VipAct: Visual-Perception Enhancement via Specialized VLM Agent Collaboration and Tool-use

Zhehao Zhang[1*], Ryan A. Rossi[2], Tong Yu[2], Franck Dernoncourt[2], Ruiyi Zhang[2], Jiuxiang Gu[2], Sungchul Kim[2], Xiang Chen[2], Zichao Wang[2], Nedim Lipka[2]

[1]The Ohio State University
[2]Adobe Inc.
zhang.16420@osu.edu,
{ryrossi, tyu, dernonco, ruizhang, jigu, sukim, xiangchen, jackwa, lipka}@adobe.com

Abstract

While vision-language models (VLMs) have demonstrated remarkable performance across various tasks combining textual and visual information, they continue to struggle with fine-grained visual perception tasks that require detailed pixel-level analysis. Effectively eliciting comprehensive reasoning from VLMs on such intricate visual elements remains an open challenge. In this paper, we present VIPACT, an agent framework that enhances VLMs by integrating multi-agent collaboration and vision expert models, enabling more precise visual understanding and comprehensive reasoning. VIPACT consists of an orchestrator agent, which manages task requirement analysis, planning, and coordination, along with specialized agents that handle specific tasks such as image captioning and vision expert models that provide high-precision perceptual information. This multi-agent approach allows VLMs to better perform fine-grained visual perception tasks by synergizing planning, reasoning, and tool use. We evaluate VIPACT on benchmarks featuring a diverse set of visual perception tasks, with experimental results demonstrating significant performance improvements over state-of-the-art baselines across all tasks. Furthermore, comprehensive ablation studies reveal the critical role of multi-agent collaboration in eliciting more detailed System-2 reasoning and highlight the importance of image input for task planning. Additionally, our error analysis identifies patterns of VLMs' inherent limitations in visual perception, providing insights into potential future improvements. VIPACT offers a flexible and extensible framework, paving the way for more advanced visual perception systems across various real-world applications.

1 Introduction

Recent advances in large multimodal models (LMMs), particularly vision-language models (VLMs) (OpenAI 2024; Bai et al. 2023; Chen et al. 2024b), have shown impressive performance in integrating textual and visual information. Models like GPT-4o (OpenAI 2024) achieve strong results on various image-text benchmarks (Yue et al. 2024) and hold promise for real-world applications such as web navigation (Zheng et al. 2024a; He et al. 2024a). However, despite these advancements, studies (Rahmanzadehgervi et al. 2024; Fu et al. 2024; Tong et al. 2024; Li et al. 2024c) show that SOTA VLMs still struggle with fine-grained visual perception tasks, such as detecting line intersections or object boundaries—tasks that are trivial for humans. Overcoming these challenges is essential for deploying VLMs in critical applications like surgical robotics and autonomous driving, which demand precise visual understanding.

To address these challenges, prior works have explored visual programming methods (Subramanian et al. 2023; Hu et al. 2024b; Gupta and Kembhavi 2023; Surís, Menon, and Vondrick 2023; Mialon et al. 2023; Wu et al. 2023a; Yang et al. 2023b), where text queries are input into LLMs to generate code that invokes vision-specific models, using their outputs directly as predictions. While effective for predefined tasks, these methods lack generalizability beyond existing toolsets, limiting their use as universal visual perception solutions. Another line of research focuses on prompting strategies to elicit foundation models' System-2 reasoning by involving iterative reasoning with intermediate tokens (Saha et al. 2024). Textual prompting methods (Wei et al. 2022; Saha et al. 2023; Yao et al. 2024; Besta et al. 2024) elicit LLMs to generate structured reasoning steps for complex text-based tasks, but their efficacy on fine-grained visual perception is underexplored. Similarly, visual prompting techniques (Yang et al. 2023a; Wu et al. 2024), which add artifacts like bounding boxes or masks to images, guide VLMs in interpreting visual data. While promising for some compositional visual reasoning, it is still unclear whether VLMs can accurately perceive such visual prompts, let alone whether these methods improve performance in visual perception.

To fill this gap, and inspired by advances in LLM-based agents (Liu et al. 2023b; Wang et al. 2024; Shen et al. 2024), we propose VIPACT (**VI**sual-**P**erception via VLM **A**gent **C**ollaboration and **T**ool-use), a VLM-based framework that integrates multi-agent collaboration and vision expert models for fine-grained visual perception tasks. As shown in Figure 1, VIPACT consists of three core components: (1) an **orchestrator agent** that manages the workflow by analyzing tasks, coordinating agents, selecting tools, summarizing evidence, and deducing final answers; (2) **specialized agents** for tasks such as image captioning, visual prompt description, and image comparison, providing detailed visual analysis

[*]Work done during Zhehao's internship at Adobe.

to the orchestrator; and (3) **vision expert models**, offering task-specific, fine-grained perceptual information to address VLMs' limitations. We evaluate VIPACT against SOTA baselines across benchmarks that include diverse visual perception tasks featuring complex elements like visual prompts and multi-image inputs. VIPACT consistently outperforms previous baselines on all tasks with different VLMs. Besides, our in-depth analysis highlights the importance of multi-agent collaboration in eliciting more detailed System-2 reasoning, as well as the critical role of visual input for task planning, with improved error handling and evidence aggregation.

Our key contributions are as follows: (1) VIPACT, a multimodal agent framework that synergizes multi-agent collaboration with vision expert models to enhance fine-grained visual perception. It is an autonomous system capable of handling diverse visual perception tasks using a single prompt template. It leverages a VLM for task analysis, planning, and invoking multi-agent collaboration, with flexible plug-and-play modular components. (2) We conduct experiments across diverse visual perception benchmarks, demonstrating VIPACT's advantages over SOTA baselines; (3) We systematically analyze previous methods that proved to be effective in improving the general capabilities of foundation models for fine-grained visual perception, revealing their inconsistent effectiveness. (4) We present comprehensive ablation studies to assess the impact of multi-agent collaboration, visual input for planning, and each component of VIPACT, along with a detailed error analysis identifying the limitations of current VLMs, which serve as bottlenecks for further improvement.

2 Related Work

VLM-based Agent. Advancements in LLM capabilities like planning have spurred the development of LLM-based agents across diverse applications (Zhang et al. 2023; Xi et al. 2023; Chen et al. 2023a; Significant-Gravitas 2024; Shen et al. 2024; Deng et al. 2024a; Zhang, Gao, and Lou 2024; Xie et al. 2024b; Liu et al. 2023b,a; Zhang, Xu, and Deng 2023; Zhou et al. 2023). The introduction of visually capable models has positioned them as backbones for vision-centric agents (Hu et al. 2024a). Current research largely focuses on Web/GUI agents for interface interaction (Yan et al. 2023; Yang et al. 2023c; Zheng et al. 2024a; Kapoor et al. 2024; Koh et al. 2024; Lù, Kasner, and Reddy 2024; Deng et al. 2024b; You et al. 2024; Zheng et al. 2024b; He et al. 2024b) and embodied agents controlling robots (Nasiriany et al. 2024; Tan et al. 2024; Xie et al. 2024a; Yang et al. 2024b). However, VLM-based agents specifically for natural image perception tasks remain unexplored.

Visual Programming. Recent LLMs excel at code generation (Gao et al. 2023; Zhang et al. 2023; Zhang, Gao, and Lou 2024; Zhang, Chen, and Yang 2024; Schick et al. 2024), enabling them to solve reasoning tasks via tool use, especially in areas like mathematical reasoning (Cobbe et al. 2021; Hendrycks et al. 2021). This paradigm has been extended to vision tasks (Subramanian et al. 2023; Hu et al. 2024b; Gupta and Kembhavi 2023; Surís, Menon, and Vondrick 2023; Mialon et al. 2023; Wu et al. 2023a; Koo et al. 2024). Systems like MM-REACT (Yang et al. 2023b) integrate LLMs with vision experts following ReAct's (Yao et al. 2023) prompt template, while ViperGPT (Surís, Menon, and Vondrick 2023) and VisProg (Gupta and Kembhavi 2023) use LLMs to generate executable code for visual reasoning without extra training. However, these often depend solely on text queries for code generation and employ rigid tool selection, hindering adaptation to new tasks. This limits their application to simpler visual QA scenarios (Hudson and Manning 2019; Suhr et al. 2019; Marino et al. 2019), lacking support for fine-grained perception, visual prompts, or multi-image inputs, thereby restricting their utility in more complex visual reasoning. Table 1 provides a detailed comparison.

3 VipAct Framework

Our proposal, **VIPACT**, is illustrated in Figure 1. It consists of three main components: (1) **orchestrator agent** (Section 3.1), which controls the entire workflow by analyzing task requirements and task plans, initiating collaboration with other agents, selecting appropriate vision expert models, summarizing evidence from other agents or tools, and deducing the final answer. (2) **specialized agents** (Section 3.2), designed to handle specific tasks such as image captioning, visual prompt description, and image comparison. These agents provide detailed information to the orchestrator agent. (3) **vision expert models** (Section 3.3), which include specialized task-specific vision models that provide accurate, fine-grained perceptual information, addressing limitations of current VLMs. Intuitively, VIPACT enhances the VLM's System-2 reasoning by generating detailed intermediate reasoning steps through multi-agent collaboration while leveraging the high-precision perceptual information from vision expert models.

3.1 Orchestrator Agent

Task Requirement Analysis and Planning: Inspired by recent works (Yao et al. 2022; Huang et al. 2022; Yang et al. 2023b; Sun et al. 2024) that integrate reasoning, planning, and action in LLM-based agent frameworks, the orchestrator agent begins by analyzing the task requirements derived from the images and queries. This analysis identifies the key elements necessary to solve the problem and the corresponding critical visual features that must be acquired in subsequent steps of the agent's workflow, as well as other criteria derived from its own knowledge. The orchestrator agent then generates a detailed plan for tackling the task, outlining the concrete steps required to obtain the necessary information to meet the objectives. For instance, in a depth estimation task as illustrated in Figure 1, the orchestrator agent would determine the essential requirements for comparing depth, such as identifying the specific objects targeted by the red circles and recognizing their relative positions to the camera.

Tool Selection and Incorporation of Specialized Agents: After analyzing the task requirements and formulating a plan, the orchestrator agent selects the appropriate tools and specialized agents to provide the visual information necessary to solve the task. Depending on the nature of the task, this may involve initiating collaboration with specialized agents or external vision expert models to gather fine-grained in-

Methods	Reas.	Tool	Multi-Ag.	Plan Img	Exec Img	Img Loop	Multi-Img	Vis. Prompt
ReAct (Yao et al. 2023)	✓	✓	✗	✗	✗	✗	✗	✗
MM-ReAct (Yang et al. 2023b)	✓	✓	✗	✗	✓	✗	✗	✗
ViperGPT (Surís, Menon, and Vondrick 2023)	✗	✓	✗	✗	✓	✗	✗	✗
VisProg (Gupta and Kembhavi 2023)	✗	✓	✗	✗	✓	✗	✗	✗
CodeVQA (Subramanian et al. 2023)	✗	✓	✗	✗	✓	✗	✗	✗
VIPACT (Ours)	✓	✓	✓	✓	✓	✓	✓	✓

Table 1: Comparison of VIPACT with other agentic frameworks. ✓ indicates the presence of a specific feature, ✗ its absence. Column abbreviations: "Reas." for modules to elicit reasoning process, "Tool." for tool integration, "Multi-Ag." for multi-agent support, "Plan Img" for image input in planning, "Exec Img" for image input in execution, "Img Loop" for image use in iterative loops, "Multi-Img" for multi-image support, and "Vis. Prompt" for specific design for images containing visual prompts.

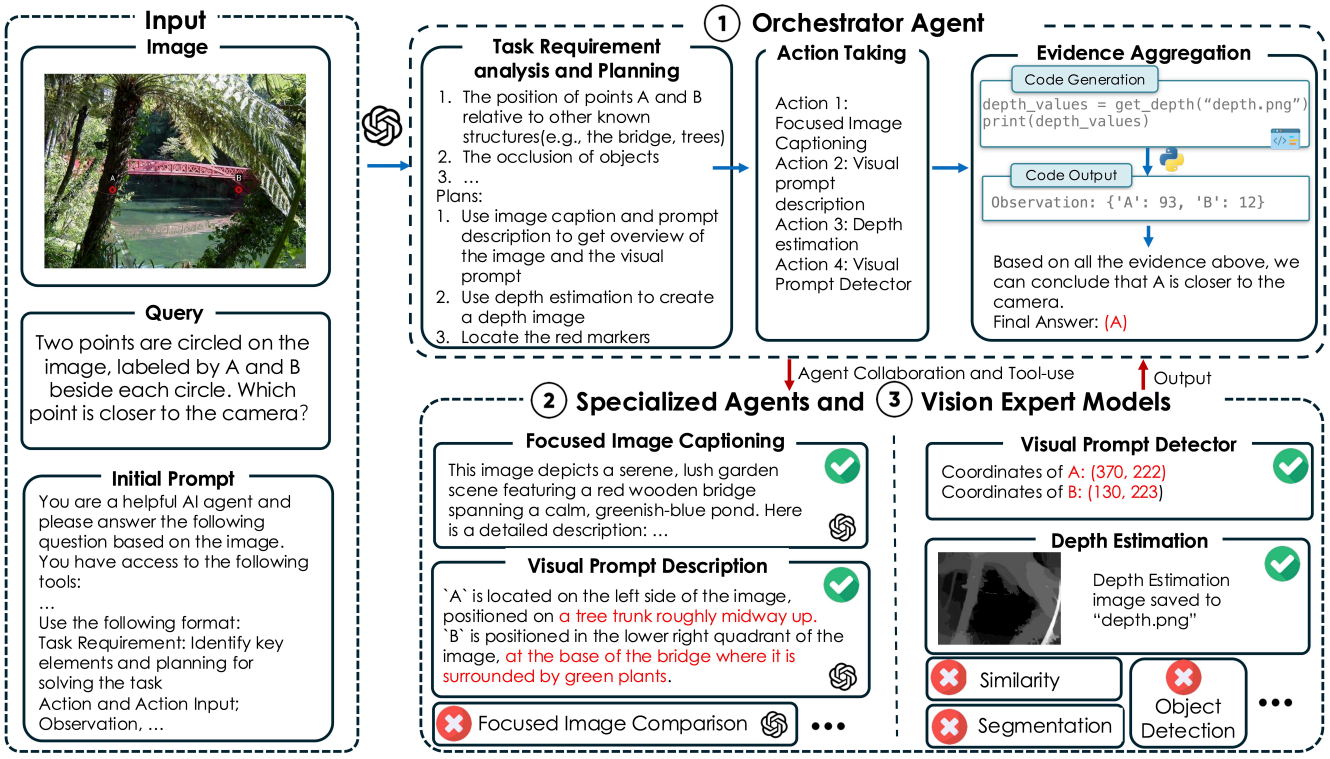

Figure 1: The VIPACT framework for visual perception. It consists of (1) an orchestrator agent for task analysis and coordination, (2) specialized agents for specialized visual analysis, and (3) vision expert models for providing pixel-level visual information. Note that not all agents and expert models are invoked in every instance. For complete task-solving processes of VIPACT, refer to the case studies in Appendix D.

formation. Details on these specialized agents and external vision expert models are provided in Sections 3.2 and 3.3.

Evidence Summarization: Once the tools and specialized agents have performed their respective tasks in separate environments, the orchestrator agent compiles and summarizes the collected evidence. This involves integrating the outputs from various tools and agents, ensuring that all relevant information is coherently synthesized to support the decision-making process. The orchestrator agent also resolves conflicting evidence and double-checks the factuality of the information, as errors or hallucinations may arise from the expert models and specialized agents.

Final Answer Deduction: With the summarized evidence, the orchestrator agent deduces the final answer. It applies reasoning based on the accumulated information to arrive at an unambiguous conclusion. Depending on the nature and format of the gathered data, the orchestrator agent may generate `Python` code, which is then executed by an external `Python` interpreter to derive the final answer. If the gathered information does not lead to a perfect answer, the orchestrator agent is designed to select the closest possible option based on the evidence, supplemented by its own understanding.

3.2 Collaboration with Specialized Agents

VIPACT incorporates three specialized agents to enhance its visual perception capabilities: focused image captioning,

visual prompt description, and focused image comparison. These agents provide task-specific, detailed information to the orchestrator agent through function calling in a separate environment, integrating their outputs into the main reasoning process. The three specialized agents are described below.

Focused Image Captioning: This agent generates detailed image descriptions, optionally emphasizing specific elements relevant to the task by specifying a `focus` argument. The `focus` argument allows for targeted analysis, ranging from general descriptions to particular aspects like "a red car and the background buildings." This flexibility enables the orchestrator agent to obtain precise, task-relevant information from images. Empirical evidence demonstrates its effectiveness across various tasks, with the focus parameter providing fine-grained control over the generated descriptions.

Visual Prompt Description: Specializing in analyzing visual prompts within images (e.g., colored circles, bounding boxes, arrows, textual labels), this agent is crucial for interpreting visual annotations. It generates detailed descriptions of these elements, including their locations, characteristics, and most importantly, the regions or objects these visual prompts target. This enables the orchestrator agent to accurately interpret highlighted or annotated image sections. The agent has shown particular efficacy in tasks involving images with visual prompts, significantly enhancing the system's ability to understand and reason about annotated visual data.

Focused Image Comparison: This agent analyzes multiple images, identifying similarities and differences with an optional focus on specific elements. Similarly, the `focus` parameter allows for targeted comparative analysis, either generally or on specific features as directed by the orchestrator agent. This function can provide a detailed comparison of orientations of objects which can be useful in tasks such as multi-view reasoning. This capability is valuable for tasks requiring multi-image input, such as change detection or pattern identification across images. Empirical results demonstrate this agent's exceptional effectiveness in tasks involving multiple image inputs, with the focus parameter enabling precise comparative analyses.

The prompts for these agents are in Appendix J. VIPACT decomposes complex visual tasks into sub-tasks handled by specialized agents, with an orchestrator agent integrating their outputs. The architecture is extensible, allowing for the addition of new agents to address emerging tasks.

3.3 Integration of Vision-Expert Models

VIPACT further enhances its visual perception capabilities by integrating a suite of vision-expert models, each specializing in specific aspects of image analysis. They collaborate with the orchestrator agent through function calling, uniquely returning both textual data and processed images—making VIPACT among the earliest agent frameworks that incorporate **visual information directly into the reasoning workflow**. These vision-expert models provide fine-grained visual perception information that is often lacking in current VLM's pre-training data (Zhang et al. 2024a). The vision expert tools used in our experiments are described below:

Visual Prompt Detector: Identifies and localizes annotated elements in images, such as circles, bounding boxes, or other highlighted regions. This tool is crucial for understanding visual instructions or annotations, enabling the agent to focus on relevant areas for analysis. It returns the coordinates of these visual prompts, which often serve as intermediate information to achieve the final answer.

Depth Estimator: Analyzes spatial relationships within scenes, providing crucial information about the relative distances of objects from the camera. This tool enhances the agent's understanding of 3D structure in 2D images, vital for spatial reasoning tasks. It returns a grey-scale depth image that can be directly input into the orchestrator agent, allowing it to interpret depth information or combine it with other evidence to reach the final answer.

Object Detection: Identifies and localizes objects within an image, providing the agent with a comprehensive inventory of visible objects, their locations, and sizes. This facilitates detailed scene understanding and object-centric reasoning. The tool returns both a processed image with detected objects' bounding boxes and textual information about these bounding boxes and objects.

Image Segmentation: Offers precise delineation of image regions, separating objects, backgrounds, and distinct areas. This enables fine-grained analysis of image components, crucial for tasks requiring detailed understanding of object boundaries and spatial relationships. It returns images with segmentation masks along with textual information.

Embedding-based Similarity Computation: Quantifies visual similarities across images by generating compact representations of visual content. This allows for nuanced comparisons and similarity assessments, particularly useful for tasks involving image retrieval or comparative analysis. It returns similarity scores based on the selected embedding model and specified similarity metrics, such as cosine similarity.

The complete function heads, including inputs, outputs, and descriptions for these vision expert models, are provided in the initial prompt for the orchestrator agents in Appendix J. This diverse toolkit empowers the orchestrator agent to select the most appropriate tools for each task dynamically, significantly enhancing the framework's ability to comprehend and reason about complex visual scenarios. The integration of processed images alongside textual outputs in the agent's workflow enables more nuanced and contextually rich visual reasoning. We provide an overview of the VipAct framework in Algorithm 1 with detailed explanations in Appendix G.

4 Experiment

Setup. We use various SOTA closed-source VLMs, including **GPT-4o** (OpenAI 2024), **Gemini-1.5-Pro** (Team et al. 2024), and **Claude-3.5-Sonnet** (Anthropic 2024), as well as open-source ones, such as **LLaVA-OneVision-7B** (Li et al. 2024a), **InternVL-2-Pro** (Chen et al. 2023d, 2024a), and **Llama-3.2-90b-Vision** (Dubey et al. 2024). Following prior works (Zheng et al. 2024a; He et al. 2024a; Liu et al. 2024; Gu et al. 2024), we focus on GPT-4o (OpenAI 2024) as the primary VLM in the main paper. Discussions on others are included in Appendix C, with implementation details in Appendix A.

Datasets. To evaluate VLMs on visual perception tasks, we use the following two challenging datasets designed to

Algorithm 1: VIPACT: VIsual-Perception via VLM Agent Collaboration & Tool-use

Require: Set of visual inputs $\mathcal{V}$, a query q, a vision-language model $\mathcal{M}$, a set of tools $\mathcal{T} = \{T_1, \ldots, T_n\}$ including specialized agents and vision expert models, and the maximum iterations K
Ensure: An answer a to the visual perception task

Initialize orchestrator agent $\mathcal{O}$ with $\mathcal{M}$ and $\mathcal{T}$
$\mathcal{P}_0 \leftarrow \text{FORMATPROMPT}(\mathcal{V}, q)$ ▷ Format initial prompt with visual inputs and query
$t \leftarrow 0, \quad \mathcal{S} \leftarrow \emptyset$ ▷ Initialize iteration counter and state
while $t < K$ and not $\text{ISTERMINATED}(\mathcal{S})$ **do**
  if $\exists T_i \in \mathcal{T} : \text{ISREQUIRED}(T_i, \mathcal{S})$ **then** ▷ Check if any tool is required
    $T^* \leftarrow \arg\max_{T_i \in \mathcal{T}} \text{UTILITY}(T_i, \mathcal{S})$ ▷ Select most useful tool
    $\mathcal{O}_t \leftarrow \text{EXECUTE}(T^*, \mathcal{S})$ ▷ Execute selected tool with the current state as input
    if $\text{CONTAINSVISUALDATA}(\mathcal{O}_t)$ **then**
      $\mathcal{V} \leftarrow \mathcal{V} \cup \text{PROCESSVISUALDATA}(\mathcal{O}_t)$ ▷ Add new visual data if needed
    end if
  else
    $\mathcal{R}_t \leftarrow \mathcal{M}(\mathcal{P}_{t-1})$ ▷ Generate VLM output
    $\mathcal{O}_t \leftarrow \text{INTERPRETOUTPUT}(\mathcal{R}_t)$ ▷ Interpret VLM output
  end if
  $\mathcal{P}_t \leftarrow \text{UPDATEPROMPT}(\mathcal{P}_{t-1}, \mathcal{O}_t)$ ▷ Update prompt with new information
  $\mathcal{S} \leftarrow \text{UPDATESTATE}(\mathcal{S}, \mathcal{O}_t); t \leftarrow t + 1$ ▷ Update state with new observations
end while
$a \leftarrow \text{EXTRACTANSWER}(\mathcal{S})$ ▷ Extract final answer from state
return a

test fine-grained visual perception: (1) **Blink** (Fu et al. 2024) includes diverse visual tasks solvable by humans "within a blink," yet difficult for SOTA VLMs. It features visual prompts such as bounding boxes and interleaved image-text formats, often with multiple images in a single query. We use Blink as the main benchmark. (2) **MMVP** (Tong et al. 2024) is a benchmark for evaluating visual grounding in VLMs, using image pairs from "CLIP-blind pairs"—visually distinct images that are similar in CLIP embedding space. It focuses on nine basic visual patterns that are easy for humans but challenging for SOTA VLMs. Details are provided in Appendix B.

Baselines. We evaluate VIPACT against four types of baselines: (1) **Text-based prompting**, including zero-shot prompting; chain-of-thought (CoT) prompting (Wei et al. 2022; Kojima et al. 2022); Least-to-most prompting (LtM) (Zhou et al. 2022); and Tree-of-thought (ToT) prompting (Yao et al. 2024). (2) **Few-shot in-context learning** (Brown 2020), where in-context exemplars are selected using different strategies, including random selection, or selection based on embedding (Radford et al. 2021; Dosovitskiy et al. 2020) similarity (analyzed separately in Appendix E). (3) **Visual Prompting**, exemplified by Set-of-Mark (SoM) (Yang et al. 2023a), which overlays marks on semantically meaningful image regions. (4) **Vision language agentic frameworks**, including MM-ReAct (Yang et al. 2023b), which integrates LLMs with vision experts via ReAct-style prompts (Yao et al. 2022); ViperGPT (Surís, Menon, and Vondrick 2023), using LLMs to generate code composing vision and language models; and VisProg (Gupta and Kembhavi 2023), which generates visual programs from textual instructions.

Result Analysis. Tables 2 and 3 present the performance of our proposed VIPACT framework and baseline methods on each sub-task of the Blink and MMVP datasets respectively. We make the following key observations: **(1) Text-based prompting methods do not consistently improve performance over zero-shot prompting.** Specifically, as shown in Tables 2 and 3, prior text-based prompting methods effective for LLMs — such as CoT — can improve performance on some sub-tasks like visual similarity, object localization, counting, and spatial relations. However, for other tasks, the improvement is minimal or even negative. More advanced techniques like LtM and ToT exhibit similar phenomena. Empirically, while these methods elicit detailed reasoning, such steps are often ungrounded in visual elements and can cause severe hallucinations. Therefore, it is non-trivial to elicit VLMs' reasoning for better general visual perception using text-based methods from text-only LLMs. **(2) SoM can impair VLMs' fine-grained perception in most scenarios.** From results on both datasets, SoM adversely affects VLM performance on almost all tasks. Empirically, overlaying labeled masks can become cluttered with numerous semantic objects or fine-grained parts, negatively influencing VLM perception of original objects and potentially confusing models with original visual prompts and labels. Consequently, SoM's effectiveness in some compositional reasoning tasks with limited semantic objects does not generalize well to broader visual perception tasks, especially those requiring visual prompt understanding. **(3) Previous visual programming methods exhibit poor generalization ability.** As shown, these methods perform adequately only on limited tasks (e.g., spatial relations, counting) similar to those in common VQA datasets (Hudson and Manning 2019; Suhr et al. 2019; Marino et al. 2019). Their generated code calls a limited set of predefined tools, lacking logic for unsupported scenarios or errors. They cannot sup-

Method	Sim	Count	Depth	Jig	Fun.C	Sem.C	Spat	Local	Vis.C	Multi-v	Average
Text-based Prompting w/ GPT-4o											
Zero-shot	65.44	50.83	64.52	60.00	57.69	56.83	79.92	56.00	86.05	60.15	63.74
CoT	63.70	65.00	73.39	62.00	57.69	57.55	82.52	60.66	82.56	53.38	65.85
LtM	62.22	64.17	70.97	62.67	55.38	55.40	76.22	59.02	83.14	45.86	63.51
ToT	64.44	58.33	71.70	64.00	57.69	59.71	83.22	61.48	78.49	50.38	64.94
Visual Prompting w/ GPT-4o											
SoM	63.70	43.33	68.55	49.33	47.69	52.52	76.22	59.84	83.72	56.40	60.13
Multi-modal Agent Framework w/ GPT-4o											
MM-ReAcT	-	30.00	0.81	-	-	-	63.64	0.00	-	-	-
ViperGPT	-	29.17	0.00	-	-	-	48.95	18.85	-	-	-
VisProg	-	3.33	0.00	-	-	-	31.47	14.75	-	-	-
VIPACT (Ours)	**81.48**	**70.00**	**90.80**	**68.00**	**61.50**	**60.40**	**86.70**	**63.11**	**91.28**	**62.63**	**73.79**

Table 2: Results for visual reasoning tasks in Blink using GPT-4o. Note that "–" indicates methods that do not support multiple images. Our VIPACT consistently outperforms baselines on all tasks.

port images with visual prompts, failing to locate them for subsequent operations (e.g., near-zero performance in depth estimation due to inability to locate red circles, leading to non-executable code). Moreover, code generated solely from text queries lacks flexibility for different image characteristics. These observations highlight the need for a generalizable agent framework leveraging both vision expert models and VLM flexibility. **(4) VIPACT consistently achieves the best performance across all sub-tasks in Blink and MMVP, demonstrating its effectiveness and generalization ability.** By examining VIPACT's reasoning traces, we observe that, compared to text-based and visual prompting methods, VIPACT effectively invokes specialized agents or vision expert models to enhance image understanding. It does not solely rely on their outputs, as evidence might be incorrect or errors may occur. Instead, it aggregates useful evidence with additional reasoning to infer the final answer, showcasing its ability to handle uncertainties and integrate multiple information sources. Figure 3 and 4 in Appendix D show complete reasoning traces of VIPACT.

Method	Accuracy (%)
Zero-shot	68.0
CoT	61.0
LtM	66.0
ToT	66.0
SoM	62.0
MM-ReAct	6.67
ViperGPT	53.0
VisPro	39.0
VIPACT (Ours)	**70.7**

Table 3: Different methods using GPT-4o on MMVP.

5 Ablation Study

To evaluate the effectiveness of various components in VIPACT, we conduct a series of ablation studies. These involve removing or modifying key components of VIPACT to assess their impact on performance across different tasks. The ablation studies are as follows: **(1) Removal of multi-agent collaboration**: We removed the specialized agents and incorporated their prompts as instructions directly into the orchestrator agent to evaluate the importance of multi-agent collaboration. **(2) Removal of image input for orchestrator agent:** We modified the input to the orchestrator agent to only include image paths as text, rather than the actual images which means the image is not visible to the orchestrator agent but still can be served as input for other specialized agents or vision expert models. This setup follows the paradigm used in previous works (Surís, Menon, and Vondrick 2023; Gupta and Kembhavi 2023) and tests the effectiveness of direct visual input to the orchestrator agent. **(3) Removal of specialized agents:** We removed all specialized agents to assess their impact on the VIPACT's performance. **(4) Removal of vision expert models:** We eliminated all vision-expert models to evaluate their contribution.

The results of ablation studies are presented in Table 4 and 5. From these results, we derive the following key insights:

- **Multi-agent collaboration enhances detailed reasoning**: The removal of multi-agent collaboration led to a consistent performance decline. By comparing reasoning steps, we observed that multi-agent collaboration enabled significantly more detailed image analysis (over 80% more generated tokens), such as thorough image captioning. This aligns with observations in LLMs (Wu et al. 2023b; Hong et al. 2023; Qian et al. 2023; Park et al. 2023; Liu et al. 2023b), where agent collaboration enhances task-solving via comprehensive reasoning from diverse perspectives.
- **Direct image input to the orchestrator agent is essential for flexible task planning and error handling**: As shown in Tables 4 and 5, removing direct image input to the orchestrator significantly degrades performance. Without direct visual access, the orchestrator agent relies solely on textual queries, lacking critical visual information for accurate task planning. This leads to suboptimal decision-making, less precise parameter selection (e.g., the `focus` parameter), and overly generalized task analysis, reducing specificity. For instance, in a multi-view reasoning task, direct image input allows the agent to identify reference objects, enabling it to accurately adjust the

Method	Sim	Count	Depth	Jig	Fun.C	Sem.C	Spat	Local	Vis.C	Multi-v
Variants of VIPACT										
VIPACT (Full)	**81.48**	**70.00**	**90.80**	**68.00**	**61.50**	**60.40**	**86.70**	63.11	**91.28**	**62.63**
w/o Multi-agent	80.00	67.50	75.00	66.00	58.46	59.71	82.52	**63.93**	85.47	48.87
w/o Visual Input	77.78	59.71	69.35	61.33	53.85	51.08	83.22	60.66	78.49	48.12
w/o Spec. Agents	65.72	62.45	85.62	62.32	55.25	56.32	81.96	58.49	75.48	46.75
w/o Vision Expert	64.34	57.44	72.58	65.67	59.42	58.59	81.37	57.44	83.63	56.40

Table 4: Ablation study results of VIPACT on the Blink benchmark using GPT-4o. VIPACT (Full) represents the complete framework with all components, while the other variants exclude specific components.

Method	Accuracy (%)
VIPACT	**70.7**
w/o Multi-agent	68.0
w/o Visual Input	54.0
w/o Spec. Agents	67.0
w/o Vision Expert	66.0

Table 5: Ablation of VIPACT on MMVP using GPT-4o.

`focus` parameter and effectively determine the direction of camera movement. Further analysis can be found in Appendix H.

- **Specialized agents and vision expert models significantly contribute to performance**: Specialized agents, though VLMs, intently analyze specific visual information without distractions from other instructions (e.g., format requirements), which can hinder LLM reasoning (Tam et al. 2024). Vision expert models perform pixel-level analyses beyond SOTA VLM capabilities, aiding the orchestrator. As shown in Table 4 and 5, removing them leads to a noticeable performance decline. Overall, our VIPACT framework combines VLM flexibility and planning with vision expert model precision, creating a cohesive system where each component is proven to be essential.

6 Error Analysis

To examine the limitations of GPT-4o's visual perception capabilities and the bottlenecks of our VIPACT, we conduct a comprehensive error analysis. Following prior works (Zhou et al. 2022; Chen et al. 2023b; Zhang, Chen, and Yang 2024), we randomly sampled 20 error cases from each sub-task within the two datasets. The errors were categorized as follows:

- **Failure to perceive small object parts (17%):** The model often overlooks small, semantically important components crucial for precise visual understanding.
- **Difficulty distinguishing closely positioned visual prompts (15%):** The model struggles to differentiate spatially proximate visual prompts.
- **Challenges in fine-grained spatial reasoning (24%):** Tasks requiring high spatial resolution highlight the model's **bias towards foreground objects over backgrounds**; e.g., misinterpreting a highlight meant for the sky near a car as associated with the car.
- **Misinterpretation of relative object positions (14%):** Errors arise when object arrangements differ from real-world expectations, as the model often lacks the ability to infer spatial relations from objects' perspectives, focusing on camera viewpoint.
- **Failure to recognize object orientation (13%):** Difficulty discerning object orientation leads to errors in recognizing object parts, such as distinguishing left/right bicycle pedals.
- **Other errors (17%):** This includes other issues like failure to detect subtle color differences, inaccuracies in multi-image fine-grained structure correspondence, and instances of refusal or instruction misinterpretation.

Case studies illustrating these errors are in Appendix D. Our analysis denotes that while VIPACT shows significant improvements in VLM visual perception, fine-grained perception remains a bottleneck. Specifically, the model lacks the **spatial intelligence or imaginative abilities** (Chen et al. 2018; Huang et al. 2024) necessary to infer relative object positions beyond their pixel positions (from the camera's perspective), especially in the context of real-life scenes. Noticeably, these limitations hinder the model's ability to accurately interpret visual prompts and process tasks involving multiple image inputs. We also examine the significance of multiple image inputs for VLMs in Appendix F.

7 Conclusion

We introduce **VIPACT**, a VLM-based agent framework that synergizes multi-agent collaboration and vision expert models for fine-grained visual perception tasks. By combining the planning and function-calling capabilities of SOTA VLMs, VIPACT enhances VLMs' System-2 reasoning through multi-agent interactions and integrates high-precision, pixel-level information from specialized vision models. Our experiments across a diverse range of visual perception tasks demonstrate that VIPACT achieves SOTA performance, outperforming previous baselines. The comprehensive ablation study highlights the critical role of multi-agent collaboration in eliciting detailed information for reasoning, as well as the importance of image input in task planning. Furthermore, our error analysis highlights several inherent limitations in current SOTA VLMs that form bottlenecks in our framework, offering valuable insights for future improvements.

References

Agrawal, S.; Zhou, C.; Lewis, M.; Zettlemoyer, L.; and Ghazvininejad, M. 2023. In-context Examples Selection for Machine Translation. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Findings of the Association for Computational Linguistics: ACL 2023*, 8857–8873. Toronto, Canada: Association for Computational Linguistics.

Alayrac, J.-B.; Donahue, J.; Luc, P.; Miech, A.; Barr, I.; Hasson, Y.; Lenc, K.; Mensch, A.; Millican, K.; Reynolds, M.; et al. 2022. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35: 23716–23736.

Alexey, D. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv: 2010.11929*.

Anthropic. 2024. Claude 3.5 Sonnet.

Awadalla, A.; Gao, I.; Gardner, J.; Hessel, J.; Hanafy, Y.; Zhu, W.; Marathe, K.; Bitton, Y.; Gadre, S.; Sagawa, S.; et al. 2023. Openflamingo: An open-source framework for training large autoregressive vision-language models. *arXiv preprint arXiv:2308.01390*.

Bai, J.; Bai, S.; Yang, S.; Wang, S.; Tan, S.; Wang, P.; Lin, J.; Zhou, C.; and Zhou, J. 2023. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*.

Bansal, A.; Zhang, Y.; and Chellappa, R. 2020. Visual question answering on image sets. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16*, 51–67. Springer.

Berg-Kirkpatrick, T.; Burkett, D.; and Klein, D. 2012. An Empirical Investigation of Statistical Significance in NLP. In Tsujii, J.; Henderson, J.; and Paşca, M., eds., *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 995–1005. Jeju Island, Korea: Association for Computational Linguistics.

Besta, M.; Blach, N.; Kubicek, A.; Gerstenberger, R.; Podstawski, M.; Gianinazzi, L.; Gajda, J.; Lehmann, T.; Niewiadomski, H.; Nyczyk, P.; et al. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 17682–17690.

Brown, T. B. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Chen, G.; Dong, S.; Shu, Y.; Zhang, G.; Sesay, J.; Karlsson, B. F.; Fu, J.; and Shi, Y. 2023a. Autoagents: A framework for automatic agent generation. *arXiv preprint arXiv:2309.17288*.

Chen, J.; Pan, X.; Yu, D.; Song, K.; Wang, X.; Yu, D.; and Chen, J. 2023b. Skills-in-context prompting: Unlocking compositionality in large language models. *arXiv preprint arXiv:2308.00304*.

Chen, M.; Tworek, J.; Jun, H.; Yuan, Q.; Pinto, H. P. D. O.; Kaplan, J.; Edwards, H.; Burda, Y.; Joseph, N.; Brockman, G.; et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Chen, S.; Han, Z.; He, B.; Buckley, M.; Torr, P.; Tresp, V.; and Gu, J. 2023c. Understanding and Improving In-Context Learning on Vision-language Models. *arXiv preprint arXiv:2311.18021*, 1(2).

Chen, X.; Li, L.-J.; Fei-Fei, L.; and Gupta, A. 2018. Iterative Visual Reasoning Beyond Convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Chen, Z.; Wang, W.; Tian, H.; Ye, S.; Gao, Z.; Cui, E.; Tong, W.; Hu, K.; Luo, J.; Ma, Z.; et al. 2024a. How Far Are We to GPT-4V? Closing the Gap to Commercial Multimodal Models with Open-Source Suites. *arXiv preprint arXiv:2404.16821*.

Chen, Z.; Wu, J.; Wang, W.; Su, W.; Chen, G.; Xing, S.; Zhong, M.; Zhang, Q.; Zhu, X.; Lu, L.; Li, B.; Luo, P.; Lu, T.; Qiao, Y.; and Dai, J. 2023d. InternVL: Scaling up Vision Foundation Models and Aligning for Generic Visual-Linguistic Tasks. *arXiv preprint arXiv:2312.14238*.

Chen, Z.; Wu, J.; Wang, W.; Su, W.; Chen, G.; Xing, S.; Zhong, M.; Zhang, Q.; Zhu, X.; Lu, L.; et al. 2024b. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 24185–24198.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; Hesse, C.; and Schulman, J. 2021. Training Verifiers to Solve Math Word Problems. *arXiv preprint arXiv:2110.14168*.

Deng, X.; Gu, Y.; Zheng, B.; Chen, S.; Stevens, S.; Wang, B.; Sun, H.; and Su, Y. 2024a. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36.

Deng, Y.; Zhang, X.; Zhang, W.; Yuan, Y.; Ng, S.-K.; and Chua, T.-S. 2024b. On the Multi-turn Instruction Following for Conversational Web Agents. *arXiv preprint arXiv:2402.15057*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Yang, A.; Fan, A.; et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Fu, X.; Hu, Y.; Li, B.; Feng, Y.; Wang, H.; Lin, X.; Roth, D.; Smith, N. A.; Ma, W.-C.; and Krishna, R. 2024. BLINK: Multimodal Large Language Models Can See but Not Perceive. *arXiv preprint arXiv:2404.12390*.

Gao, L.; Madaan, A.; Zhou, S.; Alon, U.; Liu, P.; Yang, Y.; Callan, J.; and Neubig, G. 2023. Pal: Program-aided language models. In *International Conference on Machine Learning*, 10764–10799. PMLR.

Gu, Y.; Zheng, B.; Gou, B.; Zhang, K.; Chang, C.; Srivastava, S.; Xie, Y.; Qi, P.; Sun, H.; and Su, Y. 2024. Is your llm secretly a world model of the internet? model-based planning for web agents. *arXiv preprint arXiv:2411.06559*.

Gupta, T.; and Kembhavi, A. 2023. Visual programming: Compositional visual reasoning without training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14953–14962.

He, H.; Yao, W.; Ma, K.; Yu, W.; Dai, Y.; Zhang, H.; Lan, Z.; and Yu, D. 2024a. WebVoyager: Building an End-to-End Web Agent with Large Multimodal Models. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 6864–6890. Bangkok, Thailand: Association for Computational Linguistics.

He, H.; Yao, W.; Ma, K.; Yu, W.; Dai, Y.; Zhang, H.; Lan, Z.; and Yu, D. 2024b. WebVoyager: Building an End-to-End Web Agent with Large Multimodal Models. *arXiv preprint arXiv:2401.13919*.

Hendrycks, D.; Burns, C.; Kadavath, S.; Arora, A.; Basart, S.; Tang, E.; Song, D.; and Steinhardt, J. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Hong, S.; Zheng, X.; Chen, J.; Cheng, Y.; Wang, J.; Zhang, C.; Wang, Z.; Yau, S. K. S.; Lin, Z.; Zhou, L.; et al. 2023. Metagpt: Meta programming for multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352*.

Hu, Y.; Shi, W.; Fu, X.; Roth, D.; Ostendorf, M.; Zettlemoyer, L.; Smith, N. A.; and Krishna, R. 2024a. Visual Sketchpad: Sketching as a Visual Chain of Thought for Multimodal Language Models. *arXiv preprint arXiv:2406.09403*.

Hu, Y.; Stretcu, O.; Lu, C.-T.; Viswanathan, K.; Hata, K.; Luo, E.; Krishna, R.; and Fuxman, A. 2024b. Visual Program Distillation: Distilling Tools and Programmatic Reasoning into Vision-Language Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9590–9601.

Hu, Y.; Stretcu, O.; Lu, C.-T.; Viswanathan, K.; Hata, K.; Luo, E.; Krishna, R.; and Fuxman, A. 2024c. Visual program distillation: Distilling tools and programmatic reasoning into vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9590–9601.

Huang, W.; Abbeel, P.; Pathak, D.; and Mordatch, I. 2022. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International conference on machine learning*, 9118–9147. PMLR.

Huang, W.; Wang, C.; Li, Y.; Zhang, R.; and Fei-Fei, L. 2024. ReKep: Spatio-Temporal Reasoning of Relational Keypoint Constraints for Robotic Manipulation. *arXiv preprint arXiv:2409.01652*.

Hudson, D. A.; and Manning, C. D. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 6700–6709.

Hussain, M. 2023. YOLO-v1 to YOLO-v8, the rise of YOLO and its complementary nature toward digital manufacturing and industrial defect detection. *Machines*, 11(7): 677.

Jiang, Y.; Irvin, J.; Wang, J. H.; Chaudhry, M. A.; Chen, J. H.; and Ng, A. Y. 2024. Many-Shot In-Context Learning in Multimodal Foundation Models. *arXiv preprint arXiv:2405.09798*.

Kapoor, R.; Butala, Y. P.; Russak, M.; Koh, J. Y.; Kamble, K.; Alshikh, W.; and Salakhutdinov, R. 2024. OmniACT: A Dataset and Benchmark for Enabling Multimodal Generalist Autonomous Agents for Desktop and Web. *arXiv preprint arXiv:2402.17553*.

Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A. C.; Lo, W.-Y.; et al. 2023. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4015–4026.

Koh, J. Y.; Lo, R.; Jang, L.; Duvvur, V.; Lim, M. C.; Huang, P.-Y.; Neubig, G.; Zhou, S.; Salakhutdinov, R.; and Fried, D. 2024. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. *arXiv preprint arXiv:2401.13649*.

Kojima, T.; Gu, S. S.; Reid, M.; Matsuo, Y.; and Iwasawa, Y. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35: 22199–22213.

Koo, J.; Yang, Z.; Cascante-Bonilla, P.; Ray, B.; and Ordonez, V. 2024. PropTest: Automatic Property Testing for Improved Visual Programming. In Al-Onaizan, Y.; Bansal, M.; and Chen, Y.-N., eds., *Findings of the Association for Computational Linguistics: EMNLP 2024*, 8241–8256. Miami, Florida, USA: Association for Computational Linguistics.

Li, B.; Zhang, Y.; Guo, D.; Zhang, R.; Li, F.; Zhang, H.; Zhang, K.; Li, Y.; Liu, Z.; and Li, C. 2024a. LLaVA-OneVision: Easy Visual Task Transfer. *arXiv preprint arXiv:2408.03326*.

Li, F.; Zhang, R.; Zhang, H.; Zhang, Y.; Li, B.; Li, W.; Ma, Z.; and Li, C. 2024b. Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models. *arXiv preprint arXiv:2407.07895*.

Li, X.; Zhang, F.; Diao, H.; Wang, Y.; Wang, X.; and Duan, L.-Y. 2024c. DenseFusion-1M: Merging Vision Experts for Comprehensive Multimodal Perception. *arXiv preprint arXiv:2407.08303*.

Liu, J.; Song, Y.; Lin, B. Y.; Lam, W.; Neubig, G.; Li, Y.; and Yue, X. 2024. VisualWebBench: How Far Have Multimodal LLMs Evolved in Web Page Understanding and Grounding? *arXiv preprint arXiv:2404.05955*.

Liu, X.; Yu, H.; Zhang, H.; Xu, Y.; Lei, X.; Lai, H.; Gu, Y.; Ding, H.; Men, K.; Yang, K.; et al. 2023a. Agentbench: Evaluating llms as agents. *arXiv preprint arXiv:2308.03688*.

Liu, Z.; Zhang, Y.; Li, P.; Liu, Y.; and Yang, D. 2023b. Dynamic llm-agent network: An llm-agent collaboration framework with agent team optimization. *arXiv preprint arXiv:2310.02170*.

Lù, X. H.; Kasner, Z.; and Reddy, S. 2024. Weblinx: Real-world website navigation with multi-turn dialogue. *arXiv preprint arXiv:2402.05930*.

Marino, K.; Rastegari, M.; Farhadi, A.; and Mottaghi, R. 2019. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, 3195–3204.

Mialon, G.; Dessì, R.; Lomeli, M.; Nalmpantis, C.; Pasunuru, R.; Raileanu, R.; Rozière, B.; Schick, T.; Dwivedi-Yu, J.; Celikyilmaz, A.; et al. 2023. Augmented language models: a survey. *arXiv preprint arXiv:2302.07842*.

Nasiriany, S.; Xia, F.; Yu, W.; Xiao, T.; Liang, J.; Dasgupta, I.; Xie, A.; Driess, D.; Wahid, A.; Xu, Z.; et al. 2024. Pivot: Iterative visual prompting elicits actionable knowledge for vlms. *arXiv preprint arXiv:2402.07872*.

Nguyen, T.; and Wong, E. 2023. In-context example selection with influences. *arXiv preprint arXiv:2302.11042*.

OpenAI. 2024. Hello GPT-4o. https://openai.com/index/hello-gpt-4o/. Accessed: 2024-08-22.

Park, J. S.; O'Brien, J.; Cai, C. J.; Morris, M. R.; Liang, P.; and Bernstein, M. S. 2023. Generative Agents: Interactive Simulacra of Human Behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST '23. New York, NY, USA: Association for Computing Machinery. ISBN 9798400701320.

Qian, C.; Cong, X.; Yang, C.; Chen, W.; Su, Y.; Xu, J.; Liu, Z.; and Sun, M. 2023. Communicative agents for software development. *arXiv preprint arXiv:2307.07924*, 6.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PMLR.

Rahmanzadehgervi, P.; Bolton, L.; Taesiri, M. R.; and Nguyen, A. T. 2024. Vision language models are blind. *arXiv preprint arXiv:2407.06581*.

Reid, M.; Savinov, N.; Teplyashin, D.; Lepikhin, D.; Lillicrap, T.; Alayrac, J.-b.; Soricut, R.; Lazaridou, A.; Firat, O.; Schrittwieser, J.; et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Saha, S.; Levy, O.; Celikyilmaz, A.; Bansal, M.; Weston, J.; and Li, X. 2023. Branch-solve-merge improves large language model evaluation and generation. *arXiv preprint arXiv:2310.15123*.

Saha, S.; Prasad, A.; Chen, J. C.-Y.; Hase, P.; Stengel-Eskin, E.; and Bansal, M. 2024. System-1. x: Learning to Balance Fast and Slow Planning with Language Models. *arXiv preprint arXiv:2407.14414*.

Schick, T.; Dwivedi-Yu, J.; Dessì, R.; Raileanu, R.; Lomeli, M.; Hambro, E.; Zettlemoyer, L.; Cancedda, N.; and Scialom, T. 2024. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36.

Shen, Y.; Song, K.; Tan, X.; Li, D.; Lu, W.; and Zhuang, Y. 2024. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *Advances in Neural Information Processing Systems*, 36.

Shi, L.; Ma, W.; and Vosoughi, S. 2024. Judging the Judges: A Systematic Investigation of Position Bias in Pairwise Comparative Assessments by LLMs. *arXiv preprint arXiv:2406.07791*.

Significant-Gravitas. 2024. AutoGPT. GitHub repository.

Subramanian, S.; Narasimhan, M.; Khangaonkar, K.; Yang, K.; Nagrani, A.; Schmid, C.; Zeng, A.; Darrell, T.; and Klein, D. 2023. Modular Visual Question Answering via Code Generation. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 747–761. Toronto, Canada: Association for Computational Linguistics.

Suhr, A.; Zhou, S.; Zhang, A.; Zhang, I.; Bai, H.; and Artzi, Y. 2019. A Corpus for Reasoning about Natural Language Grounded in Photographs. In Korhonen, A.; Traum, D.; and Màrquez, L., eds., *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 6418–6428. Florence, Italy: Association for Computational Linguistics.

Sun, S.; Liu, Y.; Wang, S.; Iter, D.; Zhu, C.; and Iyyer, M. 2024. PEARL: Prompting Large Language Models to Plan and Execute Actions Over Long Documents. In Graham, Y.; and Purver, M., eds., *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, 469–486. St. Julian's, Malta: Association for Computational Linguistics.

Surís, D.; Menon, S.; and Vondrick, C. 2023. ViperGPT: Visual inference via python execution for reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 11888–11898.

Tam, Z. R.; Wu, C.-K.; Tsai, Y.-L.; Lin, C.-Y.; Lee, H.-y.; and Chen, Y.-N. 2024. Let me speak freely? a study on the impact of format restrictions on performance of large language models. *arXiv preprint arXiv:2408.02442*.

Tan, W.; Ding, Z.; Zhang, W.; Li, B.; Zhou, B.; Yue, J.; Xia, H.; Jiang, J.; Zheng, L.; Xu, X.; et al. 2024. Towards general computer control: A multimodal agent for red dead redemption ii as a case study. *arXiv preprint arXiv:2403.03186*.

Team, G.; Georgiev, P.; Lei, V. I.; Burnell, R.; Bai, L.; Gulati, A.; Tanzer, G.; Vincent, D.; Pan, Z.; Wang, S.; et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Tong, S.; Liu, Z.; Zhai, Y.; Ma, Y.; LeCun, Y.; and Xie, S. 2024. Eyes Wide Shut? Exploring the Visual Shortcomings of Multimodal LLMs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9568–9578.

Wang, J.; Wang, J.; Athiwaratkun, B.; Zhang, C.; and Zou, J. 2024. Mixture-of-Agents Enhances Large Language Model Capabilities. *arXiv preprint arXiv:2406.04692*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35: 24824–24837.

Wu, C.; Yin, S.; Qi, W.; Wang, X.; Tang, Z.; and Duan, N. 2023a. Visual chatgpt: Talking, drawing and editing with visual foundation models. *arXiv preprint arXiv:2303.04671*.

Wu, Q.; Bansal, G.; Zhang, J.; Wu, Y.; Zhang, S.; Zhu, E.; Li, B.; Jiang, L.; Zhang, X.; and Wang, C. 2023b. Autogen: Enabling next-gen llm applications via multi-agent conversation framework. *arXiv preprint arXiv:2308.08155*.

Wu, Y.; Wang, Y.; Tang, S.; Wu, W.; He, T.; Ouyang, W.; Wu, J.; and Torr, P. 2024. Dettoolchain: A new prompting paradigm to unleash detection ability of mllm. *arXiv preprint arXiv:2403.12488*.

Xi, Z.; Chen, W.; Guo, X.; He, W.; Ding, Y.; Hong, B.; Zhang, M.; Wang, J.; Jin, S.; Zhou, E.; et al. 2023. The rise and potential of large language model based agents: A survey. *arXiv preprint arXiv:2309.07864*.

Xie, J.; Chen, Z.; Zhang, R.; Wan, X.; and Li, G. 2024a. Large multimodal agents: A survey. *arXiv preprint arXiv:2402.15116*.

Xie, J.; Zhang, K.; Chen, J.; Zhu, T.; Lou, R.; Tian, Y.; Xiao, Y.; and Su, Y. 2024b. Travelplanner: A benchmark for real-world planning with language agents. *arXiv preprint arXiv:2402.01622*.

Yan, A.; Yang, Z.; Zhu, W.; Lin, K.; Li, L.; Wang, J.; Yang, J.; Zhong, Y.; McAuley, J.; Gao, J.; et al. 2023. Gpt-4v in wonderland: Large multimodal models for zero-shot smartphone gui navigation. *arXiv preprint arXiv:2311.07562*.

Yang, J.; Zhang, H.; Li, F.; Zou, X.; Li, C.; and Gao, J. 2023a. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv preprint arXiv:2310.11441*.

Yang, L.; Kang, B.; Huang, Z.; Zhao, Z.; Xu, X.; Feng, J.; and Zhao, H. 2024a. Depth Anything V2. *arXiv preprint arXiv:2406.09414*.

Yang, Y.; Zhou, T.; Li, K.; Tao, D.; Li, L.; Shen, L.; He, X.; Jiang, J.; and Shi, Y. 2024b. Embodied Multi-Modal Agent trained by an LLM from a Parallel TextWorld. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 26275–26285.

Yang, Z.; Li, L.; Wang, J.; Lin, K.; Azarnasab, E.; Ahmed, F.; Liu, Z.; Liu, C.; Zeng, M.; and Wang, L. 2023b. Mm-react: Prompting chatgpt for multimodal reasoning and action. *arXiv preprint arXiv:2303.11381*.

Yang, Z.; Liu, J.; Han, Y.; Chen, X.; Huang, Z.; Fu, B.; and Yu, G. 2023c. Appagent: Multimodal agents as smartphone users. *arXiv preprint arXiv:2312.13771*.

Yao, S.; Yu, D.; Zhao, J.; Shafran, I.; Griffiths, T.; Cao, Y.; and Narasimhan, K. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.

Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; and Cao, Y. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; and Cao, Y. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. In *International Conference on Learning Representations (ICLR)*.

You, K.; Zhang, H.; Schoop, E.; Weers, F.; Swearngin, A.; Nichols, J.; Yang, Y.; and Gan, Z. 2024. Ferret-UI: Grounded Mobile UI Understanding with Multimodal LLMs. *arXiv preprint arXiv:2404.05719*.

Yue, X.; Ni, Y.; Zhang, K.; Zheng, T.; Liu, R.; Zhang, G.; Stevens, S.; Jiang, D.; Ren, W.; Sun, Y.; et al. 2024. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9556–9567.

Zhang, J.; Xu, X.; and Deng, S. 2023. Exploring collaboration mechanisms for llm agents: A social psychology view. *arXiv preprint arXiv:2310.02124*.

Zhang, R.; Zhou, Y.; Chen, J.; Gu, J.; Chen, C.; and Sun, T. 2024a. LLaVA-Read: Enhancing Reading Ability of Multimodal Language Models. *arXiv preprint arXiv:2407.19185*.

Zhang, Y.; Feng, S.; and Tan, C. 2022. Active Example Selection for In-Context Learning. In Goldberg, Y.; Kozareva, Z.; and Zhang, Y., eds., *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 9134–9148. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Zhang, Y.; Zhou, K.; and Liu, Z. 2023. What makes good examples for visual in-context learning? *Advances in Neural Information Processing Systems*, 36: 17773–17794.

Zhang, Z.; Chen, J.; and Yang, D. 2024. DARG: Dynamic Evaluation of Large Language Models via Adaptive Reasoning Graph. *arXiv preprint arXiv:2406.17271*.

Zhang, Z.; Gao, Y.; and Lou, J.-G. 2024. E^5: Zero-shot Hierarchical Table Analysis using Augmented LLMs via Explain, Extract, Execute, Exhibit and Extrapolate. In Duh, K.; Gomez, H.; and Bethard, S., eds., *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 1244–1258. Mexico City, Mexico: Association for Computational Linguistics.

Zhang, Z.; Li, X.; Gao, Y.; and Lou, J.-G. 2023. CRT-QA: A Dataset of Complex Reasoning Question Answering over Tabular Data. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2131–2153. Singapore: Association for Computational Linguistics.

Zhang, Z.; Yang, F.; Jiang, Z.; Chen, Z.; Zhao, Z.; Ma, C.; Zhao, L.; and Liu, Y. 2024b. Position-Aware Parameter Efficient Fine-Tuning Approach for Reducing Positional Bias in LLMs. *arXiv preprint arXiv:2404.01430*.

Zhao, H.; Cai, Z.; Si, S.; Ma, X.; An, K.; Chen, L.; Liu, Z.; Wang, S.; Han, W.; and Chang, B. 2023. Mmicl: Empowering vision-language model with multi-modal in-context learning. *arXiv preprint arXiv:2309.07915*.

Zheng, B.; Gou, B.; Kil, J.; Sun, H.; and Su, Y. 2024a. Gpt-4v (ision) is a generalist web agent, if grounded. *arXiv preprint arXiv:2401.01614*.

Zheng, L.; Huang, Z.; Xue, Z.; Wang, X.; An, B.; and Yan, S. 2024b. Agentstudio: A toolkit for building general virtual agents. *arXiv preprint arXiv:2403.17918*.

Zhou, D.; Schärli, N.; Hou, L.; Wei, J.; Scales, N.; Wang, X.; Schuurmans, D.; Cui, C.; Bousquet, O.; Le, Q.; et al. 2022. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.

Zhou, S.; Xu, F. F.; Zhu, H.; Zhou, X.; Lo, R.; Sridhar, A.; Cheng, X.; Ou, T.; Bisk, Y.; Fried, D.; et al. 2023. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*.