ReFAct: Empowering Multimodal Web Agents with Visual and Context Focusing

Supplementary Material

6. Limitations

While our proposed ReFAct framework and the GroundedVQA benchmark demonstrate significant progress in enhancing multimodal web agents' robustness to noise, we acknowledge several limitations that present avenues for future research.

First, the effectiveness of the `Grounding` tool, especially in the plug-and-play setting, is dependent on the base model's intrinsic visual grounding capabilities. As observed in our experiments, models with weaker innate grounding fail to benefit from the active focusing mechanism. Future work could explore methods to instill this grounding capability more efficiently, perhaps through more targeted pretraining.

Second, our current reinforcement learning stage primarily focused on optimizing the `Visual Focusing` capability. The `Memory Focusing` (`Defocus`/`Refocus`) operations, while architecturally supported, were not the primary target of the optimization. A more complex training regimen that simultaneously optimizes both visual and retrieval noise management could yield further performance gains, though it presents a significant challenge in stabilizing the RL process.

Finally, while the GroundedVQA dataset is designed to be challenging with high visual noise, its domain coverage is not exhaustive. The data construction pipeline, though automated, may also inherit biases from the models used during generation. Expanding the dataset to more diverse domains and scenarios would be beneficial for evaluating the generalizability of noise-robust agents.

7. Examples of GroundedVQA Dataset

To provide a clearer understanding of the challenges presented in the GroundedVQA dataset, we provide representative examples from both Level 1 and Level 2. These examples illustrate the high visual noise (i.e., small target entity area) and the necessity of external knowledge retrieval.

As shown in Figure 6, the Level 1 example tasks the agent with identifying a non-salient object to retrieve specific facts. The Level 2 example presents a greater challenge, necessitating the simultaneous localization of two entities followed by multi-step retrieval and associative reasoning.

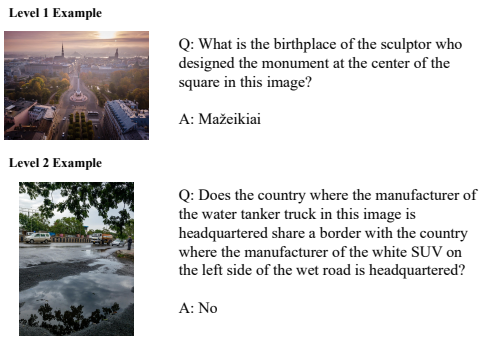

Figure 6. Examples of different difficulty levels from the GroundedVQA dataset. In the Level 1 question, the agent must first identify the central monument before it can retrieve the sculptor's birthplace; in the Level 2 question, it must first recognize the manufacturers of the two vehicles before it can determine whether their headquarter countries share a border.

8. Prompts

This section details the full prompts used for instructing the ReFAct agent, generating the GroundedVQA dataset, and evaluating model outputs with an LLM-as-Judge.

8.1. Data Construction

The construction pipeline involves three steps, with LLM prompts applied in two key phases. The prompts for Phase I are detailed across Table 5, Table 6, and Table 7, while the prompts for Phase II are shown in Table 8.

8.2. LLM-as-Judge

We employed an LLM-as-Judge for evaluating the correctness of the agent's final answers. The prompt is detailed in Table 9.

9. Case Study

To further illustrate the operational flow of the ReFAct agent, we provide detailed case studies. These cases demonstrate the agent's trajectory, including its reasoning, tool usage, and intermediate observations.

9.1. Case 1: Successful Visual Focusing

This case demonstrates the agent's ability to overcome significant visual noise. The query pertains to a small, non-salient object in a complex scene. The ReFAct agent correctly identifies the need for visual focusing, invokes the `Ground` tool, and proceeds on the correct reasoning path.

9.2. Case 2: Multi-Step Reasoning

This case illustrates a more complex, long-horizon task involving multiple entities (Level 2). The agent must retrieve a large volume of text for two different entities.

Table 4. Prompt for ReFAct Agent

Content

Answer the user's question based on the provided image.

Decision-Making Process

1. Analyze and Assess: Carefully examine the image, determine whether you have sufficient knowledge to confidently identify the key visual elements and directly answer the user's question.
2. If Confident (Direct Answer):
• First, explain your reasoning.
• Then, provide a clear and direct final answer.
3. If Not Confident (Search Required):
• Determine the most critical missing piece of information: Is it the identity of a visual element (e.g., "What building is this?") or a piece of factual knowledge (e.g., "When was this building completed?")?
• Choose the best tool for the missing information:
◦ For Visual Identification: Use the image search tool. You can search for only one partial object or the complete image at a time. The output must be: `<image_search>{"bbox_2d": [x1, y1, x2, y2]} </image_search>` or `<image_search>whole </image_search>`
◦ For Factual Knowledge: Use the text search tool. Generate a concise and specific query. The output must be: `<text_search>your query here</text_search>`.
◦ For History Check (Refocus): If previous search steps (e.g., step 1, step 2) have already been executed and you need to retrieve the complete, detailed results (beyond the compressed summary) from that specific step, use the Refocus tool. The output must be: `<refocus>step_N</refocus>`.
4. Reassess After Search: After receiving search results, combine them with the user's question and your existing knowledge. Assess whether you can now confidently answer the question. Repeat the search process if necessary until the question can be answered.

Required Formatting

• Reasoning: Before taking *any* action (invoking a search tool, or providing a final answer), you must explain your reasoning and decision-making process within `<reason>...</reason>`. This reasoning must analyze the image, justify the choice of search tool, interpret search results, or outline the logical steps.
• Tool Calls: If you invoke a tool, it must be enclosed within the appropriate tags:
`<text_search>your query here</text_search>`
`<image_search>{"bbox_2d": [x1, y1, x2, y2]} </image_search>`
`<image_search>whole </image_search>`
`<refocus>step_N</refocus>` (Retrieve complete results of step N)
• Final Answer: When you are ready to answer the question, place your final answer between `<answer>` and `</answer>` *without detailed illustrations*. For example: `<answer>Titanic</answer>`.
• Follow the above formatting exactly!

Here is the image and the question:
{image}{question}

Table 5. Prompts for Phase I: Entity Localization Part

Part	Content
Locate	Task: In the given image, locate as many entities as possible that meet the criteria below and report their coordinates in JSON format. Criteria: • Entity Domains: – Products & Artifacts: Clothing/Accessories (distinctive shoes, bags, watches), Electronics, Vehicles, Furniture, Toys/Collectibles. – Biology & Nature: Animals, Plants, Food (cultural dishes, packaging). – Places & Structures: Architecture (statues, bridges), Natural landscapes. – Brands & Info: Logos (graphic preferred over text), Art (paintings, book covers). • Instance Uniqueness: Select only one instance if multiple entities are identical in appearance. If entities belong to the same category but differ (e.g., different cars), select all. • Visual Searchability: Entities must be visually clear (not blurred/occluded) and possess unique features suitable for visual search engines. Avoid generic textures. Exclusions: • Common Symbols: Traffic signs, currency, and generic icons. • People: Do not select whole ordinary people; focus on their accessories/clothes. Cosplayers are allowed. Specific Constraints: • Shoes: Select only one clear shoe per person. • Quantity: Maximum 3 items per broad category. Output Format: Encapsulate reasoning in <reason> tags. Output strictly in JSON format: [{ “bbox_2d”: [xmin, ymin, xmax, ymax], “label”: “Broad category”, “visual_description”: “Unique visual reference (e.g., ‘the glass pyramid in the center’) in English” }, ...]

Table 6. Prompts for Phase I: Verification Part

Part	Content
Verification	Task: Compare the provided ROI (Region of Interest) image with search result images to determine the fine-grained category of the ROI. Matching Criteria: • A match is valid only if the ROI and search result depict the exact same fine-grained category or model. • Match the type/model, not the specific physical instance. Steps: Compare the ROI with search results focusing on defining features. Strict Rules: • Output ”conflict” if lacking absolute confidence. • Output ”conflict” if the answer is not specific enough. • Output ”conflict” if the answer is generic without specific brand/name details. • Prioritize certainty; prefer ”conflict” over blind confidence. Output Format: <reason> Detailed reasoning and comparison process </reason> <answer> Fine-grained category in English or ”conflict” </answer>

Table 7. Prompts for Phase I: Knowledge Parse Part

Part	Content

Knowledge Parse	Task: Extract Knowledge Graph information from the text content.
	Extraction Requirements:
	• Entities: All entities related to [Root Entity].
	• Relationships: The relationships between these entities.
	• Descriptions: Brief descriptions for each entity and relationship.
	
	Key Constraints:
	• Must include [Root Entity] in the ”entities” list.
	• Extract at least 3-5 meaningful relationships.
	
	Output Format: Strict JSON structure:
	{
	”entities”: [{”name”: ”...”, ”type”: ”...”, ”description”: ”...”}],
	”relationships”: [{”source”: ”...”, ”target”: ”...”, ”relation_type”: ”...”, ”description”: ”...”}]
	}

Table 8. Prompts for Phase II QA Generation

Level	Content
Level 1	Task: Convert a simple VQA regarding a Root Entity into a complex, multi-hop question based on a Knowledge Graph. Inputs: Root Entity, Simple VQA (visual anchor), Knowledge Graph. Core Requirements: • Visual Anchor: The question must start with a visual description of the Root Entity (e.g., "the white phone"). Never reveal the entity's name. • Forced Multi-Hop Reasoning: Requires at least 3 hops. (Hop 1: Visual $\rightarrow$ Entity; Hop 2+: Entity $\rightarrow$ Answer via KG). • Information Concealment: Never mention intermediate entities or relations. No shortcuts allowed; the user must identify the Root Entity first. • Answer Criteria: Simple, concise, factual, unique, and time-invariant. • Relation Uniqueness: Selected relations must not be one-to-many (e.g., avoid "famous buildings" of a city). Output Format: <reason> (Path planning and checks), <question>, <answer>.
Level 2	Task: Generate a high-difficulty, multi-entity, multi-hop VQA question based on multiple Anchor Entities. Inputs: Multiple sets of Anchor Entities, Simple VQAs (visual anchors), and Knowledge Graphs. Core Requirements: • Visual Anchor: Use the visual descriptions provided in the Simple VQAs for all anchors. Strictly prohibit revealing real names. • Forced Multi-Entity & Multi-Hop: Combine knowledge from at least 2 Anchor Entities. Total inference steps ≥ 4. (Steps: Visual A $\rightarrow$ Entity A; Visual B $\rightarrow$ Entity B; plus KG hops for both). • Information Concealment: No intermediate names/attributes mentioned. No shortcuts. • Answer Criteria: Simple, unique, factual, and time-invariant. • Constraint Checks: Verify relation uniqueness and ensure no "data leakage" allows answering without visual identification. Output Format: <reason> (Detailed path planning, constraint validation, and anti-leakage self-check), <question>, <answer>.

Table 9. Prompt for LLM-as-Judge

Prompt Type	Content

System Prompt	Task: Compare the "Predicted answer" with the "Gold target" and judge if it is correct. The "Gold target" may have multiple values, separated by [or].
	
	Criteria:
	accept:
	– Core facts match the "Gold target".
	– Tolerance: Ignore spelling, formatting, or case if semantics are unaffected.
	– Tolerance: Allow omission of info implied by the question (e.g., Q: "City", Gold: "San Francisco, CA", Predict: "San Francisco" is accept).
	– Tolerance: Allow redundant information, as long as the core facts from the "Gold target" are present and not contradicted.
	
	wrong:
	– Contains facts that contradict the "Gold target".
	– Lacks key information from the "Gold target".
	– Answer is "I don't know" or a refusal to answer.
	
	Output Format:
	`<reason>Brief reasoning.</reason>`
	`<answer>accept or wrong</answer>`

User Prompt	To Evaluate Question: [User's Original Question] Ground Truth: [Ground Truth Answer] Predicted Answer: [Agent's Final Answer]

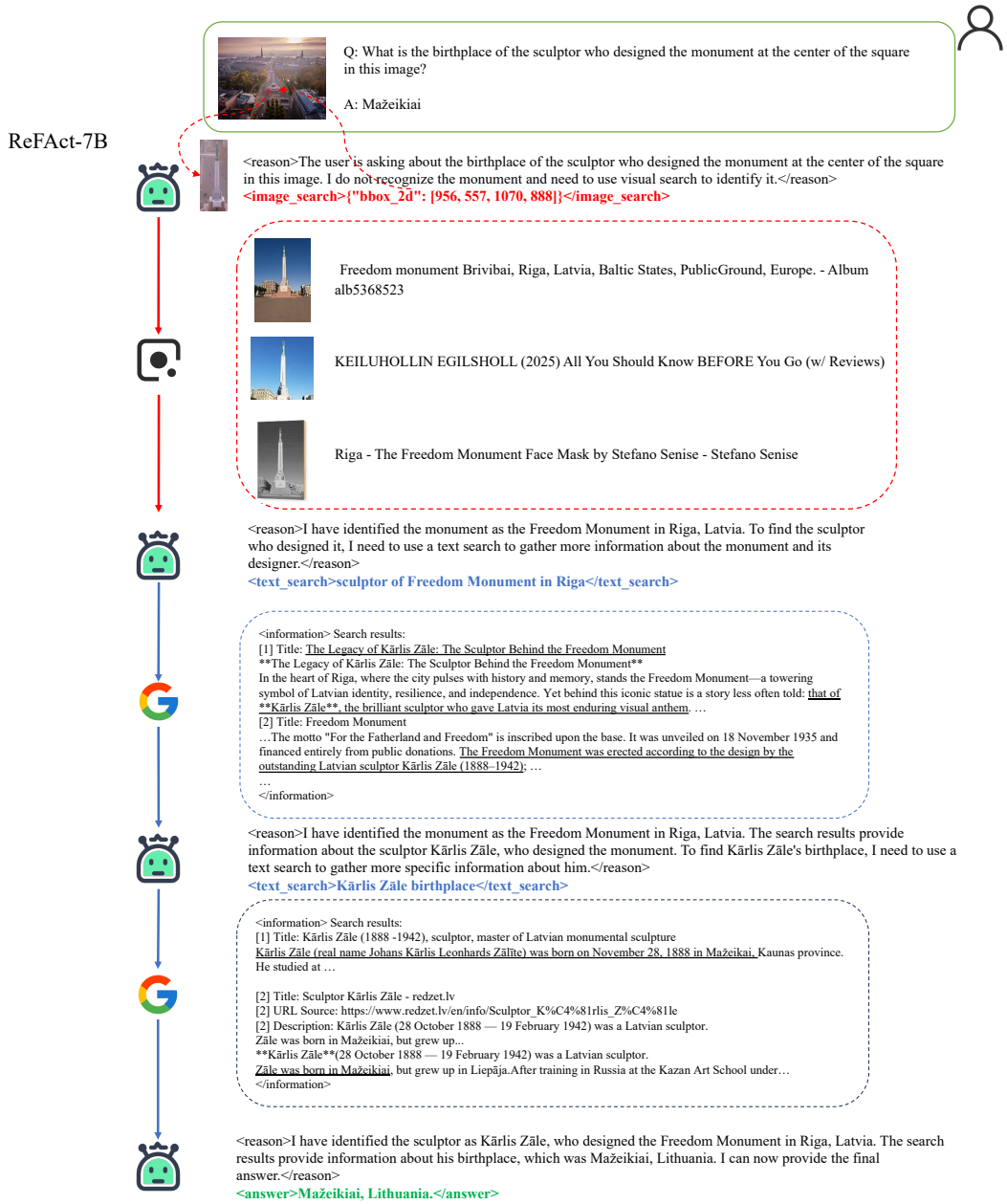

Figure 7. A trajectory flow for Case 1. ReFAct's `Ground` operation allows it to filter out the high-noise background and identify the correct entity, leading to a successful outcome.

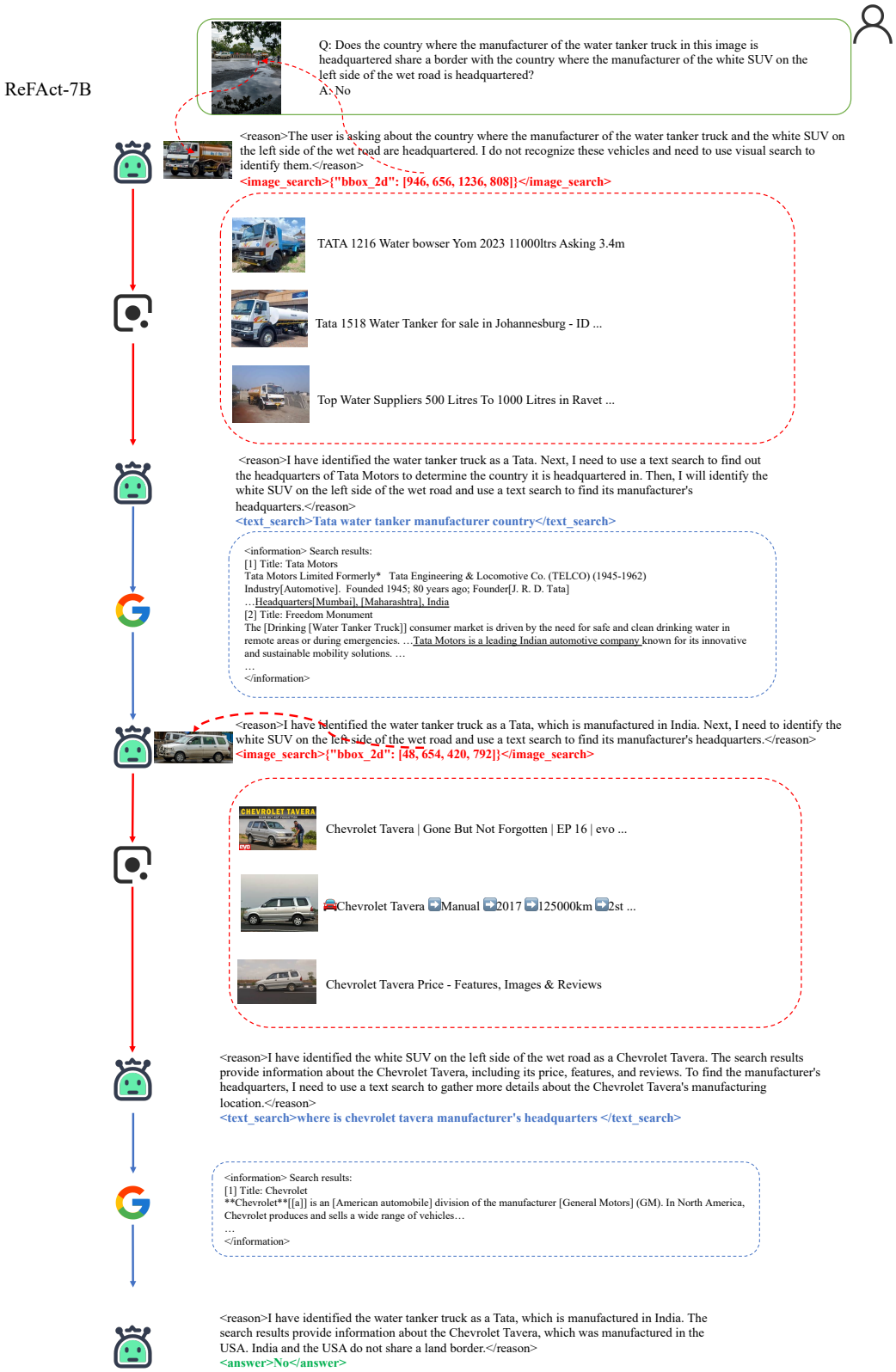

Figure 8. A trajectory flow for Case 2. The agent executes multiple retrieval steps for different entities. The process successfully synthesizes information from multiple retrieval steps to integrate information and correctly answer the question.