Grounded Reinforcement Learning for Visual Reasoning

Gabriel Sarch **Snigdha Saha** **Naitik Khandelwal** **Ayush Jain**
Michael J. Tarr **Aviral Kumar** **Katerina Fragkiadaki**
Carnegie Mellon University
visually-grounded-rl.github.io

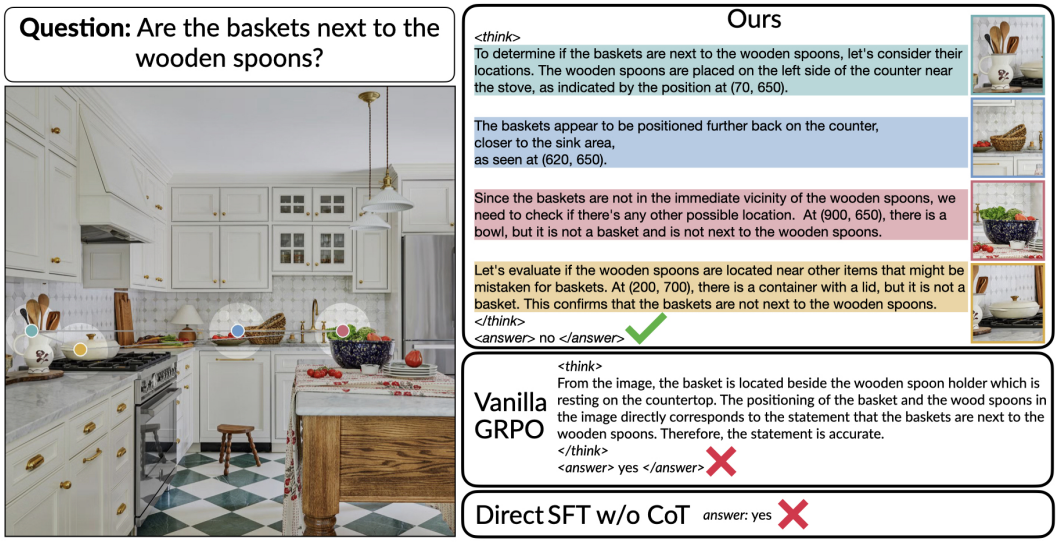


Figure 1: **Grounded visual reasoning enables interpretable and accurate answers.** ViGoRL decomposes the task into a sequence of natural language thoughts anchored in image regions. In contrast, Vanilla GRPO and SFT baselines produce ungrounded and incorrect responses.

Abstract

While reinforcement learning (RL) over chains of thought has significantly advanced language models in tasks such as mathematics and coding, visual reasoning introduces added complexity by requiring models to direct visual attention, interpret perceptual inputs, and ground abstract reasoning in spatial evidence. We introduce ViGoRL (**Vi**sually **Gro**unded **R**einforcement **L**earning), a vision-language model trained with RL to explicitly anchor each reasoning step to specific visual coordinates. Inspired by human visual decision-making, ViGoRL learns to produce spatially grounded reasoning traces, guiding visual attention to task-relevant regions at each step. When fine-grained exploration is required, our novel multi-turn RL framework enables the model to dynamically zoom into predicted coordinates as reasoning unfolds. Across a diverse set of visual reasoning benchmarks—including SAT-2 and BLINK for spatial reasoning, V*bench for visual search, and ScreenSpot and VisualWebArena for web-based grounding—ViGoRL consistently outperforms both supervised fine-tuning and conventional RL baselines that lack explicit grounding mechanisms. Incorporating multi-turn RL with zoomed-in visual feedback significantly improves ViGoRL's performance on localizing small GUI elements and visual search, achieving 86.4% on V*Bench. Additionally, we find that grounding amplifies other visual behaviors such as region exploration, grounded subgoal setting, and visual verification. Finally, human evaluations show that the model's visual references are not only spatially accurate but also helpful for understanding model reasoning steps. Our results show that visually grounded RL is a strong paradigm for imbuing models with general-purpose visual reasoning.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

1 Introduction

Visual reasoning tasks vary widely in structure and often demand different solution strategies depending on the problem at hand. Some tasks are dominated by salient visual cues, such as recognizing a refrigerator centered in a kitchen scene, while others, like locating a pair of scissors in a cluttered environment, require sequential visual search and selective attention. Despite this diversity, most state-of-the-art vision-language models (VLMs) operate in an end-to-end fashion, predicting answers directly in a single forward pass. These models often lack the ability to adapt their computational strategies to different tasks or to expose intermediate reasoning beyond visual attention maps. Prompt-based models, such as ViperGPT [56], VisualProg [21], and V* [69], explicitly decompose visual tasks into sequences of subgoals or intermediate steps to improve interpretability and performance without additional training. However, such models typically generate fixed reasoning chains that do not adapt to the structure of the input scene.

Recent advances in reinforcement learning (RL) over reasoning chains have significantly enhanced the capabilities of LLMs in text-based domains [19, 60, 28], enabling them to learn diverse reasoning strategies tailored to the context. However, RL can only build upon skills or compose reasoning behaviors that are already latent in the base model's sampling distribution [15, 84]. For e.g., Gandhi et al. [15] has identified key cognitive behaviors in text-based domains, such as setting sub-goals, backtracking, verification, that support self-improvement under RL. Models lacking these behaviors often do not benefit from RL and must be bootstrapped via supervised fine-tuning (SFT) on curated reasoning traces before RL is run [15, 84]. However, it remains unclear whether the cognitive behaviors identified in text-based domains similarly support generalization in visual reasoning tasks.

Several recent works have attempted RL fine-tuning directly on base vision-language models (VLMs) [40, 39, 89, 57, 59, 86, 53, 43, 34], implicitly assuming RL alone can induce useful cognitive behaviors. However, our analysis reveals that such naïve applications of RL typically yield abstract, ungrounded reasoning rather than richer, visually grounded cognitive behaviors (see Section 3.1, 5.3). These findings align with prior research showing that explicitly prompting VLMs to reference spatial object locations improves performance and interpretability [69, 75, 16], suggesting that grounding thought in spatial regions may serve as a key cognitive behavior for effective visual reasoning. Thus, a critical open question arises: *How can we embed useful cognitive behaviors in VLMs before applying RL to achieve robust visual reasoning?*

We hypothesize that models both "see better" and "think better" when their textual reasoning steps are explicitly grounded in specific image regions, promoting more targeted and systematic cross-referencing between textual and visual information during reasoning. This hypothesis is inspired by the fact that humans systematically shift their restricted gaze to selectively gather and integrate task-relevant information when reasoning about the world [77, 24, 76]. Grounding may serve a similar role in models, functioning as a spatial attention mechanism that enables accurate feature binding [63, 5, 6] and supports deictic reference [4] to simplify multi-step reasoning through localized perceptual anchoring. We move beyond prompt-based reasoning, proposing that learning to compose *reasoning steps explicitly anchored in image coordinates* induces structured region-level behaviors that support improved generalization in visual tasks.

Our Approach. We introduce a multi-turn RL framework for training VLMs to reason in a grounded, visually-aware manner. This stands in contrast to LLM reasoning in math or code, where grounding in external input is not strictly required. Within each reasoning step, the model produces a natural language thought along with a corresponding spatial grounding (i.e., an (x, y) location in the image). This enables it to progressively refine its attention and gather task-relevant visual information as reasoning unfolds. By incorporating multi-turn interaction into the RL process—where each turn consists of one or more reasoning steps followed by a query to a visual feedback tool—the model learns to iteratively request zoomed-in views of selected regions when fine-grained visual information is required. Critically, no external supervision or explicit human-provided grounding cues are used to supervise the spatial grounding of the thought; instead, the model autonomously learns to propose and utilize spatial grounding as an internal cognitive tool.

Current methods for training VLMs to directly produce textual answers from visual inputs inherently bias them toward abstract, ungrounded reasoning, making it fundamentally difficult for RL methods alone to spontaneously discover systematic visual strategies at the region-level. To explicitly inject grounded reasoning behaviors before RL training, we employ Monte Carlo Tree Search (MCTS) to systematically stitch together independently sampled reasoning steps, generating diverse, visually-

grounded reasoning trajectories. We bootstrap the model via supervised fine-tuning (SFT) on these MCTS-constructed paths, thus embedding rich region-level reasoning strategies into the model. We then apply RL, through Group Relative Policy Optimization (GRPO) [52], to further reinforce grounded sequences that lead to correct answers. Finally, we introduce a novel multi-turn RL formulation with visual feedback loops, allowing the model to dynamically zoom into image regions via tool calling for more detailed visual inspection when needed. This multi-turn variant of our method improves the model's capacity to localize and reason about fine-grained visual elements.

Empirical Results. We evaluate ViGoRL across a suite of visual reasoning benchmarks, including SAT-2 [50], BLINK [13], RoboSpatial [55], ScreenSpot [8, 33], VisualWebArena [31], and V*Bench [69]. Our approach consistently outperforms existing methods on all tasks. Specifically, ViGoRL achieves substantial improvements over vanilla GRPO, with accuracy gains of 12.9 points on SAT-2 and 2.0 points on BLINK. In fine-grained web grounding scenarios, our method surpasses both vanilla GRPO and large-scale web-finetuned models on ScreenSpot-Pro. By leveraging multi-turn RL for dynamic, zoomed-in visual feedback, ViGoRL further improves performance on ScreenSpot-Pro, effectively localizing small elements within high-resolution images. Moreover, multi-turn RL significantly enhances visual search capabilities, allowing ViGoRL to outperform both VLM tool-use pipelines and proprietary VLMs on V*Bench, achieving an accuracy of 86.4%. On VisualWebArena, a benchmark requiring live web interaction from image inputs alone, without access to HTML, ViGoRL outperforms both direct SFT and vanilla GRPO, and surpasses the previous state-of-the-art for this model size, ICAL [51], despite using only visual input.

Ablation studies confirm the importance of grounding: models trained without spatial anchoring perform significantly worse. Further, we find that grounding amplifies other visual cognitive behaviors such as region exploration, goal setting, and visual verification. Human evaluations show that our model's reasoning is both spatially accurate and helpful to understanding the model's reasoning steps. Our results point to visually grounded RL as a strong paradigm for general-purpose visual reasoning.

2 Related Work

Programmatic Reasoning in VLMs. Vision-Language Models (VLMs) [48, 29, 2, 7, 65, 58] excel on multimodal tasks through large-scale pretraining, but struggle with complex reasoning such as counting [49], spatial reasoning [50], and compositional understanding [62]. Prompting strategies like chain-of-thought (CoT) [67, 32], mm-CoT [88], IoT prompting [93], and Mind's Eye [71] guide models to generate explicit reasoning steps grounded in images. Methods such as V* [69], Sketchpad [26], VisProg [21], REFOCUS [14], and ViperGPT [56] use language models to produce executable visual plans but rely on frozen backbones and hand-crafted prompts.

Distillation and Supervised Fine-Tuning. To robustly instill reasoning skills, supervised fine-tuning (SFT) methods train on curated reasoning trajectories. For text tasks, STaR [85] generates reasoning via few-shot prompting and selects based on correctness, while S1 [42] distills reasoning chains into smaller models. Similar approaches in VLMs include LLAVA-CoT [74], which distills CoT reasoning from GPT-4o [1], and ICAL [51], VPD [27], and Mulberry [79], which employ LLMs or MCTS to generate reasoning data. Methods like VOCOT [35] improve grounding of entities via SFT. However, distillation methods rely solely on positive examples, neglecting failed paths. RL addresses this by learning from both successes and failures, outperforming SFT alone in our experiments.

Reinforcement Learning on Chains of Thought. Applying RL to chains of thought has improved reasoning in verifiable domains like math and coding [28, 19, 60]. Early methods [87, 45, 12, 70, 18] iteratively refine reasoning using preference-based approaches, while recent efforts like DeepSeek-R1 [19] and Kimi-1.5 [60] leverage online RL with outcome-based rewards. While initially believed to induce novel cognitive behaviors, analyses [15, 84, 38, 81] suggest RL primarily amplifies existing capabilities already found in the base model. Most prior work focuses on text-only domains, leaving visual reasoning behaviors largely unexplored. Visual-RFT [39] applies RL to textual reasoning in VLMs without incentivizing new visual behaviors. In contrast, we explicitly ground reasoning steps visually, amplifying exploration, verification, and backtracking via learned visual interactions.

3 Preliminaries

Reasoning in Vision-Language Models. The overarching objective of our work is to improve the reasoning capabilities of vision-language models (VLMs). We consider reasoning tasks defined by a dataset $\mathcal{D}$ of problem instances (I, q, a^*), where I is a visual input (e.g., an image), q is a natural language query about the image, and a^* is the correct, verifiable answer (e.g., a class label, bounding box, or discrete actions such as `click [element]`). The goal is to train a vision-language policy π_θ parameterized by θ that outputs a reasoning trace τ consisting of sequential textual reasoning steps $s_1, s_2, \ldots, s_T$ culminating in an answer a. This policy factorizes autoregressively:

$$\pi_\theta(\tau \mid I, q) \;=\; \Big(\prod_{t=1}^{T} \pi_\theta(s_t \mid I, q, s_{<t})\Big) \;\cdot\; \pi_\theta(a \mid I, q, s_{\le T}).$$

While supervised fine-tuning can teach models to mimic reasoning chains provided in training data, RL offers the potential to directly reinforce reasoning behavior sampled from the base model based on correctness or other reward signals [44, 52, 15]. Specifically, RL allows us to optimize policies over reasoning traces τ that maximize expected returns based on task performance and adherence to desired format structure. Formally, the RL objective can be expressed as:

$$\max_\theta \; \mathbb{E}_{(I,q,a^*)\sim\mathcal{D}} \; \big[\mathbb{E}_{\tau\sim\pi_\theta}\big[\, R(\tau)\big]\big], \quad \text{s.t. } \tau \text{ satisfies format constraints,}$$

where the reward $R(\tau)$ typically includes correctness of the final answer, and proper adherence to structured reasoning formats.

3.1 Do Current RL Recipes Amplify VLM Behaviors That Support Visual Reasoning?

It has been shown that RL on chains of thought alone cannot necessarily induce new behaviors from scratch; it can only amplify or chain reasoning primitives that are already present in the base model's sampling distribution [15, 84]. Do current base VLMs exhibit desirable visual reasoning behaviors and can RL amplify these behaviors to improve performance? Following previous work from LLMs [15], we categorize the behaviors of base VLMs when tasked with the Spatial Aptitude Test [50] spatial reasoning benchmark, which requires the synthesis of evidence in multiple regions of one or multiple images to answer. Our complete analysis and experimental protocol is detailed in Section 5.3 and Appendix A6. Our analysis reveals two key insights:

- ***Takeaway 1:*** *Current VLMs often fail to reference fine-grained image inputs; their reasoning is largely ungrounded.* Without explicit grounding, models treat vision as static context rather than actively referenced input, spending tokens on abstract thought instead of analyzing visual regions. Qwen2.5-VL-3B examines only 1.44 regions per task with minimal visual verification (0.14) and no backtracking. We provide an examples of such an output in Figure 2.
- ***Takeaway 2:*** *Standard RL optimization exacerbates ungrounded reasoning.* RL-tuning with task-level rewards slightly increases region exploration (1.8) but eliminates visual subgoal setting (0.00) and still shows no backtracking (0.00). Figure 2 and Appendix A9.1 illustrates typical outputs under this setup. This demonstrates that optimizing for correctness without encouraging grounded reasoning fails to instill visual reasoning skills. Furthermore, prompt engineering for broader visual behaviors is ignored during RL training (Section 5.2, Table 4).

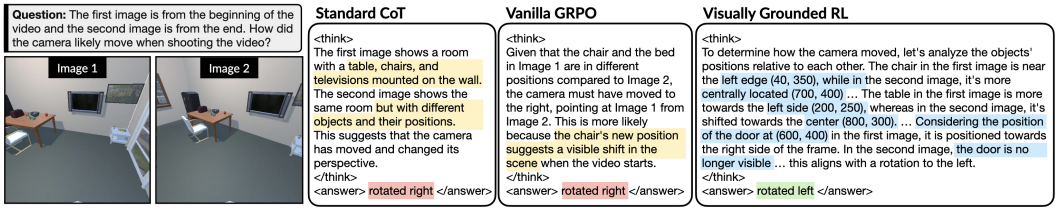


Figure 2: Without actively reinforcing visually grounded behaviors, RL collapses onto shortcuts that maximize immediate rewards at the expense of richer visual reasoning. **Standard CoT** and **Vanilla GRPO** (left and center) exhibit visually ungrounded reasoning, relying on vague references to scene elements (shown in yellow), which often results in incorrect answers (marked in red). In contrast, **Visually Grounded RL** (right) explicitly references object positions, demonstrating precise spatial grounding (shown in blue) and more often producing correct reasoning outcomes (marked in green). See Section 5.3 for further analysis.

Why does standard RL fail here? We hypothesize that the failure mode of RL comes down two interrelated issues: (1) the initial sampling distribution of pretrained VLMs heavily biases the model toward abstract, language-based strategies rather than region-level analysis (as shown in Takeaway 1 above), (2) since standard RL provides rewards based solely on final correctness and general formatting, it amplifies behaviors that attain high rewards irrespective of how they actually maximize reward. Without actively biasing the initial policy and reinforcing visually grounded behaviors, RL naturally collapses onto shortcuts that maximize immediate rewards at the expense of richer visual reasoning. *Thus, we hypothesize that encouraging explicit grounding biases the model's reasoning distributions toward exploring relevant visual regions, setting meaningful visual subgoals, verifying visual hypotheses, and effectively backtracking.*

4 Visually Grounded Reinforcement Learning (ViGoRL)

The analysis in Section 3.1 shows that naïve RL on small VLMs may degrade into ungrounded reasoning strategies, and that incentivizing grounding may improve visual reasoning behaviors that improve generalization. To incorporate explicit visual grounding into the reasoning process, we redefine each reasoning step as a tuple: $n_t = \langle s_t, (x_t, y_t) \rangle$, where s_t is a textual thought and (x_t, y_t) anchors it to a specific image location. The full trajectory becomes $\tau = [n_1, \ldots, n_T, a]$. This modifies the original factorization from Section 3:

$$\pi_\theta(\tau \mid I, q) = \left(\prod_{t=1}^{T} \pi_\theta(n_t \mid I, q, n_{<t}) \right) \cdot \pi_\theta(a \mid I, q, n_{\leq T}),$$

where each n_t now includes both the reasoning step and its visual grounding. By introducing this grounding constraint, we explicitly guide the model to systematically reference specific image locations as evidence for its reasoning. This grounding incentivizes the model to iteratively explore and verify distinct visual regions, formulate and ground intermediate subgoals visually, and revisit prior regions when uncertainty or errors arise.

4.1 Our Approach for Grounded Reinforcement Learning

Building on our findings in Section 3.1, we introduce a comprehensive approach that directly addresses the ungrounded reasoning patterns in VLMs. We propose a two-stage pipeline to incorporate grounded reasoning as detailed in the previous section: **(1) warm-start supervised finetuning** that biases the model toward generating structured reasoning chains with explicit spatial grounding, followed by **(2) reinforcement learning** that systematically refines these grounded behaviors. Finally, we extend our approach to multi-turn RL (Section 4.3.1), enabling fine-grained visual feedback at each reasoning step. This pipeline yields models that invest test-time compute into examining diverse image regions—precisely the behaviors our analysis showed were absent in standard VLM reasoning.

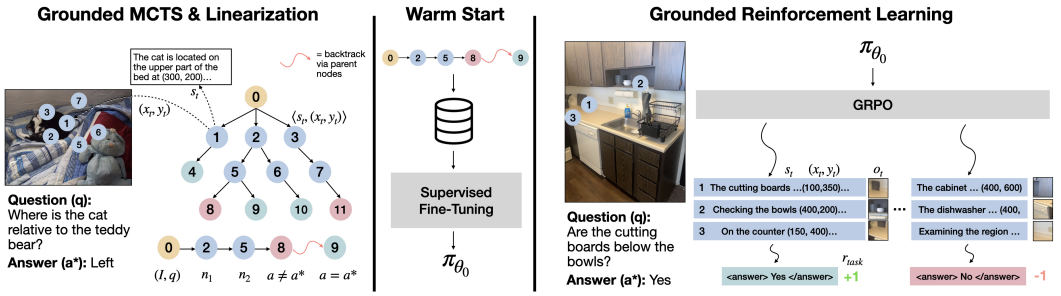


Figure 3: **Overview of the ViGoRL approach.** (Left) We use MCTS with a teacher model to generate reasoning chains grounded in specific image regions. (Middle) These reasoning trees are linearized and used for supervised fine-tuning (SFT) to train a base model. (Right) We apply GRPO with an outcome-based reward to further refine the grounded reasoning.

4.2 Warm-Start Data Generation via MCTS

MCTS with Visual Grounding. We employ MCTS to generate grounded reasoning traces, where each node is a reasoning step $n_t = \langle s_t, (x_t, y_t) \rangle$, anchoring thought s_t to image coordinates (x_t, y_t) (Figure 3, left panel). At each iteration: (1) *Selection*: nodes are traversed using UCB, prioritizing high-value, under-explored paths; (2) *Expansion*: the VLM samples new grounded steps $\langle s_t, (x_t, y_t) \rangle$ for unexplored nodes, each referencing a distinct image region; (3) *Simulation*: rollouts are performed

by recursively generating visually grounded steps until a terminal answer is produced; (4) *Backpropagation*: a judge scores terminal nodes, with rewards propagated up the path. This process ensures efficient exploration of promising image regions and reasoning steps. We provide additional details on our MCTS procedure in Appendix A5.5.

Why MCTS? Even large models (Qwen2.5-VL-72B) deployed with standard prompting explore only 2–3 regions, set few subgoals, and never backtrack (Table 5). Purely human-curated traces are costly to collect at scale, while linear rollouts cannot enforce the iterative exploration and corrective loops we desire. Empirically, distillation from such linear rollouts without MCTS leads to degraded generalization, performing worse on out-of-distribution spatial tasks after GRPO training (Table 4). In contrast, MCTS lets us systematically search the space of grounded reasoning steps, trading off exploration breadth and depth, and cheaply generate thousands of richly annotated paths that exhibit behaviors like wide exploration, through early branching to cover diverse image regions, and backtracking, by abandoning failing branches and revisiting alternatives.

Moreover, since our reasoning steps are already defined as $(s_t, (x_t, y_t))$ tuples, they map naturally onto MCTS nodes and transitions, making it both conceptually clean and computationally efficient.

Teacher-guided Search. We employ a frozen, high-capacity teacher (Qwen2.5-VL-72B) to expand each node in the MCTS tree. At node n, we prompt the teacher to either (a) generate a new grounded reasoning step s_n with coordinate, or (b) emit a candidate answer. We score leaf answers by correctness and backpropagate to guide tree expansion. From 1,500 prompts we derive ∼30k high-quality reasoning traces—a dataset orders of magnitude smaller than typical SFT corpora, but *densely* packed with exploration, verification, and backtracking behaviors.

Linearization for SFT. We then linearize selected root-to-leaf paths into two types of training examples: 1) **Direct chains:** successful trajectories leading to the correct answer with no detours, and 2) **Corrected chains:** trajectories where an initial rollout fails, triggers a "wait, that seems off" backtrack, and proceeds to the correct solution (Figure 3, left panel).

We denote the VLM finetuned on these MCTS chains as π_{θ_0} (Figure 3, middle). After fine-tuning, we commonly observe reasoning chains with dense visual subgoal setting, visual verification, broad region exploration, and visual backtracking. A representative trace can be seen in Appendix A9.2.

4.3 Reinforcement Learning with Spatially-Grounded Reasoning Steps

Although π_{θ_0} imitates high-quality traces, it does not reason *optimally* for new queries. We therefore apply Group Relative Policy Optimization (GRPO; 52) to directly maximize task reward while preserving fluency and grounding (Figure 3, right). More details can be found in Appendix A5.1.

Reward Design. We wish to incentivize reasoning traces that include explicit grounding coordinates, and reasoning traces that lead to the final answer. Our total reward is a weighted sum: $R(\mathcal{T}) = \lambda_{\text{fmt}} r_{\text{fmt}} + \lambda_{\text{task}} r_{\text{task}}$, where r_{fmt} encourages valid and interpretable output format, and r_{task} captures task-specific correctness. Importantly, along with checking outputs are formatted with correct `<think>` and `<answer>` tags, we award $+1$ r_{fmt} only if all coordinate references are valid. We provide more details on task reward in Appendix A5.2.

4.3.1 Multi-Turn Reinforcement Learning for Visual Feedback

Grounded reasoning chains generated by our MCTS procedure already push the model to *look* at many regions, but the visual encoder still processes the *same* globally resized image at every step. Fine-grained cues (small text, icons, object boundaries) are therefore blurred away, potentially limiting the benefit of additional reasoning if image region details cannot be perceived by the model. Inspired by how humans zoom in after selecting a candidate region, we let the model request a higher-resolution crop, o_t, after predicting a coordinate. This "interactive microscope" supplies fresh evidence at a detail level impossible to encode in the initial global view.

Multi-turn warm start: from single-turn chains to dialogs. To prepare the model for multi-turn rollouts, we convert single-step MCTS-derived reasoning traces from Section 4.2 into dialog-style traces. Given a linearized MCTS trace $\tau = [(s_1, \mathbf{p}_1), \ldots, (s_T, \mathbf{p}_T), a]$, we convert it into a *dialog*:

1. At turn t, the model generates a textual thought s_{t+1} (tagged `<think> ... </think>`).
2. The model then emits `<tool_call>` {"name": "crop", "arguments": {"coordinate": $\mathbf{p}_t$}} `</tool_call>`, or the answer (tagged `<answer> ... </answer>`), then the round terminates.

3. If a function call, the environment responds with o_t, with tag `<observation>` containing a $w \times w$ crop centered at $\mathbf{p}_t$, resized to $r \times r$ pixels, followed by `</observation>` and the loop repeats.

We fine-tune the base model on these multi-turn traces to initialize it for multi-step GRPO with visual feedback. We then apply GRPO, allowing the model to roll out full multi-turn dialogs, with crop feedback upon tool call outputs.

Reward Design for Multi-Turn RL. Multi-turn settings introduce new failure modes: repeating coordinates, skipping tool use, or violating dialog structure. To mitigate these, we define a composite format reward: **(1) Grammar reward** r_{grammar}: 1 if the dialog obeys a strict tag automaton: `<think>` $\rightarrow$ `</think>` $\rightarrow$ `<tool_call>` $\rightarrow$ `</tool_call>` $\rightarrow$ `<observation>` $\rightarrow$ `</observation>` $\rightarrow$ repeat or `<answer>` $\rightarrow$ `</answer>`. The dialog must end with a complete `</answer>` and contain no malformed or out-of-order tags. **(2) Diversity bonus** r_{div}: +0.2 for each sufficiently distinct coordinate in tool calls ($\geq$ 10px from all previous), up to 4 times. Finally, this leads to the overall format reward: $r_{\text{fmt}} = r_{\text{grammar}} + r_{\text{div}}$. Additional multi-turn RL details can be found in Appendix A5.3.

5 Experiments

We evaluate our model, ViGoRL, on spatial reasoning and web grounding tasks, comparing against baseline and ablation variants to understand the contributions of grounding, MCTS-based warm-start supervision, and GRPO. Our results demonstrate significant gains in visual reasoning and web-based tasks when models are trained with our grounded reasoning training recipe. We investigate the following research questions:
① **RQ1**: How much does grounded reasoning help when evaluated on visual reasoning tasks?
② **RQ2**: How important is each component of ViGoRL?
③ **RQ3**: What visual reasoning behaviors are amplified by grounded reasoning?
④ **RQ4**: Is the grounded reasoning accurate and interpretable?

Training Datasets. For **spatial reasoning**, we use SAT-2 [50], sampling 32k training and 1k validation examples. The model is tasked with selecting the correct textual option, with randomized answer order to reduce position bias. For **web grounding**, we draw 12k ⟨screenshot, referring expression, box⟩ examples from OS-ATLAS [72] (4k each from mobile, web, and desktop), plus 1.5k warm start and 1.5k validation samples evenly split by domain. For **web action prediction**, we use ICAL [51], a dataset of 92 web navigation trajectories. We remove chain-of-thought and textual set-of-marks annotations to focus on visual grounding, training on ⟨image, instruction, action history⟩ to predict the correct next action at each step. For **visual search**, we curate 11k question-answer pairs over small objects from Segment Anything [30], using GPT-4o given an object mask and scripted filtering to generate fine-grained ⟨image, question, choices, answer⟩ tuples. Each question targets a uniquely identifiable small object (<0.1% of image) and tests visual discrimination (e.g., color, material) and relation questions between object pairs.

5.1 RQ1: How much does grounded reasoning help on visual reasoning tasks?

Baselines. We compare against the following baselines:

1. *Method Comparison:* We compare ViGoRL to baselines utilizing the same training data and base model (Qwen-2.5-VL) as used in our method, but differing in their training recipe: 1) **SFT-direct:** Supervised fine-tuning on our trajectory dataset using final answers only. 2) **Vanilla GRPO:** GRPO applied to the base model with standard rewards for `<think>` and `<answer>` formatting, and answer correctness, similar to previous work [39, 40]. Our model and all baselines decode with a temperature of 0.5 during evaluation.

2. *General Proprietary and Open-Source Models:* General-purpose vision-language models with closed-weights accessible through APIs [1] and open-weight models [37, 9, 10, 65, 3, 61].

3. *VLM Tool-Using Pipelines:* We compare against prompt-based models that explicitly decompose visual tasks into sequences of subgoals or intermediate steps [69, 21, 25, 90, 78, 68].

4. *Web-Grounding Models:* Models trained to specialize in web grounding tasks, through large-scale supervised finetuning and instruction tuning on curated web data [8, 23, 72, 36, 17, 47], reinforcement learning of chain of thought with outcome reward [41], or human-in-the-loop action and chain of thought annotations collected from live webpages [51].

5.1.1 Visual Reasoning Evaluations.

Spatial Reasoning. We evaluate on SAT-2 [50] validation (4,000 questions across 5 categories), BLINK [13] (depth ordering, multi-view reasoning, spatial relationships), and ROBOSPATIAL-configuration and compatibility split (228 questions on real-world RGBD scenes) [55]. These benchmarks test generalization to novel environments, objects, and language configurations. Accuracy is measured via multiple-choice answer matching. We use a max side length of 1260 pixels while keeping aspect ratio for evaluation.

GUI Understanding. For grounding evaluation, we use ScreenSpot v2 [72, 8] (single-step localization), and ScreenSpot Pro [33] (high-resolution professional environments). To test small element grounding in low resolution, we additionally evaluate on ScreenSpot Pro with images downsampled to a lower resolution of 1920x1920, which we call ScreenSpot-Pro-LR (results in Appendix Table A6). Performance is measured by checking if predicted coordinates fall within target element bounding boxes. We also test on VisualWebArena [31] live web evaluation, using the visual-only configuration where models receive set-of-marks annotated webpage screenshots without HTML or text inputs. Task success is determined automatically by checking for specific criteria in the agent trajectory (see Koh et al. [31] for details). We use a resize image to have maximum pixel size of 3600x3600.

Visual Search. The visual search benchmark V*Bench tests fine-grained visual understanding using 191 high-resolution images from the SA-1B dataset (avg. 2246×1582). It includes 115 attribute recognition samples (e.g., color, material) and 76 spatial relationship samples, evaluating models' ability to analyze detailed object properties and relative positions in complex scenes. We use a resize image to have maximum pixel size of 3600x3600.

5.1.2 Results

Table 1: Performance (mean ± 95% CI) on ScreenSpot and VisualWebArena. Multi-t = multi-turn RL with visual feedback.

Model	ScreenSpot -V2	ScreenSpot -Pro	VWA (Vision Only)
Proprietary Models			
GPT-4o	18.1	0.8	19.8 (w/ text)
Claude Comp. Use	-	17.1	–
Open-source Models			
Qwen2-VL-7B	-	1.6	2.9 (w/ text)
Kimi-VL-16B-MoE	92.8	34.5	–
Web Grounding Models			
SeeClick	55.1	1.1	–
CogAgent-18B	-	7.7	–
OS-Atlas-4B	71.9	3.7	–
OS-Atlas-7B	84.1	18.9	–
ShowUI-2B	-	7.7	–
UGround-7B	-	16.5	–
UGround-V1-7B	–	31.1	–
UI-TARS-2B	84.7	27.7	–
UI-R1-3B	–	17.8	–
ICAL-7B	–	-	8.2 (w/ text)
Method Comparison			
Qwen2.5-VL-3B	68.4 (±2.6)	23.9	4.2 (±1.3)
+ SFT direct	80.6 (±2.2)	25.0 (±2.1)	4.5 (±1.4)
+ Vanilla GRPO	84.4 (±2.0)	29.0 (±2.2)	4.8 (±1.4)
ViGoRL-3b (Ours)	**86.5** (±1.9)	**31.1** (±2.3)	**6.4** (±1.5)
Multi-t ViGoRL-3b (Ours)	**86.1** (±1.9)	**32.3** (±2.4)	-
Qwen2.5-VL-7B	73.6 (±2.4)	29.0	5.5 (±1.5)
ViGoRL-7b (Ours)	**91.0** (±1.6)	**33.1** (±2.3)	**11.2** (±2.1)

Table 2: Accuracy (mean ± 95% CI) for spatial reasoning.

Model	SAT-2 Val	BLINK	RoboSpatial
Proprietary Models			
GPT-4o	–	60.0	76.2
GPT-4 Turbo	–	54.6	–
Claude 3 Opus	–	44.1	–
Gemini Pro 1.0	–	45.2	–
General Open-source Models			
LLaVA v1.6 34B	–	46.8	–
instructBLIP 13B	–	42.2	–
Molmo	–	-	67.1
Method Comparison			
Qwen2.5VL-3B	46.1 (±1.5)	44.4 (±2.3)	54.4 (±6.5)
+ SFT direct	58.3 (±1.5)	46.4 (±2.3)	62.3 (±6.3)
+ Vanilla GRPO	50.0 (±1.6)	46.5 (±2.3)	**69.7** (±6.0)
ViGoRL-3b (Ours)	**62.9** (±1.5)	**48.5** (±2.3)	**67.1** (±6.1)
Qwen2.5-VL-7B	52.6 (±1.6)	52.9 (±2.3)	46.7 (±9.5)
ViGoRL-7b (Ours)	**67.5** (±1.5)	**54.1** (±2.3)	**76.4** (±7.5)

Grounded reasoning improves spatial accuracy. As shown in Table 2, on a variety of spatial understanding benchmarks, ViGoRL significantly outperforms baselines without grounded reasoning. On SAT-2 test set, ViGoRL-3B achieves 62.9% accuracy—an improvement of +16.8 points over the base model and +12.9 points over Vanilla GRPO. Similar trends hold on the out of distribution benchmarks of RoboSpatial (67.1%) and BLINK (48.5%), with ViGoRL-3B outperforming SFT Direct by 2.3% and 4.8% on BLINK and RoboSpatial, respectively, and vanilla GRPO by 2.0% on BLINK. At the 7B scale, ViGoRL-7B further improves SAT-2 accuracy to 67.5%, demonstrating the

method's scalability and effectiveness across model sizes. We observe the same trend in BLINK and RoboSpatial, with improved accuracies of 54.1% and 76.4%, respectively. Importantly, these gains come without sacrificing general visual-language capabilities, as our method maintains performance on standard out-of-distribution VLM benchmarks (Table A7).

Grounded reasoning helps localize complex web elements. We evaluate performance on vision-only web interfaces (Table 1), where models must resolve ambiguous UI instructions via grounded image understanding. ViGoRL-3B outperforms both the base model, direct answer SFT, and vanilla GRPO across all tasks. For instance, on ScreenSpot-Pro, requiring grounding of small elements in high resolution webpage images, accuracy improves 21.8 points over the base model, 6.1 points over SFT direct, and 2.1 points over vanilla GRPO. ViGoRL-7B achieves 91.0% on ScreenSpot-V2 and 33.1% on ScreenSpot-Pro, outperforming open-source VLMs of comparable size, including OS-Atlas and UGround variants, which finetune on order of magnitudes more GUI grounding examples (up to 13M training samples).

ViGoRL improves accuracy on live visual web evaluation. On VisualWebArena, which requires live interaction with web pages using only set-of-marks image inputs (no access to HTML or underlying text), ViGoRL outperforms direct SFT and vanilla GRPO. Despite relying solely on images, ViGoRL surpasses the previous state-of-the-art for the same model size, ICAL [51], by 3.0%—even though ICAL has access to textual set-of-marks inputs derived from HTML.

Multi-turn RL with visual feedback improves visual search and small element detection. On V*Bench, ViGoRL-7B significantly outperforms both proprietary models and open-source tool-using pipelines. As shown in Table 3, our method reaches 86.4%, surpassing proprietary VLMs like GPT-4o (66.0%) and Gemini-Pro (48.2%), and even advanced tool-using pipelines like VisProg (41.4%) Sketchpad-GPT-4o (80.3%).

Table 3: Accuracy (mean ± 95% CI) for visual search on V*Bench.

Model Name	V*Bench
Proprietary Models	
Gemini-Pro	48.2
GPT-4V	55.0
GPT-4o	66.0
VLM Tool-Using Pipelines	
VisProg	41.4
VisualChatGPT	37.6
MM-React	41.4
Sketchpad-GPT-4o	80.3
IVM-Enhanced GPT-4V	81.2
Open-source Models	
LLaVA-1.5-7B	48.7
LLaVA-1.6-13B	61.8
SEAL	74.8
Qwen2.5-7B-VL	78.0 (±3.0)
Multi-t ViGoRL-7B	**86.4** (±2.7)
Qwen2.5-3B-VL	74.2 (±3.1)
ViGoRL-3B (Ours)	79.1 (±3.0)
Multi-t ViGoRL-3B	**81.2** (±2.8)
w/o diversity reward	78.0 (±3.0)
w/ bounding box outputs	**81.2** (±2.8)

Our model with multi-turn RL that incorporates zoomed-in visual feedback additionally shows significant improvements in small element grounding tasks. It achieves 32.3% accuracy on ScreenSpot-Pro, a 1.2 percentage point improvement over our best non-visual feedback model variant (31.1%) (Table 1). In low-resolution environments, the zooming capability delivers even more substantial relative gains, outperforming our best non-visual feedback method by 2.4% (Table A6). These gains illustrate that our approach enables more robust and precise visual reasoning through structured visual grounding and dynamic visual feedback interactions.

5.2 RQ2: Ablation Studies

Table 4: Ablation results on SAT-2 Val and BLINK. Top-1 accuracy (±95% CI) with relative change. *Model never produced correct formatting. Gnd = Explicit grounding in the reasoning steps. SFT = SFT Direct pretraining. Distill = warm-start teacher distillation.

Variant	GRPO	MCTS	Gnd	Distill	SAT-2	BLINK
Full (ours)	✓	✓	✓	✓	62.93 (±1.50)	48.50 (±2.33)
– GRPO		✓	✓	✓	58.83 (±1.53) (–4.10%)	44.97 (±2.32) (–3.53%)
– MCTS	✓		✓		N/A*	N/A*
– Grounded	✓	✓		✓	58.69 (±1.53) (–4.28%)	45.44 (±2.32) (–3.06%)
Distill w/o MCTS	✓		✓	✓	63.28 (±1.49) (+0.35%)	46.18 (±2.32) (–2.32%)
+ SFT direct pre-train	✓	✓	✓	✓	63.26 (±1.49) (+0.33%)	48.56 (±2.33) (+0.06%)

To understand which components drive performance, we conduct targeted ablations (Table 4):

1. GRPO remains important. Removing GRPO reduces performance by 4.1 points on SAT-2 and 3.5 points on BLINK, confirming the importance of RL refinement.

2. MCTS-generated warm-start is essential. Without structured traces from MCTS, the model fails to emit valid outputs, underscoring the need for scaffolded learning signals.

3. Explicit grounding helps. Running our same method without introducing explicit grounding in the warm start data resulted in a 3.4 point drop on BLINK, even with the same MCTS warm start and GRPO recipe.

4. Teacher distillation without MCTS preserves in-distribution performance but degrades out-of-distribution generalization. When using warm-start data from successful teacher linear rollouts (which contain grounded reasoning steps but no search), SAT-2 validation accuracy remains nearly unchanged (63.26% vs. 62.93%). However, performance on the more challenging out-of-distribution BLINK benchmark drops significantly by 2.32%.

5. SFT direct pretraining adds little. Adding an additional SFT stage to directly predict the answer before warm start training and GRPO yields marginal gains (<1 point), reinforcing the idea that grounded thought processes drive improvement.

6. Pointing is sufficient for visual search. As shown in Table 3, we observe no difference in V*Bench accuracy when switching to training ViGoRL to output variable size bounding box crops as opposed to the fixed-size point cropping used in ViGoRL. Fixed crops usually provide enough context and resolution for most cases, and ViGoRL can handle larger regions through multiple overlapping crops that together capture necessary information. Adaptive cropping could be more efficient but is limited by imprecise bounding boxes and inconsistent aspect ratios.

5.3 RQ3: What visual reasoning behaviors are amplified by grounded reasoning?

Following previous work on behavioral coding in language models [15], we code VLM behavior when tasked with the SAT-2 spatial reasoning benchmark, which requires synthesizing evidence across multiple image regions, in two scenarios: (1) Zero-shot prompting with "think step-by-step" instructions without explicit grounding, and (2) RL-tuned models including vanillo GRPO, ViGoRL, and an ablated version without explicit grounding incentives. We quantify reasoning behaviors on 300 representative SAT-2 samples using GPT-4o evaluation, measuring visual regions explored, subgoal setting, verification, and backtracking. Additional study details can be found in Appendix A6.

Table 5: Average visual behaviors per example.

Model	Regions Explored	Grounded Subgoals	Visually Verify	Backtrack	Acc.
Qwen2.5-VL-72B (Zero-Shot)					
Standard CoT	2.3	0.27	0.27	0.00	0.65
Qwen2.5-VL-3B (Zero-Shot)					
Standard CoT	1.44	0.07	0.14	0.00	0.38
Qwen2.5-VL-3B (RL-tuned)					
Vanilla GRPO	1.8	0.00	0.17	0.00	0.48
Ours	**3.5**	**1.1**	**0.39**	**0.47**	**0.64**
w/o grounding	1.7	0.02	0.27	0.02	0.58

Findings. As shown in Table 5, explicit grounding substantially amplifies visual reasoning behaviors. ViGoRL explores more visual regions (3.5 vs. 1.44/1.8 in zero-shot/vanilla GRPO) and demonstrates dramatic increases in grounded subgoals (1.1 vs. 0.07, 15× higher) and visual verification (0.39 vs. 0.14, 3× higher). It uniquely develops visual backtracking behavior (0.47) absent in all baselines. These enhanced behaviors enable our 3B parameter model to achieve accuracy (0.64) comparable to the 72B model (0.65). The ablation study confirms these benefits stem specifically from explicit grounding incentives, as removing them causes substantial regression in all measured behaviors, particularly regions explored (1.7 vs. 3.5) and grounded subgoals (0.02 vs. 1.1).

5.4 RQ4: Human Evaluation Shows Accuracy and Helpfulness of Grounded Reasoning

To assess our model's grounded reasoning traces, we conducted a human study evaluating whether predicted coordinates (1) correctly referred to the intended image region, and (2) helped participants understand the associated reasoning step (details are shown in Appendix A7).

Findings. As shown in Figure 4, 72.8% of predictions were judged as accurately referring to the described region (95% CI: [66.8, 78.7], N=20). On a 5-point Likert scale, participants rated the helpfulness of the highlighted region at 3.35 on average (95% CI: [3.03, 3.68], N=10). Helpfulness increased substantially when the prediction was correct (3.81; 95% CI: [3.51, 4.10]) and dropped when incorrect (2.26; 95% CI: [1.57, 2.94]).

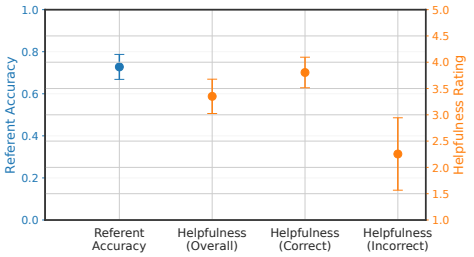


Figure 4: Human evaluation of grounded reasoning. Participants judged the grounded predictions as both accurate and helpful when correct.

These findings indicate that accurate spatial grounding meaningfully improves human interpretability and usefulness of the model's reasoning process, indicating that improvements in accuracy of reasoning step grounding can also improve human interpretability.

Acknowledgments. This material is based upon work supported by National Science Foundation grants GRF DGE1745016 & DGE2140739 (GS), ONR award N00014-23-1-2415, AFOSR Grant FA9550-23-1-0257, and DARPA No. HR00112490375 from the U.S. DARPA Friction for Accountability in Conversational Transactions (FACT) program. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the United States Army, the National Science Foundation, or the United States Air Force.

This research project has benefitted from the Microsoft Accelerate Foundation Models Research (AFMR) grant program through which leading foundation models hosted by Microsoft Azure along with access to Azure credits were provided to conduct the research.

References

[1] Openai. gpt-4 technical report. *arXiv preprint arxiv:2303.08774*, 2023.

[2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35: 23716–23736, 2022.

[3] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[4] Dana H Ballard, Mary M Hayhoe, Polly K Pook, and Rajesh PN Rao. Deictic codes for the embodiment of cognition. *Behavioral and Brain Sciences*, 20(4):723–742, 1997.

[5] Nicholas Budny, Kia Ghods, Declan Campbell, Raja Marjieh, Amogh Joshi, Sreejan Kumar, Jonathan D Cohen, Taylor W Webb, and Thomas L Griffiths. Visual serial processing deficits explain divergences in human and vlm reasoning. *arXiv preprint arXiv:2509.25142*, 2025.

[6] Declan Campbell, Sunayana Rane, Tyler Giallanza, Camillo Nicolò De Sabbata, Kia Ghods, Amogh Joshi, Alexander Ku, Steven Frankland, Tom Griffiths, Jonathan D Cohen, et al. Understanding the limits of vision language models through the lens of the binding problem. *Advances in Neural Information Processing Systems*, 37:113436–113460, 2024.

[7] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Nan Ding, Keran Rong, Hassan Akbari, Gaurav Mishra, Linting Xue, Ashish Thapliyal, James Bradbury, Weicheng Kuo, Mojtaba Seyedhosseini, Chao Jia, Burcu Karagol Ayan, Carlos Riquelme, Andreas Steiner, Anelia Angelova, Xiaohua Zhai, Neil Houlsby, and Radu Soricut. Pali: A jointly-scaled multilingual language-image model, 2023. URL https://arxiv.org/abs/2209.06794.

[8] Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Yantao Li, Jianbing Zhang, and Zhiyong Wu. Seeclick: Harnessing gui grounding for advanced visual gui agents. *arXiv preprint arXiv:2401.10935*, 2024.

[9] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023. URL https://arxiv.org/abs/2305.06500.

[10] Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, et al. Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models. *arXiv preprint arXiv:2409.17146*, 2024.

[11] Zhengyuan Ding, Yuwei Chen, Yichong Xu, Zhe Wang, Xintao Han, Dong Yu, and Zhou Yu. Attention over learned object embeddings enables complex visual reasoning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[12] Yuhao Dong, Zuyan Liu, Hai-Long Sun, Jingkang Yang, Winston Hu, Yongming Rao, and Ziwei Liu. Insight-v: Exploring long-chain visual reasoning with multimodal large language models, 2025. URL https://arxiv.org/abs/2411.14432.

[13] Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A Smith, Wei-Chiu Ma, and Ranjay Krishna. Blink: Multimodal large language models can see but not perceive. *arXiv preprint arXiv:2404.12390*, 2024.

[14] Xingyu Fu, Minqian Liu, Zhengyuan Yang, John Corring, Yijuan Lu, Jianwei Yang, Dan Roth, Dinei Florencio, and Cha Zhang. Refocus: Visual editing as a chain of thought for structured image understanding, 2025. URL https://arxiv.org/abs/2501.05452.

[15] Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D Goodman. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars. *arXiv preprint arXiv:2503.01307*, 2025.

[16] Sarthak Ghosh, Ben Lee, Jean-Baptiste Alayrac, Xuhong Zhai, Christoph Feichtenhofer, Joao Carreira, and Ishan Misra. Grounded decoding with visual descriptions reduces hallucination in large vision-language models. In *International Conference on Learning Representations (ICLR)*, 2024.

[17] Boyu Gou, Ruohan Wang, Boyuan Zheng, Yanan Xie, Cheng Chang, Yiheng Shu, Huan Sun, and Yu Su. Navigating the digital world as humans do: Universal visual grounding for GUI agents. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=kxnoqaisCT.

[18] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

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

[20] Arjun Gupta, Xi Victoria Lin, Chunyuan Zhang, Michel Galley, Jianfeng Gao, and Carlos Guestrin Ferrer. Robust compositional visual reasoning via language-guided neural module networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[21] Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training, 2022. URL https://arxiv.org/abs/2211.11559.

[22] Stevan Harnad. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1-3): 335–346, 1990.

[23] Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, et al. Cogagent: A visual language model for gui agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14281–14290, 2024.

[24] David Hoppe and Constantin A Rothkopf. Multi-step planning of eye movements in visual search. *Scientific reports*, 9(1):144, 2019.

[25] Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, and Ranjay Krishna. Visual sketchpad: Sketching as a visual chain of thought for multimodal language models. *arXiv preprint arXiv:2406.09403*, 2024.

[26] Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, and Ranjay Krishna. Visual sketchpad: Sketching as a visual chain of thought for multimodal language models, 2024. URL https://arxiv.org/abs/2406.09403.

[27] Yushi Hu, Otilia Stretcu, Chun-Ta Lu, Krishnamurthy Viswanathan, Kenji Hata, Enming Luo, Ranjay Krishna, and Ariel Fuxman. Visual program distillation: Distilling tools and programmatic reasoning into vision-language models, 2024. URL https://arxiv.org/abs/2312.03052.

[28] Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

[29] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR, 2021.

[30] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

[31] Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Chong Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Ruslan Salakhutdinov, and Daniel Fried. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. *arXiv preprint arXiv:2401.13649*, 2024.

[32] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners, 2023. URL https://arxiv.org/abs/2205.11916.

[33] Kaixin Li, Ziyang Meng, Hongzhan Lin, Ziyang Luo, Yuchen Tian, Jing Ma, Zhiyong Huang, and Tat-Seng Chua. Screenspot-pro: Gui grounding for professional high-resolution computer use, 2025. URL https://likaixin2000.github.io/papers/ScreenSpot_Pro.pdf. Preprint.

[34] Xinhao Li, Ziang Yan, Desen Meng, Lu Dong, Xiangyu Zeng, Yinan He, Yali Wang, Yu Qiao, Yi Wang, and Limin Wang. Videochat-r1: Enhancing spatio-temporal perception via reinforcement fine-tuning, 2025. URL https://arxiv.org/abs/2504.06958.

[35] Zejun Li, Ruipu Luo, Jiwen Zhang, Minghui Qiu, Xuanjing Huang, and Zhongyu Wei. Vocot: Unleashing visually grounded multi-step reasoning in large multi-modal models, 2025. URL https://arxiv.org/abs/2405.16919.

[36] Kevin Qinghong Lin, Linjie Li, Difei Gao, Zhengyuan Yang, Shiwei Wu, Zechen Bai, Weixian Lei, Lijuan Wang, and Mike Zheng Shou. Showui: One vision-language-action model for gui visual agent. *arXiv preprint arXiv:2411.17465*, 2024.

[37] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023.

[38] Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective, 2025. URL https://arxiv.org/abs/2503.20783.

[39] Ziyu Liu, Zeyi Sun, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Haodong Duan, Dahua Lin, and Jiaqi Wang. Visual-rft: Visual reinforcement fine-tuning, 2025. URL https://arxiv.org/abs/2503.01785.

[40] Zhengxi Lu, Yuxiang Chai, Yaxuan Guo, Xi Yin, Liang Liu, Hao Wang, Han Xiao, Shuai Ren, Guanjing Xiong, and Hongsheng Li. Ui-r1: Enhancing action prediction of gui agents by reinforcement learning, 2025. URL https://arxiv.org/abs/2503.21620.

[41] Zhengxi Lu, Yuxiang Chai, Yaxuan Guo, Xi Yin, Liang Liu, Hao Wang, Guanjing Xiong, and Hongsheng Li. Ui-r1: Enhancing action prediction of gui agents by reinforcement learning. *arXiv preprint arXiv:2503.21620*, 2025.

[42] Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling, 2025. URL https://arxiv.org/abs/2501.19393.

[43] NVIDIA, :, Alisson Azzolini, Hannah Brandon, Prithvijit Chattopadhyay, Huayu Chen, Jinju Chu, Yin Cui, Jenna Diamond, Yifan Ding, Francesco Ferroni, Rama Govindaraju, Jinwei Gu, Siddharth Gururani, Imad El Hanafi, Zekun Hao, Jacob Huffman, Jingyi Jin, Brendan Johnson, Rizwan Khan, George Kurian, Elena Lantz, Nayeon Lee, Zhaoshuo Li, Xuan Li, Tsung-Yi Lin, Yen-Chen Lin, Ming-Yu Liu, Alice Luo, Andrew Mathau, Yun Ni, Lindsey Pavao, Wei Ping, David W. Romero, Misha Smelyanskiy, Shuran Song, Lyne Tchapmi, Andrew Z. Wang, Boxin Wang, Haoxiang Wang, Fangyin Wei, Jiashu Xu, Yao Xu, Xiaodong Yang, Zhuolin Yang, Xiaohui Zeng, and Zhe Zhang. Cosmos-reason1: From physical common sense to embodied reasoning, 2025. URL https://arxiv.org/abs/2503.15558.

[44] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

[45] Pranav Putta, Edmund Mills, Naman Garg, Sumeet Motwani, Chelsea Finn, Divyansh Garg, and Rafael Rafailov. Agent q: Advanced reasoning and learning for autonomous ai agents, 2024. URL https://arxiv.org/abs/2408.07199.

[46] Jinyi Qi, Tao Zhang, Rui Chen, Xiaoxue Li, Yizhou Zhang, and Kai-Wei Chang. Cogcom: Compositional visual reasoning with chain-of-manipulations. In *International Conference on Learning Representations (ICLR)*, 2025.

[47] Yujia Qin, Yining Ye, Junjie Fang, Haoming Wang, Shihao Liang, Shizuo Tian, Junda Zhang, Jiahao Li, Yunxin Li, Shijue Huang, et al. Ui-tars: Pioneering automated gui interaction with native agents. *arXiv preprint arXiv:2501.12326*, 2025.

[48] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[49] Pooyan Rahmanzadehgervi, Logan Bolton, Mohammad Reza Taesiri, and Anh Totti Nguyen. Vision language models are blind: Failing to translate detailed visual features into words, 2025. URL https://arxiv.org/abs/2407.06581.

[50] Arijit Ray, Jiafei Duan, Ellis Brown, Reuben Tan, Dina Bashkirova, Rose Hendrix, Kiana Ehsani, Aniruddha Kembhavi, Bryan A. Plummer, Ranjay Krishna, Kuo-Hao Zeng, and Kate Saenko. Sat: Dynamic spatial aptitude training for multimodal language models, 2025. URL https://arxiv.org/abs/2412.07755.

[51] Gabriel Herbert Sarch, Lawrence Jang, Michael J Tarr, William W Cohen, Kenneth Marino, and Katerina Fragkiadaki. Vlm agents generate their own memories: Distilling experience into embodied programs of thought. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[52] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. *URL https://arxiv. org/abs/2402.03300*.

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

[54] 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. *arXiv preprint arXiv: 2409.19256*, 2024.

[55] Chan Hee Song, Valts Blukis, Jonathan Tremblay, Stephen Tyree, Yu Su, and Stan Birchfield. RoboSpatial: Teaching spatial understanding to 2D and 3D vision-language models for robotics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. To appear.

[56] Dídac Surís, Sachit Menon, and Carl Vondrick. Vipergpt: Visual inference via python execution for reasoning. *arXiv preprint arXiv:2303.08128*, 2023.

[57] Huajie Tan, Yuheng Ji, Xiaoshuai Hao, Minglan Lin, Pengwei Wang, Zhongyuan Wang, and Shanghang Zhang. Reason-rft: Reinforcement fine-tuning for visual reasoning, 2025. URL https://arxiv.org/abs/2503.20752.

[58] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, and et. Al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context, 2024. URL https://arxiv.org/abs/2403.05530.

[59] Kimi Team, A Du, B Gao, B Xing, C Jiang, C Chen, C Li, C Xiao, C Du, C Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms, 2025. *URL https://arxiv. org/abs/2501.12599*.

[60] Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, Chuning Tang, Congcong Wang, Dehao Zhang, Enming Yuan, Enzhe Lu, Fengxiang Tang, Flood Sung, Guangda Wei, Guokun Lai, Haiqing Guo, Han Zhu, Hao Ding, Hao Hu, Hao Yang, Hao Zhang, Haotian Yao, Haotian Zhao, Haoyu Lu, Haoze Li, Haozhen Yu, Hongcheng Gao, Huabin Zheng, Huan Yuan, Jia Chen, Jianhang Guo, Jianlin Su, Jianzhou Wang, Jie Zhao, Jin Zhang, Jingyuan Liu, Junjie Yan, Junyan Wu, Lidong Shi, Ling Ye, Longhui Yu, Mengnan Dong, Neo Zhang, Ningchen Ma, Qiwei Pan, Qucheng Gong, Shaowei Liu, Shengling Ma, Shupeng Wei, Sihan Cao, Siying Huang, Tao Jiang, Weihao Gao, Weimin Xiong, Weiran He, Weixiao Huang, Wenhao Wu, Wenyang He, Xianghui Wei, Xianqing Jia, Xingzhe Wu, Xinran Xu, Xinxing Zu, Xinyu Zhou, Xuehai Pan, Y. Charles, Yang Li, Yangyang Hu, Yangyang Liu, Yanru Chen, Yejie Wang, Yibo Liu, Yidao Qin, Yifeng Liu, Ying Yang, Yiping Bao, Yulun Du, Yuxin Wu, Yuzhi Wang, Zaida Zhou, Zhaoji Wang, Zhaowei Li, Zhen Zhu, Zheng Zhang, Zhexu Wang, Zhilin Yang, Zhiqi Huang, Zihao Huang, Ziyao Xu, and Zonghan Yang. Kimi k1.5: Scaling reinforcement learning with llms, 2025. URL https://arxiv.org/abs/2501.12599.

[61] Kimi Team, Angang Du, Bohong Yin, Bowei Xing, Bowen Qu, and et. Al. Kimi-VL technical report, 2025. URL https://arxiv.org/abs/2504.07491.

[62] Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. Winoground: Probing vision and language models for visio-linguistic compositionality, 2022. URL https://arxiv.org/abs/2204.03162.

[63] Anne M. Treisman and Garry Gelade. A feature-integration theory of attention. *Cognitive Psychology*, 12(1):97–136, 1980.

[64] Shimon Ullman. Visual routines. *Cognition*, 18(1-3):97–159, 1984.

[65] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[66] Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, Eli Gottlieb, Yiping Lu, Kyunghyun Cho, Jiajun Wu, Li Fei-Fei, Lijuan Wang, Yejin Choi, and Manling Li. Ragen: Understanding self-evolution in llm agents via multi-turn reinforcement learning, 2025. URL https://arxiv.org/abs/2504.20073.

[67] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2022.

[68] Chenfei Wu, Shengming Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. Visual chatgpt: Talking, drawing and editing with visual foundation models, 2023. URL https://arxiv.org/abs/2303.04671.

[69] Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal llms. *arXiv preprint arXiv:2312.14135*, 2023.

[70] Tianhao Wu, Janice Lan, Weizhe Yuan, Jiantao Jiao, Jason Weston, and Sainbayar Sukhbaatar. Thinking llms: General instruction following with thought generation, 2024. URL https://arxiv.org/abs/2410.10630.

[71] Wenshan Wu, Shaoguang Mao, Yadong Zhang, Yan Xia, Li Dong, Lei Cui, and Furu Wei. Mind's eye of llms: Visualization-of-thought elicits spatial reasoning in large language models, 2024. URL https://arxiv.org/abs/2404.03622.

[72] Zhiyong Wu, Zhenyu Wu, Fangzhi Xu, Yian Wang, Qiushi Sun, Chengyou Jia, Kanzhi Cheng, Zichen Ding, Liheng Chen, Paul Pu Liang, et al. Os-atlas: A foundation action model for generalist gui agents. *arXiv preprint arXiv:2410.23218*, 2024.

[73] xAI. Grok-1.5 vision preview. https://x.ai/blog/grok-1.5v, 2024. Accessed: 2025-05-21.

[74] Guowei Xu, Peng Jin, Hao Li, Yibing Song, Lichao Sun, and Li Yuan. Llava-cot: Let vision language models reason step-by-step, 2025. URL https://arxiv.org/abs/2411.10440.

[75] Jihan Yang, Shusheng Yang, Anjali W Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. *arXiv preprint arXiv:2412.14171*, 2024.

[76] Scott Cheng-Hsin Yang, Mate Lengyel, and Daniel M Wolpert. Active sensing in the categorization of visual patterns. *Elife*, 5:e12215, 2016.

[77] Scott Cheng-Hsin Yang, Daniel M Wolpert, and Máté Lengyel. Theoretical perspectives on active sensing. *Current opinion in behavioral sciences*, 11:100–108, 2016.

[78] Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. Mm-react: Prompting chatgpt for multimodal reasoning and action. 2023.

[79] Huanjin Yao, Jiaxing Huang, Wenhao Wu, Jingyi Zhang, Yibo Wang, Shunyu Liu, Yingjie Wang, Yuxin Song, Haocheng Feng, Li Shen, and Dacheng Tao. Mulberry: Empowering mllm with o1-like reasoning and reflection via collective monte carlo tree search, 2024. URL https://arxiv.org/abs/2412.18319.

[80] Alfred L. Yarbus. *Eye Movements and Vision*. Springer, 1967.

[81] Edward Yeo, Yuxuan Tong, Morry Niu, Graham Neubig, and Xiang Yue. Demystifying long chain-of-thought reasoning in llms, 2025. URL https://arxiv.org/abs/2502.03373.

[82] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, et al. Dapo: An open-source llm reinforcement learning system at scale, 2025. *URL https://arxiv. org/abs/2503.14476*.

[83] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi, 2024. URL https://arxiv.org/abs/2311.16502.

[84] Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model?, 2025. URL https://arxiv.org/abs/2504.13837.

[85] Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. Star: Bootstrapping reasoning with reasoning, 2022. URL https://arxiv.org/abs/2203.14465.

[86] Yuexiang Zhai, Hao Bai, Zipeng Lin, Jiayi Pan, Shengbang Tong, Yifei Zhou, Alane Suhr, Saining Xie, Yann LeCun, Yi Ma, and Sergey Levine. Fine-tuning large vision-language models as decision-making agents via reinforcement learning, 2024. URL https://arxiv.org/abs/2405.10292.

[87] Xuan Zhang, Chao Du, Tianyu Pang, Qian Liu, Wei Gao, and Min Lin. Chain of preference optimization: Improving chain-of-thought reasoning in llms. *Advances in Neural Information Processing Systems*, 37:333–356, 2024.

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

[89] Baining Zhao, Ziyou Wang, Jianjie Fang, Chen Gao, Fanhang Man, Jinqiang Cui, Xin Wang, Xinlei Chen, Yong Li, and Wenwu Zhu. Embodied-r: Collaborative framework for activating embodied spatial reasoning in foundation models via reinforcement learning, 2025. URL https://arxiv.org/abs/2504.12680.

[90] Jinliang Zheng, Jianxiong Li, Sijie Cheng, Yinan Zheng, Jiaming Li, Jihao Liu, Yu Liu, Jingjing Liu, and Xianyuan Zhan. Instruction-guided visual masking. *arXiv preprint arXiv:2405.19783*, 2024.

[91] Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 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)*, Bangkok, Thailand, 2024. Association for Computational Linguistics. URL http://arxiv.org/abs/2403.13372.

[92] Yaowei Zheng, Shenzhi Wang Junting Lu, Zhangchi Feng, Dongdong Kuang, and Yuwen Xiong. Easyr1: An efficient, scalable, multi-modality rl training framework. https://github.com/hiyouga/EasyR1, 2025.

[93] Qiji Zhou, Ruochen Zhou, Zike Hu, Panzhong Lu, Siyang Gao, and Yue Zhang. Image-of-thought prompting for visual reasoning refinement in multimodal large language models, 2024. URL https://arxiv.org/abs/2405.13872.

[94] Richard Zhuang*, Trung Vu*, Alex Dimakis, and Maheswaran Sathiamoorthy. Improving multi-turn tool use with reinforcement learning. https://www.bespokelabs.ai/blog/improving-multi-turn-tool-use-with-reinforcement-learning, 2025. Accessed: 2025-04-17.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction clearly state our central claim. These claims are supported by empirical results across multiple domains, including human evaluations.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The appendix explicitly discusses limitations.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [NA]

Justification: This paper does not present new theoretical results or proofs; all contributions are empirical and algorithmic.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We provide detailed descriptions of our experimental setup, model variants, training procedures, and evaluation protocols in Sections 4–6 and Appendix, ensuring reproducibility. We also release anonymized code.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: We include anonymized links in the supplemental material with all code, trained models, and instructions to reproduce experiments, as well as generated datasets used in our experiments.

Guidelines:

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

6. **Experimental setting/details**

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

Answer: [Yes]

Justification: Section 4 and Appendix provide complete details on model configurations, data splits, MCTS parameters, training settings, and evaluation metrics.

Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [Yes]

Justification: We report 95% confidence intervals for all performance metrics.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We specify GPU configuration, and estimated per-run and total training costs in Appendix.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We adhere to the NeurIPS Code of Ethics throughout the research. No ethical violations were encountered.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Section 6 and Appendix discusses broader impacts.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [Yes]

    Justification: To prevent misuse, we restrict downstream deployment via usage instructions in codebase, and impact statements in the paper.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We cite and respect the licenses for all datasets and models used (see references). All third-party assets are publicly licensed.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: We release documented anonymized code, with well-documented readmes for training and data. We will release all code models upon publication.

    Guidelines:

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

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

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

    Answer: [Yes]

    Justification: We conducted a user study on Prolific and include full HTML instructions, compensation details, and example screenshots in Appendix. All participants were paid significantly above minimum wage.

    Guidelines:

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

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

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

    Answer: [Yes]

    Justification: Risks were minimal and clearly disclosed to participants prior to consent (see Appendix).

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [Yes]

    Justification: The core method involves using vision-language models (e.g., Qwen2-VL).

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.

6 Conclusion and Discussion

Why does visual grounding help? Our findings suggest that spatially anchoring each reasoning step forces the model to engage in a more structured, human-aligned form of cognition. ViGoRL learns to iteratively reference, inspect, and verify content in specific visual regions – amplifying cognitive behaviors such as subgoal formulation, visual verification, and backtracking.

This model architecture mirrors insights from cognitive science: humans rely on spatial attention and visual routines to decompose complex problems into manageable, perceptually grounded steps [64, 4]. Grounding serves not merely to reduce computational load (as human spatial attention is often characterized), but to scaffold reasoning with external visual structure – effectively using the content of the world as part of the thinking process [80, 22]. We observe similar benefits in models: spatial grounding enables better generalization, especially in out-of-distribution settings, and improves interpretability by making intermediate steps physically referable.

Rather than treating grounding as a visualization tool or auxiliary supervision signal, our results argue for it as a central architectural and algorithmic principle. By training models to reason with deictic reference – pointing, zooming, verifying – future systems may better reflect the iterative, grounded strategies that underlie human problem-solving. This opens up promising directions for building agents that not only reason effectively, but in ways that are queryable, adaptable, and aligned with perceptual experience.

A1 Appendix Overview

The structure of this Appendix is as follows:

- Section A2 contains limitations and future work.
- Section A3 contains broader impacts.
- Section A4 contains additional discussion and connection to cognitive science.
- Section A5 contains additional methods details.
- Section A6 contains additional details on the model behavioral analysis.
- Section A7 contains additional details on the human evaluation.
- Section A8 contains additional experimental results.
- Section A9 contains example model outputs.

A2 Limitations and Future Work

While our approach demonstrates strong improvements in visual reasoning performance, several limitations remain and open avenues for future research:

Intermediate Reward. Our current reinforcement learning setup provides rewards only at the final answer level. In spatial reasoning tasks, this sometimes results in reward hacking, where the model receives positive reward despite generating partially incorrect or underspecified reasoning. Although our human evaluation confirms that the majority of grounding predictions are accurate and helpful, there remains measurable room for improvement. We hypothesize that introducing *dense intermediate rewards*—for both correct reasoning steps and accurate grounding—could mitigate these issues and improve alignment between grounding, reasoning, and reward. We leave this exploration to future work.

Expanded Tool Use and Adaptive Control. While our model can point, crop, and zoom into visual regions, other forms of tool-augmented perception (e.g., highlighting, region comparison, or 3D navigation) may further support compositional reasoning. Future work could explore mechanisms that learn to balance visual actions dynamically.

Learning When and How Much to Reason. We observe that the model often generates long reasoning chains, even for relatively simple questions. This inefficiency echoes known challenges in language-only chain-of-thought models [67, 32], where reasoning is applied uniformly regardless of

task complexity. A promising direction for future work is to develop methods that help models learn *when* to reason and *how much* reasoning is necessary, adapting the depth of their chain-of-thought to the demands of the task.

Interpreting Attention Patterns. Although our behavioral and human studies demonstrate that explicit grounding amplifies structured visual behaviors and improves interpretability, the mechanisms through which grounding improves reasoning remain underexplored. Future work should examine how explicit grounding influences a model's *internal* attention dynamics and whether these patterns resemble human gaze behaviors. Further comparisons to human cognitive strategies—such as visual routines [64], task-directed attention [80], and deictic planning [4]—could yield deeper insights into both model behavior and human cognition.

A3 Broader Impacts

Positive Societal Impacts. Grounded reasoning improves model transparency by producing interpretable chains of thought that reference specific image regions. This interpretability is especially valuable in high-stakes domains such as medical imaging, accessibility tools, assistive robotics, and scientific image analysis, where understanding the basis for model decisions is crucial. Additionally, our approach may enhance human-AI collaboration by making the model's internal decision process legible and actionable for human users. Because the model reasons over localized visual subproblems, it can better align with human intuition, which may foster trust and oversight.

Potential Negative Impacts. Despite its benefits, the ability to produce detailed visual reasoning traces could be misused for surveillance or automated profiling. A grounded model that can isolate visual elements, track reasoning over time, and justify actions may inadvertently enable more fine-grained tracking of individuals or objects in sensitive contexts. Moreover, models that appear interpretable may be perceived as more reliable than they truly are; if the reasoning trace is superficially plausible but incorrect, users may over-trust the model's output. This risk is particularly relevant in domains where visual ambiguity or annotation bias affects ground truth.

Our approach also assumes the availability of visual data and may propagate or amplify dataset biases, especially in domains where certain object types, environments, or affordances are overrepresented. Spatial grounding mechanisms may lock the model's attention onto biased or stereotyped regions of the image, reinforcing existing disparities unless care is taken in dataset construction and evaluation.

Mitigation Strategies. To mitigate these risks, future work could incorporate uncertainty estimation or confidence calibration into the reasoning trace, explicitly marking when the model is uncertain about a grounding decision. It is also important to develop tools that allow users to audit the reasoning trace and provide corrections. In addition, efforts to diversify training data and evaluate model behavior across demographic and situational axes are essential for fair deployment. Finally, deployment in sensitive settings should involve human-in-the-loop oversight, particularly when grounded outputs inform consequential decisions.

A4 Additional Discussion

Our results show that explicitly grounding reasoning steps in spatial coordinates substantially improves performance across visual reasoning benchmarks. To better understand why, we draw from both the recent machine learning literature and long-standing cognitive science theories that point to grounding as a core mechanism in human and artificial reasoning.

Grounding as a Cognitive Scaffold. Classic studies in visual cognition suggest that humans reason about complex scenes by sequentially directing attention to spatially localized regions in service of goal-driven subproblems. Yarbus [80] demonstrated that eye movement patterns depend heavily on task demands, suggesting a deep link between overt visual attention and internal reasoning. Ullman [64] introduced the concept of *visual routines*, showing that reasoning over a scene involves decomposable, spatially localized operations. Ballard et al. [4] formalized this via *deictic codes*, where visual fixations and gestures act as pointers binding abstract variables to perceptual content. These

mechanisms reduce *cognitive load* (as opposed to raw computational load), support compositionality, and facilitate subgoal execution by grounding symbolic reasoning in visual input.

Grounding Reduces Hallucination and Enhances Generalization. In machine learning, recent studies demonstrate that grounding in visual regions curbs hallucination and promotes generalization. Ghosh et al. [16] show that large VLMs hallucinate less and reason more accurately when first prompted to generate grounded visual descriptions. Qi et al. [46] propose a chain-of-manipulations method where VLMs perform spatially localized actions (e.g., zoom, crop, verify), achieving stronger generalization and interpretability. These findings echo Treisman's seminal Feature Integration Theory [63], which argues that spatial attention is essential for integrating visual features, and Harnad's symbol grounding problem [22], which proposes that abstract reasoning is only meaningful when connected to perceptual representations.

Object-centric and Compositional Inductive Biases. Explicit grounding may also encourage models to adopt object-centric and modular representations. Ding et al. [11] show that attention over learned object embeddings improves structured visual reasoning. Gupta et al. [20] demonstrate that language-guided neural modules that condition on spatial cues improve the compositional generalization of the model. These results align with the cognitive perspective that humans naturally build scene representations around individual entities and their relations [4].

Overall, the convergence of cognitive science and ML literature suggests that explicit grounding does not merely reduce the search space, but acts as a structural inductive bias that enhances compositionality, verification, and goal-directed reasoning, core ingredients for generalization in both human and machine learners.

A5 Additional Methodological Details

A5.1 Group Relative Policy Optimization (GRPO)

Group Relative Policy Optimization (GRPO) [52] stabilizes policy learning from long-form trajectories by computing group-normalized advantages and applying clipped token-level PPO-style updates.

Specifically, for a group of G trajectories $\mathcal{O} = \{\tau^{(i)}\}_{i=1}^{G}$ conditioned on input x, each trajectory $\tau^{(i)}$ has a scalar reward $r^{(i)} = R(\tau^{(i)})$. GRPO computes the centered advantage $\hat{A}^{(i)} = r^{(i)} - \bar{R}$, where $\bar{R} = \frac{1}{G}\sum_i r^{(i)}$ is the group mean.

Let $\tau_t^{(i)}$ be the t-th token of trajectory $\tau^{(i)}$. GRPO minimizes the following clipped surrogate loss:

$$\mathcal{L}_{\text{GRPO}}(\theta) = -\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|\tau^{(i)}|}\sum_{t}\min\left[\rho_t^{(i)}\hat{A}^{(i)}, \text{clip}(\rho_t^{(i)}, 1-\varepsilon, 1+\varepsilon)\hat{A}^{(i)}\right] + \beta\,\text{KL}[\pi_\theta\|\pi_{\text{ref}}], \quad (1)$$

where $\rho_t^{(i)} = \frac{\pi_\theta(\tau_t^{(i)}|\tau_{<t}^{(i)},x)}{\pi_{\text{old}}(\tau_t^{(i)}|\tau_{<t}^{(i)},x)}$ is the importance weight, $\varepsilon = 0.2$ is the clipping parameter, and β is the KL penalty coefficient. This approach stabilizes learning in long-horizon, multimodal reasoning settings.

A5.2 RL Reward Functions

Task Rewards. We define reward functions specific to each benchmark:

- **Spatial Reasoning (SAT-2).** A binary reward: $r_{\text{task}} = 1$ if the predicted answer matches the ground truth, and 0 otherwise.
- **Web Grounding (OS-Atlas).** A binary reward: $r_{\text{task}} = 1$ if the predicted coordinate lies inside the annotated bounding box.
- **Web Action Prediction (ICAL).** A decomposed reward: $r_{\text{task}} = r_{\text{type}} + r_{\text{arg}}$, where $r_{\text{type}} = 0.5$ if the predicted action type matches, and $r_{\text{arg}} = 0.5$ if the predicted argument (e.g., DOM ID, string) matches.

A5.3 Multi-turn Reinforcement Learning Details

We first apply supervised fine-tuning (SFT) on multi-turn traces. These are derived from the same MCTS chains used to train the single-turn model, but reformatted into multi-turn training data by:

(1) For each node, formulating the text in a think block followed by taking the coordinate and formulating it into a tool call. If it is a terminal node, a think block is added with any remaining text followed by the predicted answer in answer tags.

(2) Cropping around the coordinate to obtain the feedback image, and appending this image to the training data as a user turn in the sample to be used for SFT.

(3) Continuing (1) and (2) until terminal. We additionally include backtracking as described in Section 4.2.

For multi-turn scenarios, we apply GRPO over full dialog trajectories. Observation tokens are masked from the loss, ensuring gradients flow solely through the language model while retaining visual input to the encoder. We additionally mask samples not ending in an `EOS` token [82].

Termination Enforcement. During RL rollouts, if a dialog reaches $T_{\max} = 5$ turns without emitting an `<answer>` block, we append a soft prompt to the final assistant message:

```
<think> Please provide your response now </think>
```

This maintains structural fidelity and bounds rollout length.

KL collapse. While concurrent work observed occasional KL collapse in multi-turn RL when applied to base model with vision inputs [66, 94], we found that our initial warm start enabled stable training and allowed us to maintain a moderate KL coefficient (0.01).

Diversity bonus. We introduced the concept of a diversity turn bonus into our reward formulation in Section 4.3 of the main paper. In Figure A1, we show the response length over RL training with and without the diversity bonus. Without the diversity bonus, the model quickly converges to single turn outputs (green line), whereas the model avoids this and outputs >1 turn on average with the bonus reward (blue line).

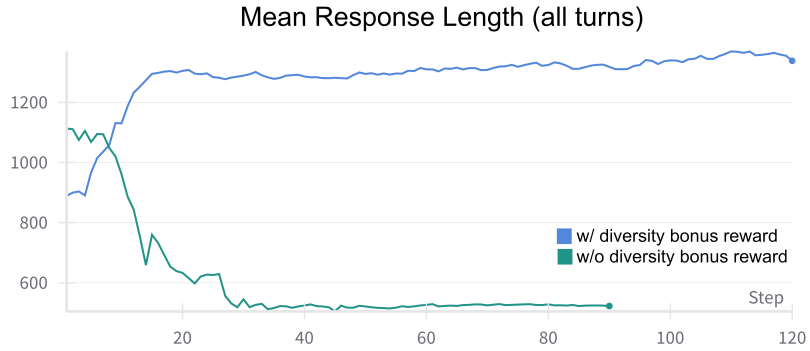


Figure A1: Response length with and without turn bonus. Without the bonus, the model converges to always taking a single turn (as also verified by examining model outputs), whereas the bonus enables the model to stabilize multi-turn.

A5.4 Training Implementation Details

General Setup. Training is conducted on 8 A100 GPUs with Qwen2.5-VL models (3B and 7B). Supervised fine-tuning uses 3 epochs, while GRPO is applied for 500 rollout-update iterations. Evaluation uses decoding temperature of 0.5. We build on Llama-Factory [91] for SFT, and EasyR1 [92, 54] for GRPO.

Training Hyperparameters Overview We provide all training hyperparameters used in SFT, and single- and multi-turn RL in Tables A1-A3.

Table A1: Supervised Fine-Tuning (SFT).

Hyperparameter	Value
Epochs	3
Learning rate	1e-6
Weight decay	0.01
Warmup ratio	0.03
Batch size	8
Gradient accumulation	4
Effective batch size	32
Scheduler	Cosine
Precision	bf16
Flash attention	fa2
Freeze vision tower	True
Max sequence length	8192
Deepspeed config	ZeRO Stage 3

Table A2: GRPO Training.

Hyperparameter	Value
Training steps	500
Learning rate	1e-6
Weight decay	0.01
Warmup ratio	0.05
Optimizer	AdamW (bf16)
Group size	5
KL coefficient	0.01
Clip ratio	0.28
Gradient clipping	Max norm 1.0
Rollout batch size	128
Gradient step batch size	32
Rollout engine	vLLM
Max prompt length	4096
Max response length	2048
Top-p	0.99
Temperature	1.0
Filter overlong	True
Freeze vision tower	True
Mixed precision	bf16

Table A3: Multi-turn GRPO.

Hyperparameter	Value
Max prompt length	4096
Max response length (includes observations)	4096
Max generation per turn	1024
Max turns	5
Crop resize	384x384
Crop size	100x100
Learning rate	1e-6
KL coefficient	0.01
Weight decay	0.01
Warmup ratio	0.05
Clip ratio	0.28
Gradient clipping	Max norm 0.2
Rollout batch size	128
Gradient step batch size	64
Training steps	500
Group size	8
Rollout engine	vLLM
Temperature	1.0
Top-p	0.99
Mixed precision	bf16
Freeze vision tower	True

A5.5 MCTS Implementation Details

To generate high-quality reasoning traces for fine-tuning, we use Monte Carlo Tree Search (MCTS) to explore possible sequences of reasoning steps over an image and question. Our MCTS procedure extends standard tree search with a VLM as the generative model for reasoning steps and a judge to evaluate final answers. We use the same reward as described in Section A5.2.

Search Structure. Each node in the search tree corresponds to a reasoning step (a candidate `<think>` or `<answer>` block). The root node contains the input question. Nodes maintain the text of the current step, accumulated visited coordinates, a running estimate of expected reward, and visit count.

Algorithm Phases. MCTS operates via the standard four-phase loop:

- **Selection:** Starting at the root, the search follows a path through children using the UCB policy: $Q + c\sqrt{\log N/n}$, where Q is the average reward, N is the parent's visit count, n the child's, and $c = 2.0$ is the exploration constant. The search terminates when a node has unvisited children or reaches a terminal state.
- **Expansion:** At each expandable node, we generate up to three children using the VLM, each corresponding to a new thought or final answer. If a generated step includes a terminal `<answer>` block, it is marked as terminal. An example prompt for this step can be found in Listing A1.
- **Rollout:** For each new child, we simulate reasoning steps using the VLM until a final answer is reached or a rollout depth limit is exceeded. These simulated trajectories are not added to the tree. Each rollout receives a scalar reward from the judge model comparing the predicted and true answers.
- **Backpropagation:** Final 0/1 rewards are backpropagated up the tree along the visited path, updating the value estimate of each node using an incremental mean and incrementing visit counts. We use the same task reward as described in Section A5.2.

We include our MCTS hyperparameters in Table A4.

A5.5.1 Linearization of Search Trees into Reasoning Chains

To convert MCTS-generated search trees into training data for supervised fine-tuning, we developed a structured linearization procedure that extracts diverse, grounded reasoning trajectories from the tree.

Tree Traversal and Chain Extraction. Each MCTS trace is stored as a tree, with nodes representing reasoning steps (`<think>`) or final answers (`<answer>`). From each trace, we recursively enumerate all root-to-leaf paths that include at least one terminal node. For each terminal node, we extract:

- **Correct rollouts**: Paths ending in a high-reward (≥ 1.0) final answer.
- **Incorrect rollouts**: Lower-reward paths used to generate synthetic backtracking examples.

If both correct and incorrect rollouts exist at a node, we concatenate the incorrect trace with a fixed backtracking phrase (e.g., "Wait, this seems off. Let's try something else.") followed by the corrected reasoning path, forming a complete trace with embedded revision behavior. All steps are wrapped in a single `<think>` block, and the final answer in a separate `<answer>` tag.

Token Cleanup and Deduplication. Each reasoning trace is cleaned to remove residual XML markers (`<think>`, `<answer>`) before joining, then wrapped again after concatenation. Chains are deduplicated across samples to prevent redundancy in training.

MCTS accuracy. As shown in Table A5, we observe improved top-1 accuracy using our MCTS procedure.

Table A4: MCTS hyperparameters for Web Grounding.

Hyperparameter	Value
Model	Qwen2.5-VL-72B-Instruct
Simulations per input	20
Max tree depth	10
Rollouts per node	2
Children per expansion	3
c_{puct}	2.0
Sampling temperature	1.0
Top-p	1.0
Max tokens per node	512
Estimated time per sample	21 minutes
Parallel processes	10

A6 Behavioral Analysis Protocol

To systematically evaluate reasoning behaviors in VLMs, we implement a behavioral annotation pipeline that categorizes model-generated chain-of-thought (CoT) traces using GPT-4o as an evaluator. This procedure is based on Gandhi et al. [15] and enables fine-grained analysis of emergent reasoning strategies across different model variants and training regimes.

Sample Selection and Preprocessing. We run the same 300 reasoning traces (obtained by randomly selecting examples from the SAT-2 validation set) per model condition where the model's final answer was verified correct (`judge_score` = 1). From each trace, we isolate the contents of the `<think>` block.

Behavioral Categories. We define four behavioral dimensions of interest:

- **Visual Verification:** Instances where the model confirms or checks a property of the visual scene (e.g., "Looking at the image, I can confirm…").
- **Backtracking:** Self-correction or re-interpretation of a previously described visual element.
- **Subgoal Setting:** Decomposition of the visual reasoning process into smaller steps across regions (e.g., "First I will check X, then Y…").
- **Visual Regions Explored:** Count of distinct visual regions explicitly referenced in the reasoning trace.

Annotation via GPT-4o. For each trace, we construct four behavior-specific prompts and submit them to GPT-4o with temperature 0.0 and a max token limit of 256. Each prompt asks the model to identify and count instances of a specific behavior, outputting a numeric value between custom '<count>' tags. These counts are extracted and recorded as the behavioral profile for the trace. Our full prompts are displayed in Listing A2.

Comparison Across Conditions. We apply the above process across multiple model variants (e.g., Our full model, Naive GRPO, Ablated grounded reasoning) and aggregate the behavior counts to compute the average number of reasoning behaviors per trace.

A7 Human Evaluation Setup

To assess the interpretability and spatial accuracy of the model's grounded visual reasoning outputs, we conducted a structured human evaluation study using Prolific. The study was designed to evaluate whether the (x,y) coordinate output by the model accurately corresponded to the referenced region in the reasoning step, and whether this visual cue helped participants interpret the reasoning step. We obtained 80 samples randomly from the robospatial evaluation, as this evaluation contained real images without visual marks on the image.

Data curation. We take 80 samples from our model run on Robospatial, and extract each sentence from the samples. To simplify the study for the participants and target step-level analysis, we filter out sentences that require context from the entire reasoning trace using GPT4o. For each reasoning step and coordinate, we draw a 100x100 pixel blue circle on the image centered at the coordinate location, and display the reasoning step text.

Study Design. Each trial presented participants with a single image containing a blue circle annotation, along with a natural language sentence that included the phrase "[shown with blue circle]" (which replaced the (x,y) coordinate) to denote the region being referenced. Participants answered two questions per image:

1. **Accuracy (binary + unsure):** Participants were asked whether the blue circle overlapped with the region described by the sentence. Options were:
   - **Yes** — if any part of the described region was inside the blue circle.
   - **No** — if the region was entirely outside the circle.
   - **Unsure** — if the sentence was ambiguous or the region could not be clearly judged. Responses marked as "Unsure" were excluded from accuracy and clarity score calculations to avoid inflating or deflating agreement metrics with ambiguous judgments.
2. **Interpretability (Likert):** Participants rated how much the blue circle helped them understand the sentence reference on a 5-point Likert scale:
   - 1 — Strongly disagree
   - 2 — Disagree
   - 3 — Neutral
   - 4 — Agree
   - 5 — Strongly agree

Interface and Instructions. Participants began the study by entering a participant ID. A detailed instruction panel introduced the task goals, decision criteria, and included two illustrative examples with images. These examples demonstrated both "Yes" and "No" cases for spatial accuracy, and clarified how to interpret the clarity rating.

Participant Recruitment and Demographics. We recruited participants via the Prolific platform. A total of 20 participants completed the study, with all submissions manually reviewed and approved. The participant pool was demographically diverse: ages ranged from 21 to 71 (mean = 39.4, SD = 13.6), with balanced gender representation (11 female, 9 male). Participants were based in the United States or the United Kingdom, and all reported fluency in English. Participants completed the task in an average of 22 minutes, and were compensated at a rate of $12/hr.

A8 Additional Experimental Results

MCTS accuracy on OS-Atlas and SAT-2 We evaluate the accuracy of our MCTS procedure using Qwen2.5-VL-72B by measuring how often it reaches the correct answer, as determined by an oracle verifier. On 123 held-out samples from OS-Atlas (Table A5), MCTS achieves an accuracy of 82.1%, indicating strong search effectiveness when guided by a reliable verifier. On the SAT-2 benchmark, MCTS reaches 96% accuracy.

ScreenSpot-Pro Low Resolution Results. We provide results on ScreenSpot-Pro-LR in Table A6. This variant of ScreenSpot-Pro has downsampled images to a max resolution of 1920x1920. On ScreenSpot-Pro-LR, the base Qwen2.5-VL-3B model achieves only 1.96% accuracy, while SFT improves performance to 16.89%. Incorporating GRPO further increases accuracy to 20.30%, and our full ViGoRL achieves 21.32%. Notably, our multi-turn ViGoRL reaches 23.72%, demonstrating that iterative visual feedback is especially beneficial in challenging perceptual settings.

Accuracy on Out-of-Distribution Benchmarks. To ensure that our model does not lose foundational knowledge after our pipeline, we evaluate on some popular VQA benchmarks: MMMU [83], RealWorldQA [73], and V* [69]. On MMMU and RealWorldQA, we evaluate our model's performance when prompted without thinking, thus performing an apples-to-apples comparison to the base model. On V*, we prompt our model to think to compare to the V* visual search paradigm. On all datasets, we find that ViGoRL either matches or exceeds the performance of the base model.

Splitting versus combining training datasets. To quantify the effect of combining data, we conducted an experiment combining spatial and web datasets during both warm-start and RL phases. The results, shown in Table A8, reveal:

- Web grounding: Small decrease (-0.2% on ScreenSpot-Pro, -0.3% on ScreenSpot-V2)

- Spatial reasoning: Improvement of +1.1% on SAT-2

- Both approaches significantly outperform vanilla GRPO (+12-15% SAT-2)

These findings indicate that combining data has small effects on relative performance gains from our method, with slight benefits for spatial reasoning at the cost of web grounding performance.

Euclidean distance reward for grounding leads to lower accuracy. As shown in Table A9, we tested a Euclidean distance reward, instead of binary reward, for grounding and found it significantly reduced performance: -2.2% on ScreenSpot-V2 and -0.7% on ScreenSpot-Pro (Table R3).

Web elements vary in size, so a distance-based reward unfairly penalizes valid clicks near the edges of large elements. For instance, clicking near the edge of an 800×60 navigation bar is functionally correct but would receive a low euclidean reward.

Comparison to teacher Qwen2.5VL-72B model We show zero-shot performance of our teacher model (Qwen2.5-VL-72B) in Table A10 below. Despite using only 1k examples for distillation before RL, ViGoRL-7B outperforms Qwen2.5-VL-72B by 6.0% on SAT-2 while being 10× smaller, and closes the gap across most benchmarks.

Table A5: Held-out performance of Qwen2.5-VL-72B on 123 samples in OS-Atlas validation and 958 samples in SAT-2 validation. * indicates MCTS Accuracy of the hold-out MCTS set used in the training pipeline.

Model	Top-1 Accuracy	Top-3 Accuracy	MCTS Accuracy
OS-Atlas	42.3%	54.5%	82.1%
SAT-2	61.48%	76.30%	96.0%*

Table A6: Accuracy (mean with 95% confidence intervals) on ScreenSpot-Pro-LR benchmark.

Model	ScreenSpot-Pro-LR
Qwen2.5-VL-3B Base	1.96% (±0.68)
+ SFT direct	16.89% (±1.85)
+ Vanilla GRPO	20.30% (±2.24)
ViGoRL-3b (Ours)	21.32% (±2.28)
ViGoRL-3b (Multi-turn)	**23.72%** (±2.39)

Table A8: Accuracy (mean ± 95% CI) on ScreenSpot-Pro/V2 and SAT-2.

Condition	ScreenSpot-V2/Pro	SAT-2
Qwen2.5-3B-VL	23.9 (±1.4) / 68.4 (±2.6)	46.1 (±1.5)
Vanilla GRPO (seperate spatial or web)	29.0 (±2.2) / 84.4 (±2.0)	50.0 (±1.6)
ViGoRL-3B (seperate spatial or web)	31.1 (±2.3) / 86.5 (±1.9)	62.9 (±1.5)
ViGoRL-3B (combined spatial+web)	30.9 (±2.3) / 86.2 (±2.0)	64.0 (±1.5)

Table A9: Euclidean vs. binary reward ablation. Accuracy (mean ± 95% CI) on ScreenSpot-V2 and ScreenSpot-Pro.

Condition	ScreenSpot-V2	ScreenSpot-Pro
Qwen2.5-3B-VL	68.4% (±2.6)	9.3% (±1.4)
ViGoRL-3B w/ binary bbox reward	86.5% (±1.9)	31.1% (±2.3)
ViGoRL-3B w/ euclidean reward	84.3% (±2.0)	30.4% (±2.3)

Table A7: Accuracy (mean with 95% confidence intervals) on Out-of-Distribution benchmarks. * indicates that the model was prompted to think.

Model	MMMU	RealWorldQA	V* Bench
Qwen2.5-VL-3B Base	47.44% (±3.26)	55.65% (±3.52)	74.21% (6.22)
ViGoRL-3b (Ours trained on SAT-2)	46.44% (±3.26)	60.65% (±3.46)	74.87% (±6.15)*
ViGoRL-3b (Ours trained on UGround)	47.56% (±3.26)	57.52% (±3.50)	75.13% (±6.16)*

A9 Example Model Outputs

A9.1 Failure Cases in Current VLMs

Despite high image-text alignment, models like Qwen2.5-VL-7B fail often produce abstract, shallow reasoning without examining many different image regions or describing content in visual depth, which often leads to errors. A representative trace is below:

<think> From the image, the twotiered metal cart is positioned closer to the modern TV stand in blue. The cart is nearer to the TV stand, which is evident from its proximity and the angle at which it is seen. Therefore, the twotiered metal cart is closer to the modern TV stand in blue. </think> <answer>twotiered metal cart</answer>

A9.2 Warm Start Reasoning Traces

Fine-tuning with MCTS-generated warm traces leads to structured and grounded reasoning. Example trace:

Table A10: Comparison to Qwen2.5-72B-VL. Accuracy (mean ± 95% CI) across benchmarks.

Model	SAT-2	BLINK	RoboSpatial	ScreenSpot-V2	ScreenSpot-Pro
Qwen2.5-VL-72B	61.5% (±1.5)	59.5% (±2.3)	55.7% (±6.5)	88.1 (±1.8)	34.2 (2.6±)
ViGoRL-7B	67.5% (±1.5)	54.1% (±2.3)	76.4% (±7.5)	91.0% (±1.6)	33.1% (±2.3)

Table A11: ViGoRL error analysis by type on SAT-2. Percent of failures by category.

Error Type	Percentage	Description
Excessive Reasoning	20%	The model overthinks the problem.
Incomplete Reasoning	10%	The model truncates its reasoning process.
Hallucination	10%	The model invents visual elements.
Challenging Visual QA	30%	The question or image presents inherently subtle cues, causing reasoning errors.
Base Perception	30%	The underlying model makes perceptual or conceptual grounding errors.

<think>
I need to locate Candice Georgiadis in the list. I see the name Candice Georgiadis at (330, 440). Now I will check for the 'Follow' button next to it. Next to Candice Georgiadis's name at (330, 440), I see a 'Follow' button at (440, 440). I will verify if there are any other mentions of this name. I should check another area to ensure there's no other 'Candice Georgiadis'. There's another sibling list and see the name Candice Georgiadis at (660, 770) but no 'Follow' button is associated with this name. Having verified the name 'Candice Georgiadis' and the 'Follow' button, and ensuring there was no other mention of the name with a follow button, I am confident the desired location is at the first identification at (440, 440).
Wait, this seems off. Let's try something else.
I need to locate Candice Georgiadis in the list. I see the name Candice Georgiadis at (330, 440). Now I will check for the 'Follow' button next to it. Next to the name Candice Georgiadis at (330, 440), I see a button at (365, 450) that looks like it says 'Follow'. I should check another name on the page for similar formatting to confirm this is the correct 'Follow' button. To ensure that this is indeed the correct 'Follow' button, I will check the name John Doe and its corresponding button at (230, 210). This button has the same format as the one I observed for Candice Georgiadis at (365, 450). It reinforces that the button next to Candice Georgiadis is the one I need. Lastly, to eliminate all doubt, I will examine a different region of the image where the 'Follow' button for Fred Smith at (500, 600) is located. It aligns with the same design as the others, confirming the earlier observations. Therefore, I can now finalize the answer for Candice Georgiadis's 'Follow' button.
</think>

A9.3 Additional Model Output Examples

We provide example outputs on RoboSpatial questions for two models: the vanilla GRPO model trained on SAT-2 (Figure A2), and our proposed ViGoRL trained on SAT-2 (Figure A3).

A9.4 Failure cases

We categorizes SAT-2 errors in Table A11. 60% arise from perceptual or task-level challenges, while 40% reflect residual reasoning flaws (e.g., overthinking, truncation, hallucination). We believe scaling or denser rewards could address aspects of both, with further discussion in Section 6 and Appendix A2.

Example of overthinking in ViGoRL for simple question.
Question: "Is there a pot in the image?"

Image: *Image of a stove with a large pot and pan on it.*

Model response:

<think>
To determine if there's a pot, I'll check the regions where pots typically appear. At (300, 600), there's a pot-like object with a handle and a lid, suggesting it could be a pot. To confirm the presence of a pot, I'll check another region. At (700, 1000), there's no pot-like object present, indicating that the previous observation was accurate. Now, I'll verify if there are any other potential pots in the image. At (500, 800), there's no pot-like object present, which supports the previous findings. Next, let's examine the area at (400, 900) to ensure there's no hidden pot. There's still no pot-like object visible here, reinforcing the previous observations. I'll now check the lower region at (600, 1200) to confirm there are no additional pots. At (600, 1200), no pot-like object is visible here, further supporting the absence of a pot in the image. Lastly, I'll check the upper region at (300, 200) to ensure there's no hidden pot. At (300, 200), no pot-like object is visible here, confirming the pot is at (300, 600) in the image.
</think>

Listing A1: MCTS prompt for web grounding.

```
You are a helpful assistant tasked with grounding an element on a web page. You should
    systematically reason through the problem step by step by checking and verifying
    relevant webpage regions, while grounding reasoning steps to specific (x, y) points
    in the image:\nEach reasoning step must be enclosed within '<think>' tags and
    reference exactly one specific coordinate (x, y):\n<think>\n{Single reasoning step
    with a grounded point} (x, y).\n</think>\nWhen ready to provide the final answer,
    enclose it within '<answer>' tags:\n<answer> (xf, yf) </answer>\nYour task is to
    help the user identify precise (x,y) coordinates of a described area/element/object
    based on a description.\n- Generate ONLY ONE reasoning step OR the final answer per
    response.\n- Regions are distinct, non-overlapping areas (e.g., quadrants like top-
    left, elements like tree/button, zones like background/foreground).\n- Each step
    should describe the region then evaluate it for its relevance to the task and to
    previous steps.\n- Never repeat coordinates from previous steps.\n- Begin by
    exploring diverse regions, even if they seem less likely, to ensure comprehensive
    coverage before narrowing down.\n- Prioritize broad coverage of diverse candidates
    before deciding.\n- Aim for accurate, representative points in the described area/
    element/object.\n- If unclear, infer based on likely context or purpose.\n- Verify
    each step by examining multiple possible solutions before selecting a final
    coordinate.\n- Format points as (x, y)
```

Listing A2: Prompts used to evaluate visual reasoning behaviors in chain-of-thought outputs.

```
Here is a chain-of-reasoning that a Language Model generated while analyzing an image.
    The chain-of-reasoning output from the model is:
'''
{reasoning}
'''
Evaluate whether the chain-of-reasoning contains any visual verification steps. A visual
    verification step is when the model confirms or checks something it sees in the
    image. Examples include: "I can see that the object is not a cat, but a dog", "The
    text confirms this visual aspect is correct", "I can verify this is indeed red", or
    "Looking at the image, I can confirm...". Count both explicit mentions of image
    regions and implicit verifications.
Count all instances where the model verifies information from the image and provide the
    count between the tags <count> </count>. If the chain-of-reasoning does not contain
    any visual verification steps, please provide a count of 0 as <count>0</count>.

---

Here is a chain-of-reasoning that a Language Model generated while analyzing an image.
    The chain-of-reasoning output from the model is:
'''
{reasoning}
'''
Evaluate whether the chain-of-reasoning contains any backtracking behavior, where the
    model changes its interpretation or corrects itself. Examples include: "At first I
    thought X, but looking more carefully I see it's actually Y", "I initially
    interpreted this as a circle, but it's actually an oval", "On second thought...", "
    Actually, I notice that...", or "Let me correct my earlier observation...".
Count all instances where the model revises its understanding and provide the count
    between the tags <count> </count>. If the chain-of-reasoning does not contain any
    backtracking behavior, please provide a count of 0 as <count>0</count>.

---

Here is a chain-of-reasoning that a Language Model generated while analyzing an image.
    The chain-of-reasoning output from the model is:
'''
{reasoning}
'''
Evaluate whether the chain-of-reasoning contains any visual subgoal setting, where the
    model breaks down the image analysis into smaller steps or focuses on different
    parts of the image in sequence. Examples include: "First, I'll examine this part,
    then I'll look at that object", "Let me check each element one by one", "I need to
    identify what's in this area", or any structured approach to analyzing different
    parts of the image.
Count all instances where the model sets up a plan or approach for analyzing the image
    and provide the count between the tags <count> </count>. If the chain-of-reasoning
    does not contain any visual subgoal setting, please provide a count of 0 as <count
    >0</count>.

---

Here is a chain-of-reasoning that a Language Model generated while analyzing an image.
    The chain-of-reasoning output from the model is:
'''
{reasoning}
'''
Count how many distinct visual regions or elements the model explicitly mentions
    examining in the image. Examples include: "I can see a dog in the corner", "There's
    text at the top of the image", "The object in the center appears to be...", "Looking
     at the left side...", or any reference to a specific part or element of the image
    that the model is analyzing.
Count all distinct visual regions or elements mentioned and provide the count between the
     tags <count> </count>. If the chain-of-reasoning does not mention any specific
    visual regions, please provide a count of 0 as <count>0</count>.
```

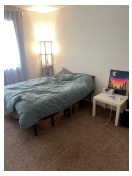
Images	Question	Base Model (SFT Direct)	Naive GRPO Model
	Q. Can the table fit above the bed? Answer yes or no. Answer with the text of the option.. GT Ans :- Yes	Thoughts: Predicted Answer: No	Thoughts: From the image, the table is positioned to the right of the bed. The table top is level with the side of the bed, and the surface of the bed is a foot or two higher. Therefore, the table can fit above the bed. Predicted Answer: Yes
	Q. Can the cactus fit in front of the litter box? Answer yes or no. Answer with the text of the option.. GT Ans :- Yes	Thoughts: Predicted Answer: No	Thoughts: Since the litters are next to the litter box, the cactus can be placed in front of the litter box because it is a long space for plants, and it can accommodate the height from the ground to the cactus. But if the plants are taller, the cactus will not fit in front of the litter box. Predicted Answer: Yes
	Q. Can the pan fit above the fridge? Answer yes or no. Answer with the text of the option.. GT Ans :- Yes	Thoughts: Predicted Answer: No	Thoughts: From the image, the fridge appears to be open and wider than the stove, allowing the pan to be lifted above it. This suggests that the pan can fit above the fridge. Therefore, the answer is "yes". Predicted Answer: Yes

Figure A2: Example reasoning traces from vanilla GRPO on the RoboSpatial dataset, which does GRPO directly on the base model without warm start.

Images	Question	ViGoRL
	Q. Can the pan fit above the fridge?Answer yes or no. GT Ans :- Yes	Thoughts: To determine if the pan can fit above the fridge, I'll need to consider the height of the pan relative to the space available above the fridge. At the point (400, 1100), which is near the top edge of the pan, we can see that the pan covers a significant vertical space. Let's check another point (750, 800) which represents a central point on the back wall behind the fridge. The available vertical clearance above the top of the fridge is clear here. There is no major obstruction like a pipe or beam that would prevent the pan being placed directly against the upper corner wall above the fridge. Checking the proximity at the point (900, 500), which is near the upper shelf of what appears to be cabinets above the refrigerator, shows an empty space that would allow an object like the pan to be placed without obstruction. At the point (650, 1250), the lower confines of the fridge, we note its compact structure. This reinforces the idea that space at the fridge's top would be suitable for the pan's placement provided no other elements are present. Predicted Answer: Yes
	Q. Can we cook rice in the vessel? Answer yes or no. Answer with the text of the option.. GT Ans :- Yes	Thoughts: The black pot on the stove appears suitable for cooking rice. (412,876) The countertop seems clean and free of barriers, suggesting an easy access for cooking rice. (1039,874) The stove looks functional and includes appropriate heating elements for cooking various dishes, including rice. (614,895) The kitchen in the image also has a refrigerator that might store necessary cooking supplies like rice. (43,656) No visible cooking ingredients like rice are present in the image other than the bowl. (412,876) Predicted Answer: Yes

Figure A3: Example reasoning traces from ViGoRL on the RoboSpatial dataset.