threshold, empirically set to 0.12. This "Survival of the Fittest" strategy ensures that computational resources are reallocated from low-promise trajectories likely caused by misrecognition or irrelevant crops to high-confidence reasoning paths. This dynamic pruning is triggered periodically (e.g., every 5 iterations) to allow sufficient accumulation of statistics before making irreversible pruning decisions.

4. Experimental Results and Analysis

This section first introduces the evaluation methodology and experimental setup, including the POPE-derived benchmark, evaluation metrics, baseline LVLMs, and implementation details. It then reports the principal results on hallucination suppression and generalization across backbone architectures and model scales. Finally, it summarizes the extended efficiency, hyperparameter-sensitivity, ROC, ablation, qualitative, and comparative analyses, whose complete results are provided in Supplementary Sections S1–S4.

4.1. Evaluation Methodology

This section outlines the experimental framework designed to validate the robustness of MAMCTS. We specifically detail the composition of the evaluation benchmarks and the quantitative metrics utilized to assess the model’s performance across diverse visual reasoning tasks.

4.1.1. Datasets and Benchmarks

To rigorously evaluate the efficacy of the proposed MAMCTS framework in mitigating object hallucinations, we established a comprehensive evaluation benchmark comprising six specialized datasets derived from Microsoft COCO(MSCOCO). Adhering to the standardized protocol of the Polling-based Object Probing Evaluation

(POPE) [10], we constructed 36,000 visual question-answering samples to probe the model’s discriminative capability between existent and non-existent objects.

The benchmark spans two distinct image domains: Real-World scenarios drawn from the MSCOCO validation set and abstract scenes utilizing scene-graph-style imagery [35]. The evaluation is stratified across three progressive negative sampling strategies—Random, Popular and Adversarial—resulting in six distinct experimental settings (2 domains $\times$ 3 strategies).

For each setting, we utilized 1,000 images. To ensure a robust and statistically significant evaluation, each image is paired with 6 binary (Yes/No) queries, consisting of 3 positive samples (ground-truth objects confirmed via annotations) and 3 negative samples (non-existent objects). This design strictly maintains a balanced 1:1 positive-to-negative ratio across the entire benchmark, yielding 6,000 samples per dataset and 36,000 samples in total.

The negative samples were generated using three strategies designed to test varying degrees of hallucination robustness:

- **Random Sampling:** Negative objects are randomly selected from the entire dataset vocabulary, serving as a baseline for general non-existence detection.
- **Popular Sampling:** Negative objects are drawn from categories with high global frequency in the dataset but are absent in the current image. This strategy evaluates the model’s resistance to prior bias towards common objects.
- **Adversarial Sampling:** Following the POPE adversarial paradigm, negative objects are selected based on high co-occurrence probability with the ground-truth objects present in the image, targeting the well-known language prior bias in VQA models [36]. These candidates are semantically plausible but visually absent, posing the most significant challenge to the model’s visual grounding capabilities.

This multi-tiered dataset structure facilitates a granular assessment of the MAM-CTS method across varying levels of scene complexity and adversarial interference.

4.1.2. Evaluation Metrics

To strictly assess the model's performance in object existence prediction and its robustness against hallucinations, we adopt a multi-dimensional evaluation protocol aligned with the POPE benchmark. We define the model's response of "Yes" (object exists) as the Positive class and "No" (object does not exist) as the Negative class. Accordingly, we denote TP (True Positive) and TN (True Negative) as correct predictions, while FP (False Positive) and FN (False Negative) represent errors where the model hallucinates a non-existent object or misses an existing one, respectively.

The specific metrics are formulated as follows:

- **Accuracy:** This metric provides a holistic view of the model's discriminative capability across both real and nonexistent objects. Given the balanced 1:1 ratio of positive to negative samples in our datasets, accuracy serves as an unbiased indicator of overall performance:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (20)$$

- **Precision and Recall:** To offer a granular analysis of the model's grounding ability, we report Precision (the proportion of valid detections among all positive responses) and Recall (the proportion of actual objects successfully detected):

$$Precision = \frac{TP}{TP + FP}, \quad Recall = \frac{TP}{TP + FN} \quad (21)$$

- **F1 Score:** As the harmonic mean of Precision and Recall, the F1 Score evaluates the trade-off between avoiding hallucinations and ensuring object coverage, preventing the model from achieving high scores via trivial strategies (e.g., always predicting "Yes"):

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (22)$$

- **False Positive Rate (FPR):** This is the critical metric for quantifying object hallucination. It measures the probability that the model incorrectly identifies a non-existent object as present. A lower FPR explicitly indicates superior resistance to hallucination and reduced fabrication of visual content:

$$FPR = \frac{FP}{FP + TN} \quad (23)$$

By integrating these metrics, we aim to demonstrate that the proposed MAMCTS not only improves general accuracy but specifically targets the suppression of false positives (hallucinations) without compromising the recall of actual visual details.

4.2. Experimental Setup

To ensure the reproducibility of our results and provide a comprehensive assessment of the MAMCTS framework, we detail the baseline selection and implementation specifics below.

4.2.1. Baseline Models

To verify the generality and training-free transferability of MAMCTS, we selected two representative Large Vision-Language Models (LVLMs) that span distinct visual processing paradigms. Our selection criteria focus on covering both the classic fixed-resolution architecture and the state-of-the-art dynamic-resolution architecture:

1. LLaVA-v1.5-7B [3]: representing the canonical fixed-resolution paradigm. It utilizes a CLIP-ViT-L-336px encoder connected to the Vicuna LLM via a two-layer MLP. As a widely adopted open-source benchmark, it serves as a robust baseline to evaluate how MAMCTS enhances models with standard visual perception capabilities.
2. Qwen2-VL-7B [6]: representing the state-of-the-art dynamic-resolution paradigm. Unlike LLaVA, Qwen2-VL employs the Naive Dynamic Resolution mechanism, allowing it to process images of arbitrary aspect ratios and resolutions. Including this model challenges our framework to demonstrate improvements even atop a backbone that already possesses sophisticated fine-grained visual processing capabilities.

For all baseline models, using their 7B versions unless otherwise specified, we utilized the official pre-trained weights without any additional fine-tuning or weight modification. This strict zero-shot setting ensures that all performance gains are derived exclusively from maximizing the potential of the frozen models via the MAMCTS inference mechanism. Furthermore, to evaluate the transferability of MAMCTS across both backbone architectures and model sizes, we extended our experiments beyond the

two 7B baselines to include two lightweight 4B models, namely Gemma-3-4b-it and InternVL2.5-4B, under the same frozen zero-shot protocol.

4.2.2. Implementation Details

All experiments were conducted on a high-performance computing cluster equipped with NVIDIA L40s GPUs (48GB VRAM). The framework was implemented using PyTorch and the HuggingFace Transformers library.

(1) MAMCTS Hyperparameter Configuration: The core parameters of the MAMCTS framework were empirically tuned to balance exploration efficiency (inference latency) and reasoning depth (accuracy). These parameters were kept fixed in the main benchmark experiments unless otherwise stated:

- Tree Search Specifications: We set the maximum search depth (d_{max}) to 4, allowing for sufficient multi-step reasoning without combinatorial explosion. The simulation budget was capped at 24 iterations per query. To accelerate inference, we implemented a parallel simulation strategy with a batch size of 3, allowing concurrent evaluation of multiple search trajectories.
- Exploration-Exploitation Balance: In the expansion phase, we utilized Dirichlet noise with a concentration parameter $\alpha = 0.2$ and a mixing weight of 0.12. This specific configuration encourages the exploration of diverse visual paths while maintaining adherence to the action priors derived from the semantic context.
- Dynamic Pruning Strategy: To optimize memory usage and search speed, we employed a relative threshold-based pruning mechanism. Nodes with a value lower than 12% of their siblings' average value were pruned ($threshold < 0.12 \times avg_sibling_value$). This ensures that computational resources are concentrated solely on high-potential reasoning trajectories.
- Crop-Area Robustness and Fallback: In the current implementation, crop-size regularization is enforced through a hard geometric constraint rather than an additive scalar penalty. Specifically, during Adaptive_Crop, a proposed crop is rejected if either its width or height is smaller than 10% of the current view dimensions. In such cases, the framework falls back to the Zoom_Out action to

recover broader context. This corresponds to an effective lower bound of approximately 0.01 on the valid crop-area ratio.

Table 2: Performance of baseline LVLMs and their MAMCTS-enhanced versions on POPE-derived benchmarks across Real-World and abstract domains. Best results within each setting are bolded.

Dataset	Data Characteristic	POPE	Model	Method	Accuracy	Precision	Recall	F1 Score	FPR(%)
MSCOCO	Real-World Image Datasets	Random	LLaVA-v1.5	Original	78.47	73.45	89.17	80.55	32.23
				Ours	83.60	86.98	79.03	82.82	11.83
			Qwen2-VL	Original	86.27	89.97	81.63	85.60	9.10
				Ours	86.80	92.69	79.90	85.82	6.30
		Popular	LLaVA-v1.5	Original	85.60	82.84	89.80	86.18	18.60
				Ours	85.90	91.97	78.67	84.80	6.87
			Qwen2-VL	Original	88.00	94.36	80.36	87.07	4.83
				Ours	88.35	95.78	80.23	87.32	3.53
		Adversarial	LLaVA-v1.5	Original	78.40	73.11	89.83	80.62	33.03
				Ours	82.72	85.65	78.60	81.97	13.17
			Qwen2-VL	Original	86.18	90.04	80.97	85.42	8.60
				Ours	87.03	92.83	80.27	86.09	6.20
	Synthetic Image Datasets	Random	LLaVA-v1.5	Original	80.97	77.02	88.27	82.26	26.33
				Ours	83.82	86.96	79.57	83.10	11.93
			Qwen2-VL	Original	93.15	89.31	98.03	93.47	10.57
				Ours	94.12	91.09	97.80	94.33	9.57
		Popular	LLaVA-v1.5	Original	74.58	69.30	88.27	77.64	39.10
				Ours	80.47	81.03	79.57	80.29	18.63
			Qwen2-VL	Original	92.10	87.63	98.03	92.54	13.83
				Ours	92.70	88.72	97.83	93.06	12.43
		Adversarial	LLaVA-v1.5	Original	66.13	61.18	88.27	72.27	56.00
				Ours	75.92	74.15	79.57	76.76	27.73
			Qwen2-VL	Original	89.02	83.07	98.00	89.92	19.97
				Ours	89.35	83.68	97.77	90.18	19.07

4.3. Quantitative Results and Performance Analysis

This section evaluates the MAMCTS framework across three core dimensions: (i) Predictive Performance, focusing on accuracy and hallucination suppression (FPR); (ii) Model Generalizability, assessing training-free compatibility across varying architectures and parameter scales, while in a few settings it trades some recall-driven accuracy for markedly better precision and FPR; and (iii) Operational Efficiency, analyzing the accuracy-latency trade-off. These experiments aim to substantiate the framework’s capacity to enhance visual grounding and decision reliability in frozen Large Vision-Language Models.

4.3.1. Performance on Standard POPE Benchmarks

We first evaluate the impact of MAMCTS on two representative baselines, LLaVA-v1.5 (fixed-resolution) and Qwen2-VL (dynamic-resolution), across six POPE-derived datasets. Table 2 summarizes the comparative results.

Empirical evaluations demonstrate that MAMCTS consistently surpasses original baselines across all sampling strategies, with the most profound gains manifested in the Adversarial Sampling setting—the most rigorous testbed for object hallucination. Specifically, on Real-World adversarial datasets, the framework enhances the performance of the fixed-resolution LLaVA-v1.5 by boosting accuracy from 78.40% to 82.72% (+4.32%) and drastically curtailing the False Positive Rate (FPR) from 33.03% to 13.17%. This equates to a relative reduction of nearly 60%, indicating that the active search mechanism effectively rectifies the high hallucination propensity inherent in standard encoders. Furthermore, when applied to the state-of-the-art Qwen2-VL, MAMCTS yields incremental gains, elevating accuracy to 87.03% and suppressing FPR to 6.20% in adversarial settings. These results substantiate that MAMCTS functions not merely as a compensatory module for visual deficiencies, but as an orthogonal reasoning enhancement that augments even advanced dynamic-resolution architectures.

The robustness of the proposed framework is further elucidated by the performance disparity observed in the Abstract (Synthetic) domain, a setting where models typically falter due to the paucity of rich textural cues. In this challenging context, MAMCTS delivers its most substantial improvements: for LLaVA-v1.5 on abstract adversarial samples, it achieves a remarkable accuracy increase of 9.79% (rising from 66.13% to 75.92%) while simultaneously suppressing the FPR by a significant margin of 28.27% (declining from 56.00% to 27.73%). This stark contrast underscores the framework’s capability to systematically inhibit the fabrication of non-existent objects, thereby ensuring reliable decision-making in visually abstract or unfamiliar scenes where baseline models inevitably exhibit performance degradation.

Table 3: Performance of MAMCTS on lightweight LLM backbones (Gemma-3-4B and InternVL-2.5-4B) on POPE benchmarks. Best results within each setting are bolded.

Dataset	Data Characteristic	POPE	Model	Method	Accuracy	Precision	Recall	F1 Score	FPR(%)
MSCOCO	Real-World Image Datasets	Random	InternVL-2.5	Original	84.60	81.02	90.37	85.44	21.17
				Ours	85.88	95.87	75.00	84.16	3.23
			Gemma-3	Original	79.25	75.56	86.47	80.65	27.97
				Ours	82.05	80.36	84.83	82.54	20.73
		Popular	InternVL-2.5	Original	89.28	88.23	90.67	89.43	12.10
				Ours	87.15	97.94	75.90	85.52	1.60
			Gemma-3	Original	80.95	76.83	88.63	82.31	26.73
				Ours	84.07	82.65	86.23	84.40	18.10
		Adversarial	InternVL-2.5	Original	84.85	81.18	90.73	85.69	21.03
				Ours	86.42	96.12	75.90	84.82	3.07
			Gemma-3	Original	77.88	73.03	88.43	79.99	32.67
				Ours	81.43	78.70	86.20	82.28	23.33
	Synthetic Image Datasets	Random	InternVL-2.5	Original	92.87	88.50	98.53	93.25	12.80
				Ours	93.38	92.30	94.67	93.47	7.90
			Gemma-3	Original	89.40	83.05	99.00	90.33	20.20
				Ours	90.42	84.57	98.87	91.16	18.03
		Popular	InternVL-2.5	Original	92.33	87.66	98.53	92.78	13.87
				Ours	92.72	91.03	94.77	92.86	9.33
			Gemma-3	Original	89.23	82.82	99.00	90.19	20.53
				Ours	89.72	83.57	98.87	90.58	19.43
		Adversarial	InternVL-2.5	Original	87.03	80.11	98.53	88.37	24.47
				Ours	88.07	83.51	94.87	88.83	18.73
			Gemma-3	Original	83.00	75.00	99.00	85.34	33.00
				Ours	83.16	75.22	98.87	85.44	32.57

4.3.2. Generalizability Across Model Scales

Together with the 7B results in Table 2, the 4B experiments in Table 3 demonstrate transfer across model scales and backbone architectures. For Gemma-3 on the Real-World Adversarial split, MAMCTS raises accuracy from 77.88% to 81.43% and reduces FPR from 32.67% to 23.33%. For InternVL-2.5-4B, it reduces Real-World FPR from approximately 21% to 3.23% on Random sampling and 3.07% on Adversarial sampling. The Real-World Popular split exhibits a conservative accuracy–precision trade-off: accuracy decreases from 89.28% to 87.15%, while precision reaches 97.94% and FPR falls to 1.60%. Thus, the most consistent cross-scale benefit is stronger false-positive suppression without parameter fine-tuning.

4.3.3. Additional Robustness and Comparative Analyses

Supplementary Sections S1–S3 report efficiency, sensitivity, ablation, and qualitative analyses. MAMCTS raises Real-World Adversarial accuracy from 78.40% to

82.72%, yields an ROC AUC of 0.94, and remains stable across the tested search budgets and depths. Its average latency increases from approximately 0.08 s to 0.9 s per sample. Ablations confirm contributions from search, dual verification, and ensemble aggregation.

Supplementary Section S4 compares MAMCTS with DyFo [25] on GQA and A-OKVQA [37], and with Osprey [38], Ferret [39], Shikra [40], InstructBLIP [5], MiniGPT-4 [41], and mPLUG-Owl [42]. MAMCTS achieves the highest POPE-COCO accuracy on Popular and Adversarial sampling and the highest precision across all three settings.

5. Conclusion

This study introduced MAMCTS as a training-free approach to reducing object hallucination in Visual Question Answering through active, object-centric tree search. Its principal strength is that it changes the inference procedure without updating the parameters of the underlying vision-language model. The same search wrapper can therefore be applied to frozen backbones of different scales, while its explicit action space, dual-verification reward, dynamic pruning, and answer aggregation provide interpretable points at which visual evidence is proposed, checked, and combined. Across the evaluated POPE-derived real-world and synthetic benchmarks, as well as GQA, A-OKVQA, and POPE-COCO, this design consistently reduced false-positive object predictions and produced its largest gains in adversarial or visually ambiguous settings. For Pattern Recognition readers, the framework offers a reusable mechanism for testing whether multimodal decisions remain tied to observable visual patterns, and a modular basis for studying how search, grounding, and confidence estimation interact under misleading evidence.

These benefits are accompanied by several limitations. First, multi-step tree expansion requires more model calls and greater inference latency than one-pass prediction, which may restrict real-time or resource-constrained deployment. Second, the search is guided by confidence estimates from a frozen evaluator and by the quality of its object-focused visual states; poorly calibrated confidence or an uninformative crop can

therefore propagate through the search. Third, the present experiments focus mainly on object-existence hallucination in static-image VQA. The more task-dependent results on A-OKVQA also indicate that visual search alone cannot fully address errors that arise from missing external knowledge or broader semantic reasoning. Consequently, the current evidence should not be interpreted as covering every form of multimodal hallucination.

Future work will develop adaptive search budgets, early-exit criteria, cached evaluations, and batched simulations to reduce computational cost while retaining verification quality. It will also investigate explicit uncertainty calibration, more diverse open-ended VQA settings, and spatiotemporal search for video understanding. Broader failure analysis across domains and model families will help determine when object-centric search is most beneficial and when it should be combined with external knowledge or specialized visual tools. Overall, the results support inference-time search as a practical complement to pretrained vision-language models, while also identifying efficiency, calibration, and task coverage as the main directions required for wider use.

References

- [1] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, D. Parikh, VQA: Visual question answering, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 2425–2433.
- [2] D. A. Hudson, C. D. Manning, GQA: A new dataset for real-world visual reasoning and compositional question answering, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 6700–6709.
- [3] H. Liu, C. Li, Q. Wu, Y. J. Lee, Visual instruction tuning, *Advances in neural information processing systems* 36 (2023) 34892–34916.
- [4] H. Liu, C. Li, Y. Li, Y. J. Lee, Improved baselines with visual instruction tuning, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2024, pp. 26296–26306.

- [5] W. Dai, J. Li, D. Li, A. Tiong, J. Zhao, W. Wang, B. Li, P. N. Fung, S. Hoi, InstructBLIP: Towards general-purpose vision-language models with instruction tuning, *Advances in neural information processing systems* 36 (2023) 49250–49267.
- [6] P. Wang, S. Bai, S. Tan, S. Wang, Z. Fan, J. Bai, K. Chen, X. Liu, J. Wang, W. Ge, et al., Qwen2-VL: Enhancing vision-language model’s perception of the world at any resolution, *arXiv preprint arXiv:2409.12191* (2024).
- [7] Z. Chen, J. Wu, W. Wang, W. Su, G. Chen, S. Xing, M. Zhong, Q. Zhang, X. Zhu, L. Lu, et al., InternVL: Scaling up vision foundation models and aligning for generic visual-linguistic tasks, in: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 24185–24198.
- [8] S. Hao, Y. Gu, H. Ma, J. Hong, Z. Wang, D. Wang, Z. Hu, Reasoning with language model is planning with world model, in: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023, pp. 8154–8173.
- [9] Z. Zhang, A. Zhang, M. Li, H. Zhao, G. Karypis, A. Smola, Multimodal chain-of-thought reasoning in language models, *Transactions on Machine Learning Research* (2024).
- [10] Y. Li, Y. Du, K. Zhou, J. Wang, W. X. Zhao, J.-R. Wen, Evaluating object hallucination in large vision-language models, in: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2023, pp. 292–305. doi:10.18653/v1/2023.emnlp-main.20.
- [11] H. Liu, W. Xue, Y. Chen, D. Chen, X. Zhao, K. Wang, L. Hou, R. Li, W. Peng, A survey on hallucination in large vision-language models, *arXiv preprint arXiv:2402.00253* (2024).
- [12] Y. Shi, S. Di, Q. Chen, W. Xie, Enhancing Video-LLM reasoning via agent-of-

- thoughts distillation, in: Proceedings of the Computer Vision and Pattern Recognition Conference, 2025, pp. 8523–8533.
- [13] Z. Zheng, M. Yang, J. Hong, C. Zhao, G. Xu, L. Yang, C. Shen, X. Yu, Deep-Eyes: Incentivizing “thinking with images” via reinforcement learning, in: The Fourteenth International Conference on Learning Representations, 2026.
URL <https://openreview.net/forum?id=nVf2TQFuty>
- [14] D. Peng, Z. Li, Unbiased VQA via modal information interaction and question transformation, *Pattern Recognition* 162 (2025) 111394.
- [15] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, S. Colton, A survey of monte carlo tree search methods, *IEEE Transactions on Computational Intelligence and AI in Games* 4 (1) (2012) 1–43. doi:10.1109/TCIAIG.2012.2186810.
- [16] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al., Mastering the game of go with deep neural networks and tree search, *nature* 529 (7587) (2016) 484–489.
- [17] S. Yao, D. Yu, J. Zhao, I. Shafran, T. Griffiths, Y. Cao, K. Narasimhan, Tree of thoughts: Deliberate problem solving with large language models, *Advances in neural information processing systems* 36 (2023) 11809–11822.
- [18] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. Shamma, M. Bernstein, L. Fei-Fei, Image retrieval using scene graphs, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3668–3678.
- [19] R. Zellers, M. Yatskar, S. Thomson, Y. Choi, Neural motifs: Scene graph parsing with global context, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 5831–5840.
- [20] R. Koner, H. Li, M. Hildebrandt, D. Das, V. Tresp, S. Günnemann, GraphHopper: Multi-hop scene graph reasoning for visual question answering, in: International Semantic Web Conference, Springer, 2021, pp. 111–127.

- [21] Y. Wang, M. Yasunaga, H. Ren, S. Wada, J. Leskovec, VQA-GNN: Reasoning with multimodal knowledge via graph neural networks for visual question answering, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 21525–21535. doi:10.1109/ICCV51070.2023.01973.
- [22] K. Yi, J. Wu, C. Gan, A. Torralba, P. Kohli, J. Tenenbaum, Neural-symbolic VQA: Disentangling reasoning from vision and language understanding, *Advances in neural information processing systems* 31 (2018).
- [23] X. Ke, P. Xu, W. Guo, Language–image consistency augmentation and distillation network for visual grounding, *Pattern Recognition* 166 (2025) 111663. doi:10.1016/j.patcog.2025.111663.
- [24] S. A. M. Mohamud, A. Jalali, M. Lee, Encoder–decoder cycle for visual question answering based on perception–action cycle, *Pattern Recognition* 144 (2023) 109848. doi:10.1016/j.patcog.2023.109848.
- [25] G. Li, J. Xu, Y. Zhao, Y. Peng, DyFo: A training-free dynamic focus visual search for enhancing LMMs in fine-grained visual understanding, in: Proceedings of the Computer Vision and Pattern Recognition Conference, 2025, pp. 9098–9108.
- [26] A. A. Yusuf, C. Feng, X. Mao, X. Li, Y. Haruna, R. A. Duma, Multi-scale dual-stream visual feature extraction and graph reasoning for visual question answering, *Applied Intelligence* 55 (6) (2025) 1–18.
- [27] P. Lu, B. Peng, H. Cheng, M. Galley, K.-W. Chang, Y. N. Wu, S.-C. Zhu, J. Gao, Chameleon: Plug-and-play compositional reasoning with large language models, *Advances in Neural Information Processing Systems* 36 (2023) 43447–43478.
- [28] Y. Shen, K. Song, X. Tan, D. Li, W. Lu, Y. Zhuang, HuggingGPT: Solving AI tasks with ChatGPT and its friends in hugging face, *Advances in Neural Information Processing Systems* 36 (2023) 38154–38180.
- [29] S. Yin, C. Fu, S. Zhao, T. Xu, H. Wang, D. Sui, Y. Shen, K. Li, X. Sun, E. Chen, Woodpecker: Hallucination correction for multimodal large language models, *Science China Information Sciences* 67 (12) (2024) 220105.

- [30] B. Lin, Z. Tang, Y. Ye, J. Cui, B. Zhu, P. Jin, J. Huang, J. Zhang, Y. Pang, M. Ning, et al., MoE-LLaVA: Mixture of experts for large vision-language models, *IEEE Transactions on Multimedia* 28 (2026) 4408–4419. doi:10.1109/TMM.2026.3654458.
- [31] N. Jegham, M. Abdelatti, A. Hendawi, Visual reasoning consistency and robustness analysis of multimodal LLMs, *Pattern Recognition* 172 (2026) 112765. doi:10.1016/j.patcog.2025.112765.
- [32] A. Zhou, K. Yan, M. Shlapentokh-Rothman, H. Wang, Y.-X. Wang, Language agent tree search unifies reasoning acting and planning in language models, in: *Proceedings of the 41st International Conference on Machine Learning*, Vol. 235 of *Proceedings of Machine Learning Research*, PMLR, 2024, pp. 62138–62160.
- [33] V. Mnih, N. Heess, A. Graves, K. Kavukcuoglu, Recurrent models of visual attention, *Advances in neural information processing systems* 27 (2014).
- [34] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, et al., Mastering the game of go without human knowledge, *nature* 550 (7676) (2017) 354–359.
- [35] B. S. Kim, J. Kim, D. Lee, B. Jang, Visual question answering: A survey of methods, datasets, evaluation, and challenges, *ACM Computing Surveys* 57 (10) (2025) 1–35.
- [36] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, D. Parikh, Making the V in VQA matter: Elevating the role of image understanding in visual question answering, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 6904–6913.
- [37] D. Schwenk, A. Khandelwal, C. Clark, K. Marino, R. Mottaghi, A-OKVQA: A benchmark for visual question answering using world knowledge, in: *European conference on computer vision*, Springer, 2022, pp. 146–162.
- [38] Y. Yuan, W. Li, J. Liu, D. Tang, X. Luo, C. Qin, L. Zhang, J. Zhu, Osprey: Pixel understanding with visual instruction tuning, in: *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition, 2024, pp. 28202–28211.

- [39] H. You, H. Zhang, Z. Gan, X. Du, B. Zhang, Z. Wang, L. Cao, S.-F. Chang, Y. Yang, Ferret: Refer and ground anything anywhere at any granularity, in: The Twelfth International Conference on Learning Representations, 2024.
- [40] K. Chen, Z. Zhang, W. Zeng, R. Zhang, F. Zhu, R. Zhao, Shikra: Unleashing multimodal LLM’s referential dialogue magic, arXiv preprint arXiv:2306.15195 (2023).
- [41] D. Zhu, J. Chen, X. Shen, X. Li, M. Elhoseiny, MiniGPT-4: Enhancing vision-language understanding with advanced large language models, in: The Twelfth International Conference on Learning Representations, 2024.
- [42] Q. Ye, H. Xu, G. Xu, J. Ye, M. Yan, Y. Zhou, J. Wang, A. Hu, P. Shi, Y. Shi, et al., mPLUG-Owl: Modularization empowers large language models with multimodality, arXiv preprint arXiv:2304.14178 (2023).