VS-BENCH: EVALUATING VLMS FOR STRATEGIC ABILITIES IN MULTI-AGENT ENVIRONMENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent advancements in Vision Language Models (VLMs) have expanded their capabilities to interactive agent tasks, yet existing benchmarks remain limited to single-agent or text-only environments. In contrast, real-world scenarios often involve multiple agents interacting within rich visual and textual contexts, posing challenges with both multimodal observations and strategic interactions. To bridge this gap, we introduce VISUAL STRATEGIC BENCH (VS-BENCH), a multimodal benchmark that evaluates VLMs for strategic abilities in multi-agent environments. VS-BENCH comprises ten vision-grounded environments that cover cooperative, competitive, and mixed-motive interactions. The performance of VLM agents is evaluated across three dimensions: perception measured by element recognition accuracy; strategic reasoning measured by next-action prediction accuracy; and decision-making measured by normalized episode return. Extensive experiments on fifteen leading VLMs show that, although current models exhibit strong perception abilities, there remains a significant gap to optimal performance in reasoning and decision-making, with the best-performing model attaining 46.6% prediction accuracy and 31.4% normalized return. We further analyze the key factors influencing performance, conduct human experiments, and examine failure modes to provide a deeper understanding of VLMs' strategic abilities. By standardizing the evaluation and highlighting the limitations of existing models, we envision VS-BENCH as a foundation for future research on strategic multimodal agents. Code and data are available at `https://sites.google.com/view/vs-bench`.

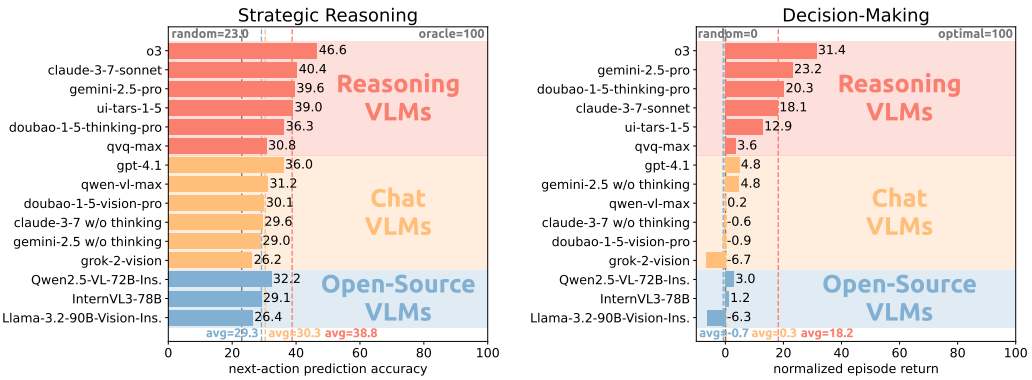


Figure 1: Evaluation results of fifteen state-of-the-art VLMs on strategic reasoning and decision-making averaged over ten multi-agent environments in VS-BENCH.

1 INTRODUCTION

Vision Language Models (VLMs) have recently unlocked impressive capabilities in open-world perception, multimodal reasoning, and interactive problem-solving (Bai et al., 2023; Liu et al., 2023; Zhu et al., 2025). Driven by these advancements, evaluations of VLMs have progressed beyond

static tasks such as image captioning (Chen et al., 2015) and visual reasoning (Antol et al., 2015; Yue et al., 2024) toward dynamic agent benchmarks including software engineering (Chan et al., 2024; Yang et al., 2024), computer use (Koh et al., 2024; Xie et al., 2024b), game environments (Zhang et al., 2023; Wang et al., 2025), and embodied control (Guruprasad et al., 2024; Team et al., 2025).

However, existing VLM benchmarks mainly focus on single-agent settings, where one agent reasons and acts in isolation. The real world, by contrast, is inherently a multi-agent environment that involves cooperation, competition, and mixed-motive interactions (Ferber & Weiss, 1999; Wooldridge, 2009). This poses new challenges to the ability of intelligent agents. First, an agent's outcome depends not only on its own action but also on others' actions, requiring theory-of-mind reasoning (Kosinski, 2023) to infer others' intentions and future moves. Second, as all agents learn and adapt concurrently, the underlying dynamics become non-stationary, demanding decision-making under uncertainty that optimizes for long-term objectives. Third, the coexistence of cooperation and competition gives rise to social dilemmas where agents must strategically balance self-interest and collective welfare. These challenges raise a crucial question that current benchmarks leave underexplored: ***How capable are VLMs in terms of strategic abilities in multi-agent environments?***

While prior efforts (Agashe et al., 2023; Duan et al., 2024; Xie et al., 2024a) have explored multi-agent evaluation for Large Language Models (LLMs), these benchmarks remain restricted to text-only environments, limiting their capability to assess agents in multimodal scenarios. On the one hand, many strategic domains (Bellemare et al., 2013; Lowe et al., 2017; Carroll et al., 2019) intrinsically rely on visual observations like game frames that encode rich spatial layouts, motion cues, and temporal dynamics. Reducing these visual observations to text-only descriptions inevitably discards critical information that is essential for reasoning and decision-making. On the other hand, humans naturally receive and convey both visual and textual information when interacting with others. Purely text-based environments diverge from real-world human-agent interactions and obscure progress toward developing human-compatible intelligent agents. These limitations underscore the need for a multimodal benchmark that incorporates visual context in multi-agent environments.

To bridge this gap, we introduce VISUAL STRATEGIC BENCH (VS-BENCH), a multimodal benchmark that evaluates VLMs' strategic abilities in multi-agent environments. VS-BENCH comprises ten vision-grounded environments that cover three multi-agent dynamics including cooperative, competitive, and mixed-motive interactions. VLMs are evaluated across three dimensions: perception, strategic reasoning, and decision-making. Perception requires extracting key information from multimodal observations and is evaluated by VLMs' recognition accuracy of key elements. Strategic reasoning is the theory-of-mind ability to infer other agents' intentions and is evaluated by VLMs' prediction accuracy of others' next actions. Decision-making is the ability to optimize for long-term objectives under non-stationary dynamics and is evaluated by the normalized episode returns of VLM agents. By jointly analyzing all three perspectives, our benchmark provides a comprehensive evaluation of VLMs' strategic abilities in multi-agent environments.

We evaluate fifteen leading VLMs, including six commercial reasoning models, six commercial chat models, and three open-source models on VS-BENCH. Extensive results show that while current VLMs exhibit strong perception ability, there remains a significant gap to optimal performance in strategic reasoning and decision-making, with the best-performing model attaining 46.6% prediction accuracy and 31.4% normalized return averaged across all environments. Notably, although commercial reasoning models in general achieve the best results, some open-source models can achieve comparable performance in certain mixed-motive games with prosocial behaviors for mutual benefit. We further perform analyses on multimodal observations, test-time scaling, social behaviors, human experiments, and failure modes to provide a deeper understanding of VLMs' strategic abilities.

In summary, our contributions are threefold:

- We introduce VS-BENCH, a multimodal benchmark for evaluating VLMs in ten vision-grounded multi-agent environments across cooperative, competitive, and mixed-motive interactions.
- We consider three evaluation dimensions, including perception measured by element recognition accuracy, strategic reasoning measured by next-action prediction accuracy, and decision-making measured by normalized episode returns, to provide a comprehensive assessment of VLM agents.
- We perform extensive experiments of twelve commercial VLMs and three open-source VLMs and provide in-depth analyses of multimodal observation, test-time scaling, social behaviors, human experiments, and failure modes, highlighting significant performance gaps for future research.

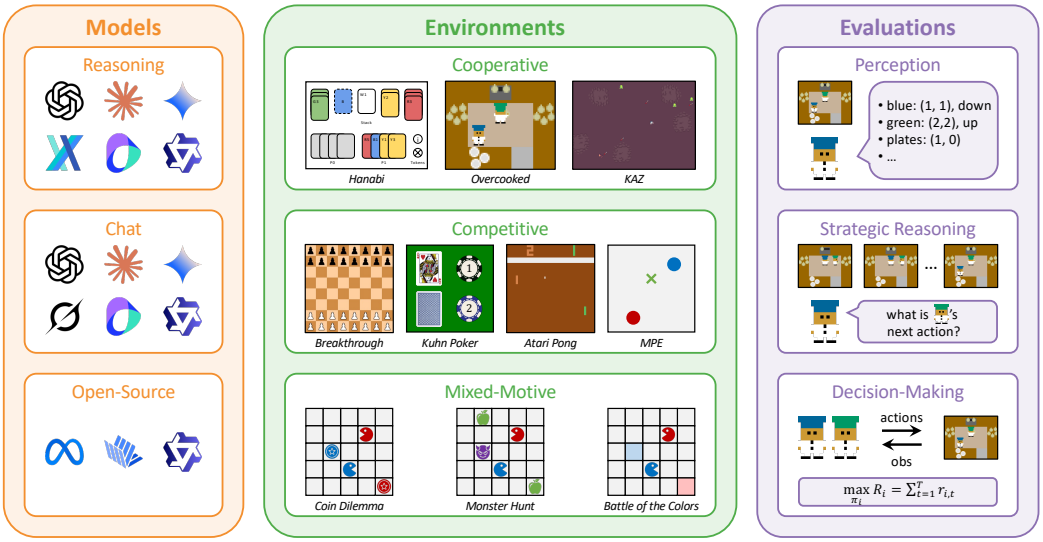


Figure 2: Overview of VS-BENCH, a multimodal benchmark for evaluating VLMs in multi-agent environments. We evaluate fifteen models in ten vision-grounded environments across three dimensions, including perception measured by element recognition accuracy, strategic reasoning measured by next-action prediction accuracy, and decision-making measured by normalized episode return.

2 VS-BENCH ENVIRONMENTS

In this section, we first formalize the evaluation of VLMs in multi-agent environments and then introduce ten vision-grounded games comprising VS-BENCH. These games are carefully curated from game theory and multi-agent reinforcement learning (MARL), each serving as a well-recognized environment in the literature. We further adapt these games to incorporate image and text observations while preserving their strategic dynamics. By covering cooperation, competition, and mixed-motive interactions, these games provide a diverse set of multi-agent environments for evaluating VLMs. An additional set of simpler games and three-player games can be found in Appendix H.

2.1 PROBLEM FORMULATION

Multi-agent environments are formalized as Partially Observable Markov Games (POMG) (Shapley, 1953). A POMG is defined by a tuple $\mathcal{G} = (\mathcal{N}, \mathcal{S}, \{\mathcal{A}_i\}_{i\in\mathcal{N}}, \{\mathcal{O}_i\}_{i\in\mathcal{N}}, \mathcal{P}, \{\mathcal{R}_i\}_{i\in\mathcal{N}}, \gamma)$, where $\mathcal{N} = \{1, \cdots, n\}$ is the set of agents; $\mathcal{S}$ is the state space; $\mathcal{A}_i$ and $\mathcal{O}_i$ are the action space and observation space of agent i, respectively; $\mathcal{P} : \mathcal{S} \times \{\mathcal{A}_i\}_{i\in\mathcal{N}} \to \Delta(\mathcal{S})$ is the transition function; $\mathcal{R}_i : \mathcal{S} \times \{\mathcal{A}_i\}_{i\in\mathcal{N}} \to \mathbb{R}$ is the reward function of agent i; and γ is the discount factor. In step t, each agent i receives an observation $o_{i,t}$ and chooses an action $a_{i,t}$ according to its policy π_i. Given the current state s_t and the joint action $a_t = (a_{1,t}, \cdots, a_{n,t})$, the environment transitions to the next state $s_{t+1} \sim \mathcal{P}(s_t, a_t)$ and each agent i receive a reward $r_{i,t} = \mathcal{R}_i(s_t, a_t)$. The objective of agent i is to maximize its expected accumulated reward $\mathbb{E}_{\pi_1,\cdots,\pi_n}[\sum_t \gamma^t r_{i,t}]$. To evaluate VLM agents, we consider a multimodal observation space $\mathcal{O}_i = (\mathcal{I}_i, \mathcal{T}_i)$, where $\mathcal{I}_i$ is the image observation space and $\mathcal{T}_i$ is the text observation space. We also consider a text action space $\tilde{\mathcal{A}}_i$ and a mapping function $\mathcal{M} : \tilde{\mathcal{A}}_i \to \mathcal{A}_i$ that converts each text action into the original action space.

2.2 MULTI-AGENT ENVIRONMENTS

We consider three types of multi-agent environments, including cooperative, competitive, and mixed-motive games. The taxonomy and required abilities of the games in VS-BENCH are listed in Table 1. The detailed description of each game and required ability can be found in Appendix I.

Cooperative. All agents share the same objective in cooperative games. Formally, the reward functions in cooperative games are identical: $\mathcal{R}_1(s, a) = \cdots = \mathcal{R}_n(s, a)$ for all $(s, a) \in \mathcal{S} \times \{\mathcal{A}_i\}_{i\in\mathcal{N}}$.

Table 1: Taxonomy and required abilities of the ten multi-agent environments in VS-BENCH.

Type	Games	Full Obs.	Deter-ministic	Simul-taneous	Sym-metric	Spatial Perception	Theory of Mind	Long-Term Planning	Team Collaboration
Cooperative	*Hanabi*	✗	✗	✗	✗	☆	☆☆☆	☆☆☆	☆☆☆
	Overcooked	✓	✓	✓	✗	☆☆	☆☆	☆☆	☆☆☆
	KAZ	✓	✗	✓	✗	☆☆☆	☆	☆☆	☆☆
Competitive	*Board* [1]	✓	✓	✗	✗	☆☆	☆☆☆	☆☆☆	☆
	Poker	✗	✗	✗	✗	☆	☆☆☆	☆	☆
	Pong	✓	✗	✓	✓	☆☆☆	☆	☆	☆
	MPE	✓	✗	✓	✗	☆☆☆	☆☆	☆☆	☆
Mixed-Motive	*Dilemma*	✓	✗	✓	✓	☆☆	☆☆	☆☆	☆
	Hunt	✓	✗	✓	✓	☆☆	☆☆☆	☆☆	☆☆
	Battle	✓	✗	✓	✓	☆☆	☆☆	☆☆	☆☆☆

To achieve strong performance in cooperative games, agents must understand their teammates' intentions under partial observability, divide the tasks to improve efficiency, and coordinate their actions to optimize for the shared objective. We consider three representative cooperative games in the MARL literature, namely *Hanabi* (Bard et al., 2020), *Overcooked* (Games, 2016), and *Knights Archers Zombies (KAZ)* (Terry et al., 2021).

Competitive. The objective of each agent strictly contradicts those of the others in competitive games. Formally, the reward functions in competitive games are zero-sum: $\sum_{i=1}^{n} \mathcal{R}_i(s, a) = 0$ for all $(s, a) \in \mathcal{S} \times \{\mathcal{A}_i\}_{i \in \mathcal{N}}$. To succeed in competitive games, agents must model their opponents to predict their future moves, stay robust against adversarial exploitation, and adapt to non-stationary dynamics. We consider four representative competitive games in game theory and MARL literature, namely *Breakthrough* (Troyka, 2000), *Kuhn Poker* (Kuhn, 1950), *Atari Pong* (Atari, 1972), and *Multi-agent Particle Environment (MPE)* (Lowe et al., 2017).

Mixed-motive. Agents' objectives are partially aligned and partially divergent in mixed-motive games. Formally, the reward functions are neither identical nor zero-sum, that is, there exists (s, a) such that $\mathcal{R}_i(s, a) \neq \mathcal{R}_j(s, a)$ and $\sum_{i=1}^{n} \mathcal{R}_i(s, a) \neq 0$. To excel in mixed-motive games, agents must anticipate the hidden intentions of others, balance self-interest and common welfare, and achieve high-payoff equilibria. We consider three mixed-motive games adapted from classic social dilemmas in game theory, namely *Coin Dilemma* (Lerer & Peysakhovich, 2017), *Monster Hunt* (Peysakhovich & Lerer, 2017), *Battle of the Colors*.

3 EVALUATING VLMS IN MULTI-AGENT ENVIRONMENTS

To comprehensively benchmark VLMs in multi-agent environments, we consider three evaluation dimensions including perception, strategic reasoning, and decision-making. We also provide several findings that highlight the limitations of existing VLMs and motivate our analysis in the next section.

Model setup. We select fifteen state-of-the-art VLMs for evaluation. For commercial VLMs, we select six reasoning models and six chat models from OpenAI GPT (OpenAI, 2025a) and o-series (OpenAI, 2025b), Anthropic Claude (Anthropic, 2025), Google Gemini (DeepMined, 2025), xAI Grok (xAI, 2024), Qwen (team, 2025), and Doubao (seed, 2025). For open-source VLMs, we select three leading models from Llama-3.2-Vision (meta, 2024), InternVL3 (Zhu et al., 2025), and Qwen2.5-VL (Bai et al., 2025). We set the temperature to 1.0 and the maximum number of output tokens to 8k for all models. We also set the maximum number of reasoning tokens to 16k for reasoning models. When encountering a cutoff for reaching maximum tokens, we dynamically extend the tokens to the model's limit. Detailed descriptions of model setups can be found in Appendix B.

3.1 PERCEPTION

Perception is the ability to extract and represent key information from raw multimodal observations. This requires agents to accurately recognize entities, spatial relations, and numerical attributes to

[1] *Board* corresponds to *Breakthrough*, and the remaining rows below it correspond to *Kuhn Poker*, *Atari Pong*, *MPE*, *Coin Dilemma*, *Monster Hunt*, and *Battle of the Colors*.

Table 2: Perception evaluation results. For each environment, the first, second, and third best results are highlighted in green.

Models[2]	Overall	Cooperative			Comptitive			Mixed-Motive			
		Hanabi	*Overcooked*	*KAZ*	*Board*	*Poker*	*Pong*	*MPE*	*Dilemma*	*Hunt*	*Battle*
Oracle	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
o3	84.9	79.7	69.8	90.1	97.2	99.1	64.6	97.5	85.5	80.2	85.4
gemini-2.5-pro	83.4	79.9	54.5	88.6	98.5	100.0	86.5	96.5	76.4	73.0	79.9
claude-3-7-sonnet	81.1	73.0	62.8	92.9	75.2	99.5	69.4	89.8	82.6	79.7	85.7
ui-tars-1-5	81.5	75.4	59.8	83.0	98.0	100.0	74.4	92.2	76.0	74.9	81.0
doubao-1-5-thinking-pro	74.7	42.2	52.4	77.5	91.0	98.6	66.0	87.2	76.7	74.9	80.0
qvq-max	74.5	75.1	63.6	74.6	83.3	95.2	50.7	83.6	69.5	72.0	77.7
gemini-2.5 w/o thinking	84.5	79.9	38.8	82.8	88.2	97.2	84.3	97.3	92.5	93.1	90.8
claude-3-7 w/o thinking	80.5	75.9	59.7	85.6	79.0	99.6	70.1	90.6	81.8	80.4	82.8
gpt-4.1	80.3	72.1	62.0	92.6	67.0	100.0	75.4	98.9	76.7	76.8	81.2
qwen-vl-max	80.2	76.1	68.2	85.2	81.2	99.2	69.0	88.2	78.4	76.4	80.6
doubao-1-5-vision-pro	77.6	80.0	33.1	67.8	89.3	100.0	77.7	90.8	78.0	77.6	81.2
grok-2-vision	70.2	75.2	46.8	59.8	80.3	59.5	71.0	78.7	76.4	73.3	81.0
Qwen2.5-VL-72B-Ins.	80.3	76.0	72.9	85.5	75.1	100.0	65.2	87.7	79.8	79.0	82.4
InternVL3-78B	74.1	74.6	43.6	63.8	64.3	99.2	66.7	89.2	81.1	76.7	81.8
Llama-3.2-90B-Vision-Ins.	67.8	30.7	58.6	87.1	59.7	81.6	59.5	94.3	68.5	66.0	72.4
Random	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

facilitate downstream reasoning and decision-making. We evaluate the perception ability of VLMs by measuring their recognition accuracy of key elements like object position, agent orientation, and game status. We collect 400 samples with ground-truth labels from actual game-play in each environment. Detailed descriptions of perception evaluation can be found in Appendix C.

Existing VLMs show competent perception ability. The evaluation results in Fig. 1 and Table 2 show that current VLMs achieve high performance in perception of key elements. All models achieve at least 67.8% overall accuracy and the best model o3 attains 84.9% overall accuracy. Reasoning models do not show a significant advantage over chat models and open-source models. These results show that all VLMs demonstrate adequate perception ability for subsequent strategic reasoning and decision-making in multi-agent environment.

3.2 Strategic Reasoning

Strategic reasoning is the theory-of-mind ability to infer the hidden beliefs, desires, and intentions of other agents (Rao et al., 1995; Kosinski, 2023). This requires agents to think from others' perspectives to answer: *What are the next actions of other agents?* Strategic reasoning is crucial in multi-agent environments because an agent's reward function depends not only on its own action, but also on other participants' actions. We evaluate the strategic reasoning ability of VLMs by their prediction accuracy of other agents' next actions on a dataset for each environment. To ensure a rigorous and thorough evaluation, we selectively collect 400 samples and construct a predictable, diverse, and balanced dataset for each environments. Detailed descriptions of dataset construction and strategic reasoning evaluation can be found in Appendix D.

Existing VLMs exhibit preliminary strategic reasoning ability but are still far from oracle performance. The evaluation results in Fig. 1 and Table 3 show that current VLMs show basic strategic reasoning ability by surpassing random agents in overall prediction accuracy, yet they still lag behind the oracle results by a noticeable margin. Most models perform better than random guesses in at least eight of the ten games, demonstrating non-trivial theory-of-mind capability in multi-agent environments. In general, reasoning models achieve better results than chat models and open-source models, with the best-performing model o3 attaining an overall accuracy of 46.6% and ranking first in six environments. Notably, the three leading open-source models achieve an average overall accuracy of 29.2%, which is comparable to commercial chat models with a 30.3% average overall accuracy. However, even these most capable models attain less than 50% overall accuracy, and some even fail to outperform random. This deficit is especially pronounced in *Overcooked*, *Atari Pong*, and *Monster Hunt*, which are all adapted from video games.

[2]Specific model versions and links to open-source models can be found in Appendix B.

Table 3: Strategic reasoning evaluation results. For each environment, the first, second, and third best results are highlighted in green, while the results below random are highlighted in red.

Models	Overall	Cooperative			Comptitive			Mixed-Motive			
		Hanabi	*Overcooked*	*KAZ*	*Board*	*Poker*	*Pong*	*MPE*	*Dilemma*	*Hunt*	*Battle*
Oracle	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
`o3`	46.6	61.2	40.5	30.8	29.0	67.0	25.8	33.8	59.2	57.5	61.5
`claude-3-7-sonnet`	40.4	39.0	26.0	29.0	24.2	65.2	44.8	35.8	53.8	42.5	43.2
`gemini-2.5-pro`	39.6	51.2	22.0	28.8	26.8	59.8	24.5	35.0	52.5	35.8	60.0
`ui-tars-1-5`	39.0	25.5	18.2	38.5	23.2	65.8	30.8	33.5	63.0	41.5	50.0
`doubao-1-5-thinking-pro`	36.3	32.8	26.2	27.5	19.8	65.8	44.2	34.8	26.5	45.2	40.0
`qvq-max`	30.8	32.2	19.0	24.5	21.8	63.0	37.8	35.5	25.2	21.5	27.0
`gpt-4.1`	36.0	23.0	27.0	27.8	22.5	58.2	41.5	35.8	49.8	36.8	38.0
`qwen-vl-max`	31.2	26.5	26.0	46.0	19.5	52.2	23.5	31.0	26.2	23.5	37.2
`doubao-1-5-vision-pro`	30.1	15.0	22.2	18.5	15.8	58.2	31.2	31.5	37.2	36.0	34.8
`claude-3-7` w/o thinking	29.6	9.8	16.0	35.8	18.0	57.2	43.2	32.2	31.0	26.0	26.8
`gemini-2.5` w/o thinking	29.0	21.5	19.2	20.8	14.8	56.5	34.0	27.0	31.8	30.5	33.8
`grok-2-vision`	26.2	12.8	17.2	25.5	10.8	59.5	20.8	25.2	30.2	31.5	29.0
`Qwen2.5-VL-72B-Ins.`	32.2	26.8	26.5	39.5	23.8	50.8	27.0	34.2	30.0	27.2	36.8
`InternVL3-78B`	29.1	25.2	20.5	24.0	14.0	45.2	34.8	30.2	37.0	30.0	30.2
`Llama-3.2-90B-Vision-Ins.`	26.4	20.0	16.5	20.8	11.8	52.8	36.2	32.8	22.2	26.2	25.0
Random	23.0	8.8	16.7	16.2	4.2	50.0	33.3	19.5	25.4	29.3	26.5

3.3 Decision-making

Decision-making is the ability to optimize for one's long-term objective under uncertainty (Edwards, 1954). This requires agents to prioritize future accumulated returns over immediate gains, adapt to non-stationary dynamics with evolving agents, and balance cooperation and competition for favorable equilibria. We evaluate the decision-making ability of VLMs by their normalized episode returns through self-play or interactions with conventional agents in each environment. We also evaluate the random agents and the oracle agents with maximum return to normalize the results so that the normalized return for random agents is 0 and the normalized return for oracle agents is 100. Detailed descriptions of decision-making evaluation can be found in Appendix E.

Existing VLMs struggle with decision-making in multi-agent environments. The evaluation results in Fig. 1 and Table 4 show that current VLMs have limited decision-making ability in multi-agent environments, highlighting a significant gap between existing models and oracle performance. As illustrated by the large swaths of red cells, four out of fifteen models' overall performance is even worse than random agents, indicating their incompetence to optimize long-term return under non-stationary, interdependent multi-agent dynamics. Although reasoning models achieve relatively better results than chat models and open-source models, even the best-performing model `o3` only attains an overall normalized return of 31.4%, which is far behind the oracle agent. Surprisingly, we observe that some open-source models can achieve comparable results to reasoning models in certain mixed-motive games like `InternVL3-78B` in *Coin Dilemma* and `Qwen2.5-VL-72B-Ins.` in *Monster Hunt*. We also observe that the failures to outperform random agents are concentrated on video games like *Overcooked*, *KAZ*, *Atari Pong*, and *Coin Dilemma*, which underscores the coupled difficulty of multimodal perception and strategic decision-making.

4 Analysis

In this section, we aim to investigate the key factors that influence the performance of VLMs in multi-agent environments. We perform in-depth analyses on multimodal observations, test-time scaling, and social behaviors. We also compare VLMs with human performance and provide failure cases for future development. More experiment results can be found in Appendix F.

4.1 Multimodal observations

Although VLMs achieve good performance in multimodal perception, the evaluations on reasoning and decision-making show that environments with rich visual observations are especially challenging for VLM agents. To investigate the influence of multimodal observations, we further evaluate VLMs with text-only observations that replace images with corresponding text descriptions. We se-

Table 4: Decision-making evaluation results. For each environment, the first, second, and third best results are highlighted in green, while the results below or equal to random are in red.

Models	Overall	Cooperative			Comptitive			Mixed-Motive			
		Hanabi	*Overcooked*	*KAZ*	*Board*	*Poker*	*Pong*	*MPE*	*Dilemma*	*Hunt*	*Battle*
Oracle	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
`o3`	31.4	55.8	15.6	29.5	65.0	61.8	8.6	37.2	−0.4	24.0	16.7
`gemini-2.5-pro`	23.2	32.9	17.1	8.4	55.0	27.3	6.5	38.9	−9.6	21.5	33.8
`doubao-1-5-thinking-pro`	20.3	56.7	10.1	14.1	10.0	48.5	2.9	38.5	0.7	17.2	4.0
`claude-3-7-sonnet`	18.1	6.7	10.1	5.3	20.0	72.9	−0.5	39.4	4.6	19.9	2.5
`ui-tars-1-5`	12.9	0.0	2.0	−4.7	5.0	36.4	−0.9	19.4	33.1	24.9	13.6
`qvq-max`	3.6	0.0	2.0	−1.7	5.0	−8.7	0.4	38.6	0.0	0.7	−0.5
`gpt-4.1`	4.8	0.0	−0.5	−5.3	0.0	−7.1	0.2	31.5	17.8	11.2	0.5
`gemini-2.5` w/o thinking	4.8	0.0	2.0	0.0	0.0	18.2	1.0	23.8	−0.7	0.7	2.5
`qwen-vl-max`	0.2	1.2	−0.5	−5.3	0.0	−20.5	−0.3	14.4	−0.4	13.2	−0.5
`claude-3-7` w/o thinking	−0.6	0.0	2.0	3.5	5.0	−23.6	−0.9	5.6	1.4	0.2	1.0
`doubao-1-5-vision-pro`	−0.9	0.0	−0.5	−5.3	0.0	−40.2	−0.9	32.9	−2.1	7.8	−0.5
`grok-2-vision`	−6.7	0.0	1.5	0.0	0.0	−11.8	−0.1	−58.2	1.1	−0.4	0.5
`Qwen2.5-VL-72B-Ins.`	3.0	0.8	−0.5	−5.3	0.0	−3.1	−0.8	19.6	0.0	19.6	−0.5
`InternVL3-78B`	1.2	0.0	0.0	−3.5	0.0	4.0	−0.9	5.9	5.0	−0.2	1.5
`Llama-3.2-90B-Vision-Ins.`	−6.3	0.0	1.5	0.0	0.0	−39.4	−0.9	−29.6	0.4	3.6	1.0
Random	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

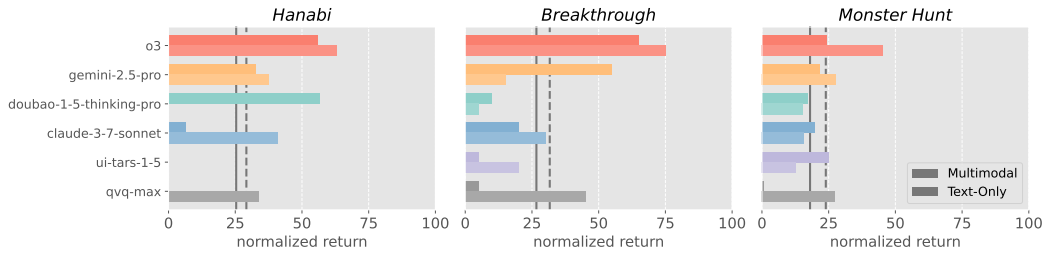


Figure 3: Comparison of reasoning VLMs on decision-making with multimodal and text-only observations. The solid and dashed vertical lines represent the average results of two settings.

lect a board game, a card game, and a video game. The decision-making results of reasoning models in Fig. 3 show that VLMs achieve slightly better results with text-only observations, but are still far from oracle results. This indicates that the weak decision-making performance arises from the coupled challenges of multimodal observation and strategic interactions in multi-agent environments.

4.2 Test-time Scaling

We observe in the evaluation results that reasoning models generally achieve better performance than chat models. We further investigate the test-time scaling of VLMs in multi-agent environments by using Chain-of-Thought (CoT) (Wei et al., 2022) prompting for chat models and comparing their performance with reasoning models and chat models with simple IO prompting. The evaluation results in Fig. 4 show that CoT prompting significantly improves chat models' performance in all three environments, while reasoning models still achieve the best results. This demonstrates that test-time scaling like reasoning and CoT prompting can substantially improve VLMs' performance.

4.3 Social Behaviors and Personas

Another interesting observation in the evaluation results is that open-source models can achieve comparable results to reasoning models in some mixed-motive games. We investigate this by visualizing the behaviors of `o3` and the best-performing open-source models in each social dilemma game. As shown in Fig. 5, in *Coin Dilemma*, `o3` are better at collecting coins but are also more self-interested, while `InterVL3-78B` is more inclined to cooperation that leads to a win-win situation. Similar behaviors can be found in *Monster Hunt*, where the two reasoning models tend to safely eat apples alone, while `Qwen2.5-VL-72B-Ins.` prefers taking the risk to cooperate and defeat the

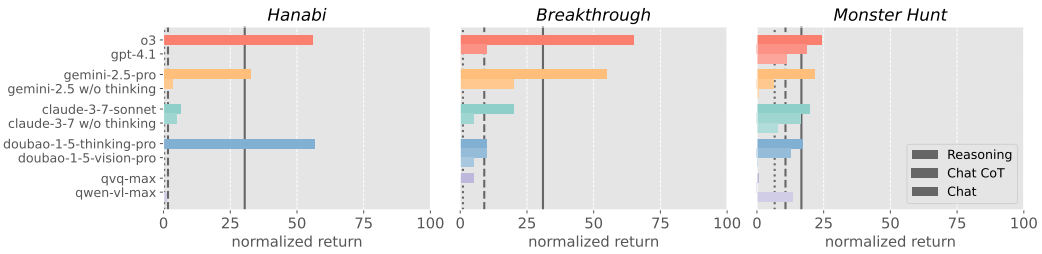


Figure 4: Comparison of reasoning VLMs and chat VLMs on decision-making with IO and CoT prompting. The solid, dashed, and dotted vertical lines represent the average results of three settings.

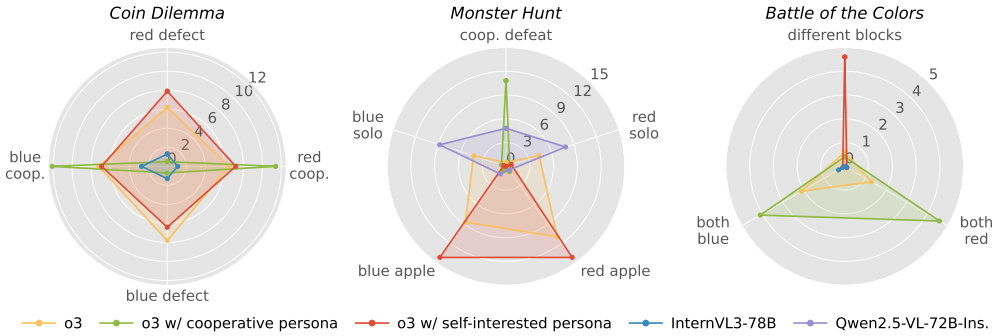


Figure 5: Social behaviors of `o3` with different personas and the best-performing open-source model in each social dilemma game. Dimensions are agents' behaviors described in Appendix I.

monster together. We also explicitly prompt `o3` with self-interested and cooperative personas, and find that different personas significantly change the behaviors and performance of VLMs.

4.4 Human Experiments

We further conduct human experiments in all ten games to better understand the current capability of VLMs in multi-agent environments. To ensure a fair comparison, we let human participants receive the same observations and play games for the same number of times as the VLMs. We collect the normalized episode returns of 26 human participants and visualize their distribution. As shown in Fig. 6, human participants achieve an average overall normalized return of 62.7. For comparison, `o3` attains an overall return of 31.4, which surpasses 12.9% of human results; `gpt-4.1` attains an overall return of 4.4, which surpasses only 4.3% of human results.

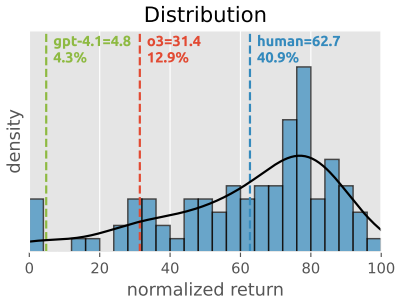


Figure 6: Human experiment results.

4.5 Failure Cases

To investigate why VLMs underperform in multi-agent environments, we conduct a qualitative analysis of their failure cases. In strategic reasoning, two common failure cases are ignoring history and private information. For example, in *Hanabi*, players' cards are observable to other agents but not to themselves. VLMs often overlook this information asymmetry and incorrectly use their private information to predict the next actions of others. In decision-making, another common failure case is focusing excessively on one's own actions while ignoring those of others. For example, in *Breakthrough*, VLMs tend to persistently advance their own pieces and fail to identify defensive vulnerabilities that directly result in losing the match. More failure cases can be found in Appendix G.

5 RELATED WORK

5.1 MULTI-AGENT ENVIRONMENTS AND BENCHMARKS

Early work on multi-agent reasoning and decision-making is grounded in classical game theory (Fudenberg & Tirole, 1991; Von Neumann & Morgenstern, 2007), which models rational interactions among self-interested players and introduces canonical environments like board games (Shannon, 1950; Tesauro, 1994), card games (Kuhn, 1950; Southey et al., 2012), and social dilemmas (Luce & Raiffa, 1957; Rapoport & Chammah, 1965; Rousseau, 1985). Building on these foundations, breakthroughs in multi-agent reinforcement learning (MARL) (Silver et al., 2016; Brown & Sandholm, 2019) have expanded the field toward complex, high-dimensional environments like video games. These environments cover a diverse range of multi-agent dynamics like cooperation (Carroll et al., 2019; Samvelyan et al., 2019; Bard et al., 2020), competition (Moravčík et al., 2017; Vinyals et al., 2019), and mixed-motive interactions (Lowe et al., 2017; Berner et al., 2019). Despite their impressive achievements, agents in these environments are typically specialized for a single task and lack general-purpose abilities for strategic reasoning and decision-making across different domains.

Recent advancements in Large Language Models (LLMs) (Ouyang et al., 2022; Touvron et al., 2023; Guo et al., 2025) have catalyzed a paradigm shift toward generalist agents that can perceive and act in various environments without task-specific training. A growing body of text-based benchmarks has been proposed to evaluate different facets of LLM agents in multi-agent environments. LLM-Coordination (Agashe et al., 2023) analyzes LLM agents' ability in pure cooperative tasks. GT-Bench (Duan et al., 2024) and GAMA-Bench (Huang et al., 2024) consider non-cooperative games and evaluate LLMs through the lens of game theory. Trust Games (Xie et al., 2024a) studies the trust behaviors of LLM agents in their interactions. MAgIC (Xu et al., 2023) and LLMArena (Chen et al., 2024) include both cooperative and competitive games for a comprehensive evaluation. However, these benchmarks mainly focus on text-only environments, which are different from real-world decision-making that integrates both visual and textual observations. Our work fills this gap by introducing ten vision-grounded games to evaluate multimodal agents in multi-agent environments.

5.2 VLM AGENT BENCHMARKS

The rapid evolution of Vision Language Models (VLMs) (Bai et al., 2023; Liu et al., 2023) has driven evaluation beyond static tasks like image captioning (Chen et al., 2015) and visual reasoning (Antol et al., 2015; Yue et al., 2024) toward interactive agent environments. Existing benchmarks can be broadly categorized into four domains: coding, GUI interaction, game environments, and embodied control. Coding benchmarks (Chan et al., 2024; Li et al., 2024a; Yang et al., 2024) consider software engineering and machine learning engineering with both image and text input. GUI benchmarks evaluate VLMs on graphic interface operations like web browsing (He et al., 2024; Koh et al., 2024; Zheng et al., 2024), computer use (Xie et al., 2024b; Bonatti et al., 2024), and phone use (Li et al., 2024b; Rawles et al., 2024). Game benchmarks (Zhang et al., 2023; Liu et al., 2024; Wang et al., 2025) offer dynamic virtual environments with structured rewards to assess VLMs. Embodied benchmarks (Guruprasad et al., 2024; Team et al., 2025; Yang et al., 2025) evaluate VLMs in vision-driven robotics control and physical world interactions. Nevertheless, these benchmarks predominantly concentrate on single-agent tasks, which overlook the distinctive challenges of multi-agent environments including non-stationary dynamics, interdependent decision-making, and equilibrium selection. Our work bridges this gap by evaluating VLMs' strategic ability in multi-agent games.

6 CONCLUSION

In this work, we present VS-BENCH, a multimodal benchmark for evaluating VLMs' strategic abilities in multi-agent environments. By introducing ten vision-grounded environments and three evaluation dimensions, including perception, strategic reasoning, and decision-making, we establish a comprehensive multi-agent evaluation of VLMs. Extensive experiments on fifteen leading VLMs reveal a significant gap between existing models and optimal performance on reasoning and decision-making, highlighting their limitations for future development. We further provide detailed analyses on multimodal observations, test-time scaling, social behaviors, human experiments, and failure cases of VLM agents. By releasing VS-BENCH as an open platform, we seek to spur research on strategic multimodal agents that excel in vision-grounded multi-agent environments.

ETHICS STATEMENT

Positive research and societal value. VS-Bench targets a core capability that future AI systems will increasingly need: making strategic, multi-step decisions while perceiving the world through vision and language. By standardising how this ability is measured, the benchmark can accelerate reproducible research on safer, more reliable multimodal agents. Concretely, it enables (1) principled comparisons across models, (2) diagnostic analyses that pinpoint specific failure modes such as myopic play or poor opponent modelling, and (3) a shared testbed for developing methods that foster cooperation, fairness, or robustness in complex interactive settings. Beyond academic progress, stronger decision-making agents could benefit applications like assistive household robotics, disaster-response swarms, automated traffic control, and large-scale scientific simulations where coordination and strategic planning are essential.

Risk of misuse and dual-use considerations. At the same time, more capable agents that reason strategically can be repurposed for adversarial or deceptive objectives — for example, collusive price-setting, automated disinformation campaigns, or the coordination of autonomous weapons systems. VS-Bench lowers the barrier to evaluating such capabilities, potentially making it easier to select or fine-tune models for harmful ends. To mitigate this, we (1) release only simulated environments that do not directly embody real-world attack surfaces, (2) distribute the benchmark and evaluation code under licenses that forbid the use of our assets in weaponised or surveillance applications, and (3) encourage follow-up work on safety safeguards (e.g., opponent-aware alignment checks) by providing explicit hooks for auditing model rationales and behaviors.

Human subjects protection. To minimize potential risks to human participants, we conduct the human experiment with departmental approval and enroll individuals only after obtaining informed consent. The consent form explicitly stated that the games involve cooperative, competitive, and mixed-motive interactions between participants. Participants are paid for taking part in this study and can choose to withdraw from the study at any time without consequences.

Privacy and data ethics. All VS-Bench environments are synthetic with no personally identifiable information or copyrighted third-party imagery that is not permissively licensed. Replays, logs, and intermediate states are derived entirely from simulation and are released under an open licence to avoid common privacy pitfalls in dataset creation and facilitate unrestricted academic use.

REPRODUCIBILITY STATEMENT

All data, code, and training scripts required to reproduce the results in this paper are released in anonymized repositories. The dataset is publicly available at `https://huggingface.co/datasets/VS-Bench/VS-Bench`. The code is publicly available at `https://anonymous.4open.science/r/vsbench-iclr`.

REFERENCES

Saaket Agashe, Yue Fan, Anthony Reyna, and Xin Eric Wang. Llm-coordination: evaluating and analyzing multi-agent coordination abilities in large language models. *arXiv preprint arXiv:2310.03903*, 2023.

Anthropic. Claude 3.7 sonnet system card, 2025.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pp. 2425–2433, 2015.

Atari. Pong. Arcade Video Game, 1972.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwenvll: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 2023.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Nolan Bard, Jakob N Foerster, Sarath Chandar, Neil Burch, Marc Lanctot, H Francis Song, Emilio Parisotto, Vincent Dumoulin, Subhodeep Moitra, Edward Hughes, et al. The hanabi challenge: A new frontier for ai research. *Artificial Intelligence*, 280:103216, 2020.

Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of artificial intelligence research*, 47: 253–279, 2013.

Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

Rogerio Bonatti, Dan Zhao, Francesco Bonacci, Dillon Dupont, Sara Abdali, Yinheng Li, Yadong Lu, Justin Wagle, Kazuhito Koishida, Arthur Bucker, et al. Windows agent arena: Evaluating multi-modal os agents at scale. *arXiv preprint arXiv:2409.08264*, 2024.

Noam Brown and Tuomas Sandholm. Superhuman ai for multiplayer poker. *Science*, 365(6456): 885–890, 2019.

Micah Carroll, Rohin Shah, Mark K Ho, Tom Griffiths, Sanjit Seshia, Pieter Abbeel, and Anca Dragan. On the utility of learning about humans for human-ai coordination. *Advances in neural information processing systems*, 32, 2019.

Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, Evan Mays, Giulio Starace, Kevin Liu, Leon Maksin, Tejal Patwardhan, et al. Mle-bench: Evaluating machine learning agents on machine learning engineering. *arXiv preprint arXiv:2410.07095*, 2024.

Junzhe Chen, Xuming Hu, Shuodi Liu, Shiyu Huang, Wei-Wei Tu, Zhaofeng He, and Lijie Wen. Llmarena: Assessing capabilities of large language models in dynamic multi-agent environments. *arXiv preprint arXiv:2402.16499*, 2024.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

Google DeepMined. Gemini 2.5: Our most intelligent ai model, 2025.

Jinhao Duan, Renming Zhang, James Diffenderfer, Bhavya Kailkhura, Lichao Sun, Elias Stengel-Eskin, Mohit Bansal, Tianlong Chen, and Kaidi Xu. Gtbench: Uncovering the strategic reasoning limitations of llms via game-theoretic evaluations. *arXiv preprint arXiv:2402.12348*, 2024.

Ward Edwards. The theory of decision making. *Psychological bulletin*, 51(4):380, 1954.

Jacques Ferber and Gerhard Weiss. *Multi-agent systems: an introduction to distributed artificial intelligence*, volume 1. Addison-wesley Reading, 1999.

Jakob N Foerster, Richard Y Chen, Maruan Al-Shedivat, Shimon Whiteson, Pieter Abbeel, and Igor Mordatch. Learning with opponent-learning awareness. *arXiv preprint arXiv:1709.04326*, 2017.

Drew Fudenberg and Jean Tirole. *Game theory*. MIT press, 1991.

Ghost Town Games. Overcooked, 2016.

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.

Pranav Guruprasad, Harshvardhan Sikka, Jaewoo Song, Yangyue Wang, and Paul Pu Liang. Benchmarking vision, language, & action models on robotic learning tasks. *arXiv preprint arXiv:2411.05821*, 2024.

Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. Webvoyager: Building an end-to-end web agent with large multimodal models. *arXiv preprint arXiv:2401.13919*, 2024.

Hengyuan Hu and Dorsa Sadigh. Language instructed reinforcement learning for human-ai coordination. In *International Conference on Machine Learning*, pp. 13584–13598. PMLR, 2023.

Hengyuan Hu, Adam Lerer, Alex Peysakhovich, and Jakob Foerster. "other-play" for zero-shot coordination. In *International Conference on Machine Learning*, pp. 4399–4410. PMLR, 2020.

Jen-tse Huang, Eric John Li, Man Ho Lam, Tian Liang, Wenxuan Wang, Youliang Yuan, Wenxiang Jiao, Xing Wang, Zhaopeng Tu, and Michael R Lyu. How far are we on the decision-making of llms? evaluating llms' gaming ability in multi-agent environments. *arXiv preprint arXiv:2403.11807*, 2024.

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.

Michal Kosinski. Theory of mind may have spontaneously emerged in large language models. *arXiv preprint arXiv:2302.02083*, 4:169, 2023.

Harold W Kuhn. A simplified two-person poker. *Contributions to the Theory of Games*, 1(97-103): 2, 1950.

Marc Lanctot, Edward Lockhart, Jean-Baptiste Lespiau, Vinicius Zambaldi, Satyaki Upadhyay, Julien Pérolat, Sriram Srinivasan, Finbarr Timbers, Karl Tuyls, Shayegan Omidshafiei, et al. Openspiel: A framework for reinforcement learning in games. *arXiv preprint arXiv:1908.09453*, 2019.

Joel Z Leibo, Edgar A Dueñez-Guzman, Alexander Vezhnevets, John P Agapiou, Peter Sunehag, Raphael Koster, Jayd Matyas, Charlie Beattie, Igor Mordatch, and Thore Graepel. Scalable evaluation of multi-agent reinforcement learning with melting pot. In *International conference on machine learning*, pp. 6187–6199. PMLR, 2021.

Adam Lerer and Alexander Peysakhovich. Maintaining cooperation in complex social dilemmas using deep reinforcement learning. *arXiv preprint arXiv:1707.01068*, 2017.

Kaixin Li, Yuchen Tian, Qisheng Hu, Ziyang Luo, Zhiyong Huang, and Jing Ma. Mmcode: Benchmarking multimodal large language models for code generation with visually rich programming problems. *arXiv preprint arXiv:2404.09486*, 2024a.

Wei Li, William E Bishop, Alice Li, Christopher Rawles, Folawiyo Campbell-Ajala, Divya Tyamagundlu, and Oriana Riva. On the effects of data scale on ui control agents. *Advances in Neural Information Processing Systems*, 37:92130–92154, 2024b.

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

Xiao Liu, Tianjie Zhang, Yu Gu, Iat Long Iong, Yifan Xu, Xixuan Song, Shudan Zhang, Hanyu Lai, Xinyi Liu, Hanlin Zhao, et al. Visualagentbench: Towards large multimodal models as visual foundation agents. *arXiv preprint arXiv:2408.06327*, 2024.

Richard Lorentz and Therese Horey. Programming breakthrough. In *International Conference on Computers and Games*, pp. 49–59. Springer, 2013.

Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, 30, 2017.

Christopher Lu, Timon Willi, Christian A Schroeder De Witt, and Jakob Foerster. Model-free opponent shaping. In *International Conference on Machine Learning*, pp. 14398–14411. PMLR, 2022.

R Duncan Luce and Howard Raiffa. *Games and decisions: Introduction and critical survey*. Wiley, 1957.

meta. Llama 3.2: Revolutionizing edge ai and vision with open, customizable models, 2024.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Matej Moravčík, Martin Schmid, Neil Burch, Viliam Lisỳ, Dustin Morrill, Nolan Bard, Trevor Davis, Kevin Waugh, Michael Johanson, and Michael Bowling. Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. *Science*, 356(6337):508–513, 2017.

Paul Muller, Shayegan Omidshafiei, Mark Rowland, Karl Tuyls, Julien Perolat, Siqi Liu, Daniel Hennes, Luke Marris, Marc Lanctot, Edward Hughes, et al. A generalized training approach for multiagent learning. *arXiv preprint arXiv:1909.12823*, 2019.

OpenAI. Introducing gpt-4.1 in the api, 2025a.

OpenAI. Openai o3 and o4-mini system card, 2025b.

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.

Alexander Peysakhovich and Adam Lerer. Prosocial learning agents solve generalized stag hunts better than selfish ones. *arXiv preprint arXiv:1709.02865*, 2017.

Anand S Rao, Michael P Georgeff, et al. Bdi agents: from theory to practice. In *Icmas*, volume 95, pp. 312–319, 1995.

Anatol Rapoport and Albert M Chammah. *Prisoner's dilemma: A study in conflict and cooperation*, volume 165. University of Michigan press, 1965.

Christopher Rawles, Sarah Clinckemaillie, Yifan Chang, Jonathan Waltz, Gabrielle Lau, Marybeth Fair, Alice Li, William Bishop, Wei Li, Folawiyo Campbell-Ajala, et al. Androidworld: A dynamic benchmarking environment for autonomous agents. *arXiv preprint arXiv:2405.14573*, 2024.

Jean-Jacques Rousseau. *A discourse on inequality*. Penguin, 1985.

Alexander Rutherford, Benjamin Ellis, Matteo Gallici, Jonathan Cook, Andrei Lupu, Garðar Ingvarsson Juto, Timon Willi, Ravi Hammond, Akbir Khan, Christian Schroeder de Witt, et al. Jaxmarl: Multi-agent rl environments and algorithms in jax. *Advances in Neural Information Processing Systems*, 37:50925–50951, 2024.

Abdallah Saffidine, Nicolas Jouandeau, and Tristan Cazenave. Solving breakthrough with race patterns and job-level proof number search. In *Advances in Computer Games: 13th International Conference, ACG 2011, Tilburg, The Netherlands, November 20-22, 2011, Revised Selected Papers 13*, pp. 196–207. Springer, 2012.

Mikayel Samvelyan, Tabish Rashid, Christian Schroeder De Witt, Gregory Farquhar, Nantas Nardelli, Tim GJ Rudner, Chia-Man Hung, Philip HS Torr, Jakob Foerster, and Shimon Whiteson. The starcraft multi-agent challenge. *arXiv preprint arXiv:1902.04043*, 2019.

ByteDance seed. Doubao-1.5-pro, 2025.

Claude E Shannon. Xxii. programming a computer for playing chess. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 41(314):256–275, 1950.

Lloyd S Shapley. Stochastic games. *Proceedings of the national academy of sciences*, 39(10): 1095–1100, 1953.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

Finnegan Southey, Michael P Bowling, Bryce Larson, Carmelo Piccione, Neil Burch, Darse Billings, and Chris Rayner. Bayes' bluff: Opponent modelling in poker. *arXiv preprint arXiv:1207.1411*, 2012.

DJ Strouse, Kevin McKee, Matt Botvinick, Edward Hughes, and Richard Everett. Collaborating with humans without human data. *Advances in Neural Information Processing Systems*, 34: 14502–14515, 2021.

Zhenggang Tang, Chao Yu, Boyuan Chen, Huazhe Xu, Xiaolong Wang, Fei Fang, Simon Du, Yu Wang, and Yi Wu. Discovering diverse multi-agent strategic behavior via reward randomization. *arXiv preprint arXiv:2103.04564*, 2021.

Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, et al. Gemini robotics: Bringing ai into the physical world. *arXiv preprint arXiv:2503.20020*, 2025.

Qwen team. Qvq-max: Think with evidence, 2025.

Jordan Terry, Benjamin Black, Nathaniel Grammel, Mario Jayakumar, Ananth Hari, Ryan Sullivan, Luis S Santos, Clemens Dieffendahl, Caroline Horsch, Rodrigo Perez-Vicente, et al. Pettingzoo: Gym for multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 34:15032–15043, 2021.

Gerald Tesauro. Td-gammon, a self-teaching backgammon program, achieves master-level play. *Neural computation*, 6(2):215–219, 1994.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Dan Troyka. Breakthrough. About Board Games 8x8 Game Design Competition Winner, 2000.

Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *nature*, 575(7782):350–354, 2019.

John Von Neumann and Oskar Morgenstern. Theory of games and economic behavior: 60th anniversary commemorative edition. In *Theory of games and economic behavior*. Princeton university press, 2007.

Xinyu Wang, Bohan Zhuang, and Qi Wu. Are large vision language models good game players? *arXiv preprint arXiv:2503.02358*, 2025.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Michael Wooldridge. *An introduction to multiagent systems*. John wiley & sons, 2009.

xAI. Grok-2 beta release, 2024.

Chengxing Xie, Canyu Chen, Feiran Jia, Ziyu Ye, Shiyang Lai, Kai Shu, Jindong Gu, Adel Bibi, Ziniu Hu, David Jurgens, et al. Can large language model agents simulate human trust behavior? In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024a.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh J Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, et al. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *Advances in Neural Information Processing Systems*, 37:52040–52094, 2024b.

Lin Xu, Zhiyuan Hu, Daquan Zhou, Hongyu Ren, Zhen Dong, Kurt Keutzer, See Kiong Ng, and Jiashi Feng. Magic: Investigation of large language model powered multi-agent in cognition, adaptability, rationality and collaboration. *arXiv preprint arXiv:2311.08562*, 2023.

John Yang, Carlos E Jimenez, Alex L Zhang, Kilian Lieret, Joyce Yang, Xindi Wu, Ori Press, Niklas Muennighoff, Gabriel Synnaeve, Karthik R Narasimhan, et al. Swe-bench multimodal: Do ai systems generalize to visual software domains? *arXiv preprint arXiv:2410.03859*, 2024.

Rui Yang, Hanyang Chen, Junyu Zhang, Mark Zhao, Cheng Qian, Kangrui Wang, Qineng Wang, Teja Venkat Koripella, Marziyeh Movahedi, Manling Li, et al. Embodiedbench: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents. *arXiv preprint arXiv:2502.09560*, 2025.

Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in neural information processing systems*, 35:24611–24624, 2022.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. 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*, pp. 9556–9567, 2024.

Ceyao Zhang, Kaijie Yang, Siyi Hu, Zihao Wang, Guanghe Li, Yihang Sun, Cheng Zhang, Zhaowei Zhang, Anji Liu, Song-Chun Zhu, et al. Proagent: building proactive cooperative agents with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17591–17599, 2024.

Chi Zhang, Penglin Cai, Yuhui Fu, Haoqi Yuan, and Zongqing Lu. Creative agents: Empowering agents with imagination for creative tasks. *arXiv preprint arXiv:2312.02519*, 2023.

Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. Gpt-4v (ision) is a generalist web agent, if grounded. *arXiv preprint arXiv:2401.01614*, 2024.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Yuchen Duan, Hao Tian, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

A LIMITATIONS

Evaluation metric. For decision-making evaluation, we measure the episode return in self-play and interaction with conventional agents. Incorporating results against a population of diverse opponents provides a more comprehensive assessment of the generalization and adaptability of VLM agents.

Number of agents. Many real-world scenarios involve more than two participants. Although our current environments mainly focus on two-player games, some of them naturally support more than two agents. At present, we have conducted preliminary decision-making experiments in three-player settings for both *Hanabi* and the *Coin Dilemma* in Appendix F.5. We believe that extending our benchmark to include games with even more agents is a promising direction for future work.

B MODELS CONFIGURATION DETAILS

Table 5: Model configurations used in the evaluation.

Models	Version	Evaluated	Reasoning	Multimodal	Open-Source
o3	o3-2025-04-16	✓	✓	✓	✗
gemini-2.5-pro	gemini-2.5-pro-preview-03-25	✓	✓	✓	✗
claude-3-7-sonnet	claude-3-7-sonnet-20250219	✓	✓	✓	✗
doubao-1-5-thinking-pro	doubao-1-5-thinking-pro-m-250428	✓	✓	✓	✗
ui-tars-1-5	doubao-1-5-ui-tars-250428	✓	✓	✓	✗
qvq-max	qvq-max-2025-03-25	✓	✓	✓	✗
gpt-4.1	gpt-4.1-2025-04-14	✓	✗	✓	✗
gemini-2.5 w/o thinking	gemini-2.5-flash-preview-04-17	✓	✗	✓	✗
claude-3-7 w/o thinking	claude-3-7-sonnet-20250219	✓	✗	✓	✗
grok-2-vision	grok-2-vision-1212	✓	✗	✓	✗
doubao-1-5-vision-pro	doubao-1-5-vision-pro-250328	✓	✗	✓	✗
qwen-vl-max	qwen-vl-max-2025-04-08	✓	✗	✓	✗
Llama-3.2-90B-Vision-Ins.	huggingface link	✓	✗	✓	✓
InternVL3-78B	huggingface link	✓	✗	✓	✓
Qwen2.5-VL-72B-Ins.	huggingface link	✓	✗	✓	✓
claude-4-opus	N/A	✗	✓	✓	✗
grok-4	N/A	✗	✓	✗	✗
grok-4-fast	N/A	✗	✓	✓	✗
deepseek-r1	N/A	✗	✓	✗	✓
deepseek-v3	N/A	✗	✗	✗	✓

In this work, we evaluated the state-of-the-art VLMs released before September 1, 2025. All models in our experiments are listed in Table 5. For each model, the table specifies the exact version, whether it is a reasoning model, supports multimodal inputs, and is open-source. For open-source models, the corresponding Hugging Face links are recorded in the Version column. We also list several leading models that are not included in our evaluation. Because our environments provide both visual and linguistic observations, we do not evaluate models lacking multimodal input support, such as `grok-4`, `deepseek-r1`, `deepseek-v3`. At present, `claude-4-opus` is not evaluated due to its high cost (exceeding $1000), and `grok-4-fast` is excluded because it was released on September 20, 2025, which is after our specified model release deadline.

<BACK> to Section 3 Evaluating VLMs in multi-agent environments.

C PERCEPTION EVALUATION DETAILS

In this section, we describe the construction of the perception datasets and the evaluation protocols used to assess the perception abilities of VLMs across different environments. For each game, we construct a dataset of 400 samples. Our dataset is available at `https://huggingface.co/datasets/VS-Bench/VS-Bench`.

C.1 HANABI

The samples are uniformly drawn from the trajectories of `o3` and `gemini-2.5-pro`, the most capable reasoning models that explore a more diverse range of game states.

Given the rich information present in the *Hanabi* environment, we select four key elements as evaluation targets: *life tokens*, *info tokens*, *fireworks*, and *cardinfo*. For each sample, a screenshot of the game state (e.g., Fig. 7) is provided, and the model is prompted to infer the values of the specified elements. Specifically, *fireworks* consists of five color-based stacks, while *cardinfo* represents ten cards (five per player), each requiring the model to output 11 units of information: card value (known for the opponent's hand, but "?" for the player's own hand), and the color/number knowledge that each player currently has about the cards. In total, each sample requires predicting 117 units of information: 1 for *life tokens*, 1 for *info tokens*, 5 for *fireworks*, and 110 (11×10) for *cardinfo*. Each unit is equally weighted, and the final accuracy is normalized to the range $[0, 1]$.

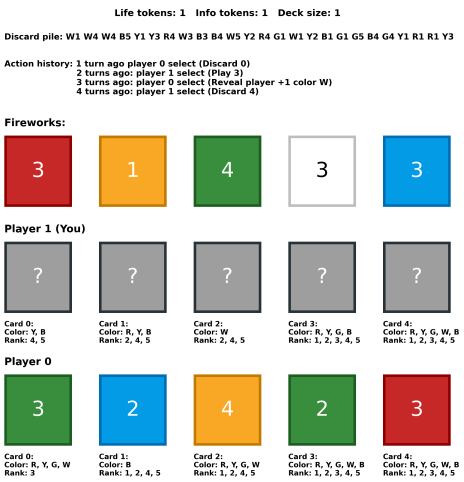


Figure 7: *Hanabi*.

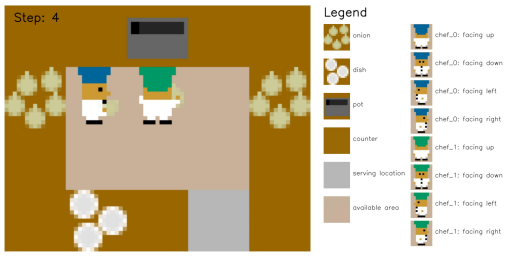


Figure 8: *Overcooked*.

C.2 Overcooked

This dataset is derived from the human experiment data provided by Overcooked-AI Carroll et al. (2019), which contains gameplay trajectories recorded from multiple human participants.

For the *Overcooked* environment, each state is represented as a 4×5 grid. For each sample, a screenshot corresponds to this 4-row by 5-column layout (e.g., Fig. 8), and the model is prompted to infer the content of each grid cell, producing a 4×5 matrix as output.

Each grid cell may contain objects such as tables, pots, onions, dishes, service desks, or empty areas. To simplify model predictions, we map each possible object to a symbolic representation (e.g., `P` stands for pot). The model's score is computed as the number of correctly predicted grid cells out of 20, normalized to the range $[0, 1]$.

C.3 Knights Archers Zombies

The samples are uniformly drawn from trajectories generated by a `random` agent. This choice is motivated by the fact that the game may involve a large number of zombies. To simplify perception for VLMs, we limit the maximum number of zombies to five. Under this constraint, we observe that the `random` agent explores a broader distribution of game states compared to other VLMs, thereby providing a more comprehensive benchmark for evaluating perception ability.

The original *KAZ* environment is continuous (e.g., Fig. 9). To facilitate visual recognition by VLMs, we implement an additional wrapper that overlays coordinate axes and scales on the environment. For each sample, a screenshot of the game state with coordinate axes is provided (e.g., Fig. 10), and the model is prompted to infer the following elements: the coordinates of the archer and the knight, the number of zombies present (up to five), and the positions of all detected zombies. The four elements are weighted as follows: 0.2, 0.2, 0.2, and 0.4, respectively.

For zombie count, exact matching is required to receive full credit. For coordinate predictions, full credit is awarded if the predicted position lies within a predefined radius centered on the ground-

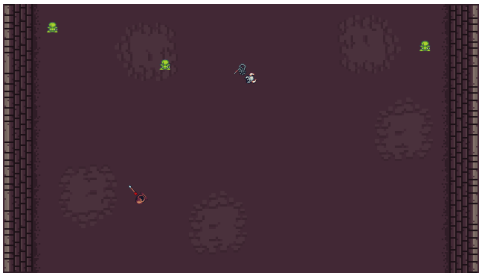


Figure 9: Origin *KAZ*.

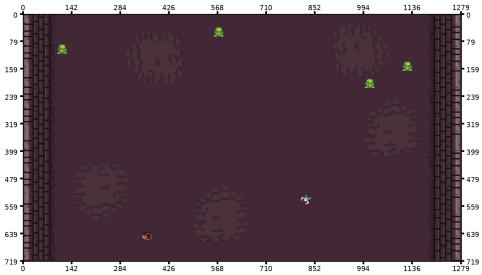


Figure 10: *KAZ* with coordinate axes.

truth location. Beyond this radius, the score decays proportionally to the distance from the correct position. In our setting, the radius is fixed at 100.

C.4 BREAKTHROUGH

The samples are uniformly drawn from trajectories generated by a `random` agent. This choice is motivated by the observation that it can explore board states that other VLMs are unlikely to reach, thereby providing a more comprehensive benchmark for evaluating perception ability.

In the *Breakthrough* environment, each state is represented as an 8×8 grid. Accordingly, a screenshot of the game state (e.g., Fig. 11) is provided for each sample, and the model is prompted to infer the content of each grid cell, ultimately producing an 8×8 matrix as output.

Each grid cell may contain one of three possible objects: `b` for a black piece, `w` for a white piece, and `.` for an empty cell. The model's score is computed as the number of correctly predicted grid cells out of 64, normalized to the range $[0, 1]$.

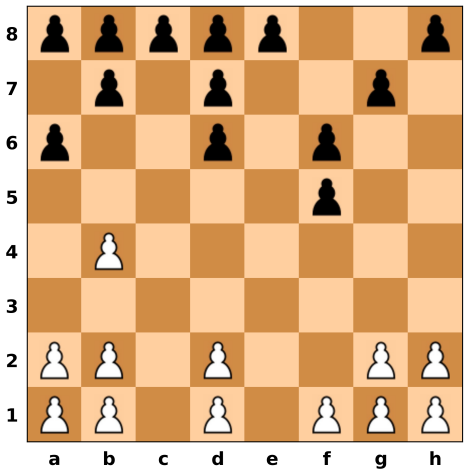


Figure 11: *Breakthrough*.

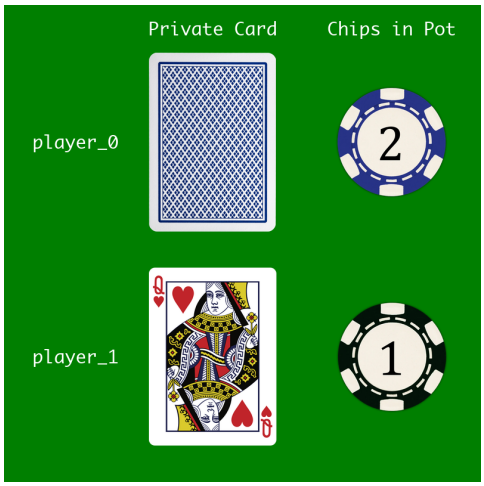


Figure 12: *Kuhn Poker*.

C.5 KUHN POKER

The samples are uniformly drawn from trajectories generated by `random` agent, since the overall state space is relatively small and a random agent is able to traverse it more effectively.

In *Kuhn poker*, the environment contains only four variables: the card values of both players (with the opponent's card being hidden and thus marked as `unknown`), and the chips in pot of both players (e.g., Fig. 12). Accordingly, the model is prompted to infer these four elements, each assigned an equal weight, and the final score is normalized to the range $[0, 1]$.

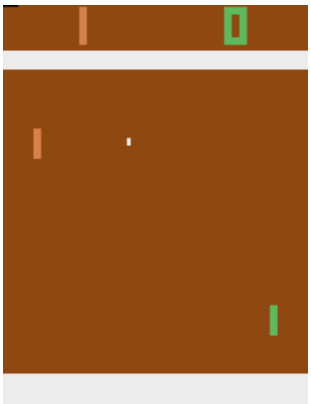


Figure 13: Origin *Atari Pong*.

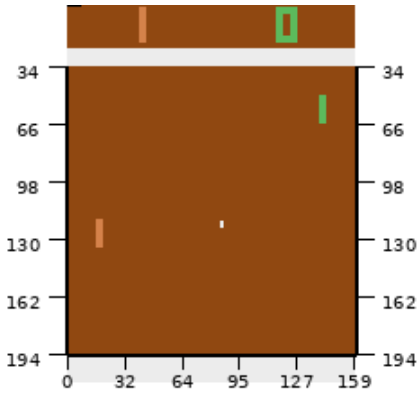


Figure 14: *Atari Pong* with coordinate axes.

C.6 ATARI PONG

The samples are uniformly drawn from trajectories generated by `o4-mini` and `doubao-1-5-thinking-pro`, both of which exhibit strong performance in the game.

The original *Atari Pong* environment is continuous (e.g., Fig. 13). To facilitate visual recognition by VLMs, we implement an additional wrapper that overlays coordinate axes and scales on the environment. For each sample, a screenshot of the game state with coordinate axes is provided (e.g., Fig. 14), and the model is prompted to infer the following elements: the positions of the two paddles, the position of the ball, and the current score, with weights of $4 : 4 : 2$, respectively.

Since the paddles move only vertically, the model is required to predict their y-coordinates only. For the ball, the model is required to predict both x- and y-coordinates. A prediction receives full credit if it lies within a predefined radius of the ground truth; beyond this radius, the score decays with distance. The radius is fixed at 10 in our setting.

C.7 MPE

The samples are uniformly drawn from trajectories generated by `random` agents, which provide a broader distribution of game states compared to other VLMs.

In the *MPE* environment, there are only two variables representing the coordinates of the two players and a target fixed at the origin (e.g., Fig. 15). Accordingly, the model is prompted to infer the x- and y-coordinates of both players. For coordinate predictions, full credit is awarded if the predicted position lies within a predefined radius centered on the ground-truth location. Beyond this radius, the score decays proportionally to the distance from the correct position. In our setting, the radius is set to half of the unit spacing of the coordinate axis.

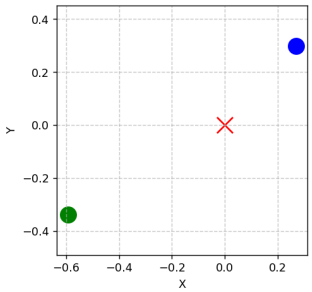


Figure 15: *MPE*.

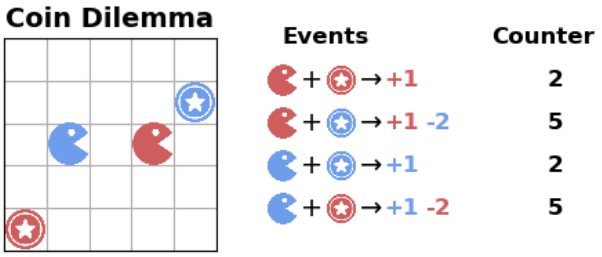


Figure 16: *Coin Dilemma*.

C.8 COIN DILEMMA

The sampling strategy and dataset construction follow the same procedure as in the strategic reasoning setting (Appendix D.8).

In the *Coin Dilemma* environment, each state is represented as a 5×5 grid. Accordingly, a screenshot of the game state (e.g., Fig. 16) is provided for each sample, and the model is prompted to infer the content of each grid cell, ultimately producing a 5×5 matrix as output.

Each grid cell may contain the following objects: an empty cell, a red coin, a blue coin, a red player, or a blue player. The model's score is computed as the number of correctly predicted grid cells out of 25, normalized to the range $[0, 1]$.

C.9 MONSTER HUNT

The sampling strategy and dataset construction follow the same procedure as in the strategic reasoning setting (Appendix D.9).

In the *Monster Hunt* environment, each state is represented as a 5×5 grid. Accordingly, a screenshot of the game state (e.g., Fig. 17) is provided for each sample, and the model is prompted to infer the content of each grid cell, ultimately producing a 5×5 matrix as output.

Each grid cell may contain the following objects: an empty cell, an apple, a red player, a blue player, or a monster. The model's score is computed as the number of correctly predicted grid cells out of 25, normalized to the range $[0, 1]$.

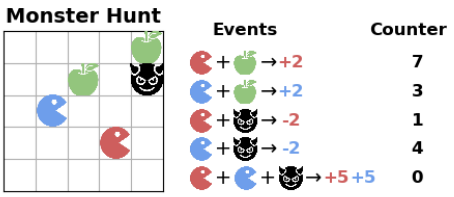


Figure 17: *Monster Hunt*.

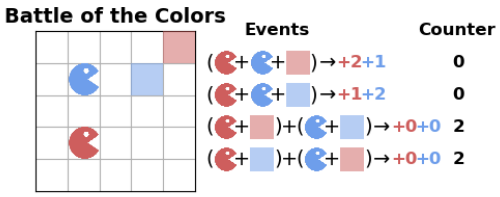


Figure 18: *Battle of the Colors*.

C.10 BATTLE OF THE COLORS

The sampling strategy and dataset construction follow the same procedure as in the strategic reasoning setting (Appendix D.10).

In the *Battle of the Colors* environment, each state is represented as a 5×5 grid. Accordingly, a screenshot of the game state (e.g., Fig. 18) is provided for each sample, and the model is prompted to infer the content of each grid cell, ultimately producing a 5×5 matrix as output.

Each grid cell may contain the following objects: an empty cell, a red block, a blue block, a red player or a blue player. The model's score is computed as the number of correctly predicted grid cells out of 25, normalized to the range $[0, 1]$.

<BACK> to Section 3.1 Perception.

D STRATEGIC REASONING EVALUATION DETAILS

In this section, we describe the dataset generation procedure for each environment to evaluate the strategic reasoning of VLMs. Each game dataset contains 400 samples and our datasets is available at `https://huggingface.co/datasets/VS-Bench/VS-Bench`.

D.1 HANABI

The dataset is generated in three steps. First, to include data with different gameplay levels, 90% of the samples are obtained through mutual prediction between two reasoning models, while the

remaining 10% are generated by a chat model predicting a reasoning model's move. We chose `doubao-1-5-thinking-pro` and `doubao-1-5-vision-pro`, both of which demonstrated top-tier decision-making performance, to represent the reasoning and chat models, respectively. Second, to balance the ratio of different actions to approximate human gameplay, we analyze the gameplay data of leading VLM agents and set the ratio of <Play>:<Discard>:<Reveal> to $2:3:4$ in our dataset. Third, we balance the agent order and step index in the dataset so that the first and second agents each account for 50% samples and the step index is uniformly distributed across the entire game sequence.

D.2 OVERCOOKED

This dataset is derived from the human experiment data provided by Overcooked-AI Carroll et al. (2019), which comprises game trajectories recorded from multiple human participants. We focus on the trial-train subset of these data, with instances of invalid actions filtered out from the trajectories. We then use random sampling on these filtered trajectories to ensure comprehensive coverage of possible game states. Each sample comprises a sequence of four consecutive game frames. Additionally, we apply constraints to simulate realistic game scenarios and control the distribution of target actions. Specifically, the proportion of <STAY> action among the possible actions is limited to 10%. The dataset is balanced for two chefs, each accounting for 50% samples of the dataset.

D.3 KNIGHTS ARCHERS ZOMBIES

The dataset is uniformly sampled from the logged trajectories of the two best-performing models in the decision-making process: `o4-mini` and `doubao-1-5-thinking-pro`. The next actions taken by these VLM agents are used as ground truth labels. Since the environment involves two types of agents, the knight and the archer, we ensure an equal split of samples, with 50% corresponding to each agent type.

D.4 BREAKTHROUGH

All samples are generated by a minimax algorithm with alpha–beta pruning, a widely adopted baseline in *Breakthrough* research Lorentz & Horey (2013); Saffidine et al. (2012). Since minimax search does not always reach terminal positions to determine win–loss outcomes, we implement a state evaluation function: upon reaching a fixed search depth, we compute the difference between the maximum effective forward advancement of our deepest piece and that of the opponent's deepest piece, then normalize this difference to obtain a reward for the state. We configure minimax agents with maximum search depths for the first and second players as $(3, 4)$, $(3, 5)$, $(4, 5)$, $(4, 6)$, $(4, 4)$, and $(5, 5)$, respectively, and sample step indices uniformly to ensure coverage of diverse game states.

D.5 KUHN POKER

Since *Kuhn Poker* admits a continuum of mixed Nash equilibria parameterized by a single probability $\alpha \in [0, 1/3]$, representing the likelihood of betting when holding a Jack Kuhn (1950), we evaluate a discrete subset of strategies. Specifically, we consider all pairwise matchups among three representative values of $\alpha = 0, 1/6, 1/3$, yielding nine distinct strategy combinations. For each combination, we simulate 600 head-to-head games and uniformly sample 400 game states in total to construct the final dataset.

D.6 ATARI PONG

We uniformly sample 400 state transitions from the logged trajectories of two strong reasoning models in the decision-making process, namely `o4-mini` and `doubao-1-5-thinking-pro`. The subsequent actions taken by these VLM agents are treated as ground-truth labels. To construct the evaluation setting, we modify the prompts such that the VLMs are required to control the left paddle (originally the built-in bot's paddle) and predict the corresponding actions.

D.7 MPE

The MPE environment consists of two agents navigating a two-dimensional plane. We uniformly sample 400 state transitions from the logged trajectories of three strong reasoning models in the decision-making process, namely `o4-mini`, `claude-3.7-sonnet`, and `doubao-1-5-thinking-pro`. The subsequent actions taken by these VLM agents are treated as ground-truth labels. It is important to emphasize that we only log trajectories in which the VLMs assume the blocker role; consequently, the prediction task is restricted to the blocker's actions.

D.8 COIN DILEMMA

We consider two types of heuristic strategies for playing *Coin Dilemma* and generate the dataset by simulating game play with these strategies:

1. **Common welfare**: the agent only collects the coin of its own color.
2. **Self interest**: the agent always collects the closest coin, regardless of the color.

Concretely, we collect samples from six settings, resulting in a dataset of 400 samples:

1. **Common welfare + common welfare**: collect 100 samples.
2. **Self interest + self interest**: collect 100 samples.
3. **Common welfare + self interest**: collect 50 samples.
4. **Self interest + common welfare**: collect 50 samples.
5. **Random + self interest**: collect 50 samples.
6. **Self interest + random**: collect 50 samples.

For *Coin Dilemma*, different actions can lead to the same transition in the environment. Therefore, the actions with the same outcome as the ground truth action are all considered correct. For example, if a player is at the top-left corner of the grid map, then actions <UP> and <LEFT> are both considered correct with ground truth <STAY> as they all result in no movement of the player.

D.9 MONSTER HUNT

We consider four types of heuristic strategies for playing *Monster Hunt* and generate the dataset by simulating game play with these strategies:

1. **Common welfare 1**: the agent always moves towards the monster.
2. **Common welfare 2**: the agent first moves to the middle block of the grid map and stays there to wait for the other agent and the monster.
3. **Common welfare 3**: the agent first moves to a corner of the grid map and stays there to wait for the other agent and the monster.
4. **Self interest**: the agent always moves towards the closest apple.

Concretely, we collect samples from six settings, resulting in a dataset of 400 samples:

1. **Common welfare 1 + common welfare 1**: collect 80 samples.
2. **Common welfare 2 + common welfare 2**: collect 80 samples.
3. **Common welfare 3 + common welfare 3**: collect 80 samples.
4. **Self interest + self interest**: collect 80 samples.
5. **Random + self interest**: collect 40 samples.
6. **Self interest + random**: collect 40 samples.

For *Monster Hunt*, different actions with the same outcome as the ground truth action are all considered correct, as in *Coin Dilemma*.

D.10 BATTLE OF THE COLORS

We consider four types of heuristic strategies for playing *Battle of the Colors* and generate the dataset by simulating game play with these strategies:

1. **Common welfare**: the agent always moves to the closest color block (to both players) and stays there to wait for the other player.
2. **Self interest**: the agent always moves to the block of its own color.
3. **Biased red**: the agent always moves to the red block.
4. **Biased blue**: the agent always moves to the blue block.

Concretely, we collect samples from six settings, resulting in a dataset of 400 samples:

1. **Common welfare + common welfare**: collect 100 samples.
2. **Self interest + self interest**: collect 100 samples.
3. **Common welfare + self interest**: collect 50 samples.
4. **Self interest + common welfare**: collect 50 samples.
5. **Biased red + biased red**: collect 50 samples.
6. **Biased blue + biased blue**: collect 50 samples.

For *Battle of the Colors*, different actions with the same outcome as the ground truth action are all considered correct, as in *Coin Dilemma*.

<BACK> to Section 3.2 Strategic reasoning.

E DECISION-MAKING EVALUATION DETAILS

In this section, we describe the methodology for evaluating the decision-making abilities of VLMs by assigning appropriate rewards after each step of interaction.

E.1 HANABI

In *Hanabi*, we consider two kinds of return.

Standard return. This is the standard return in existing *Hanabi* environments Bard et al. (2020), and we use it in the main text results. If all life tokens are consumed before the fireworks are completed, the agents get a reward of 0. If all fireworks stacks are built successfully, the agents get a return of 25. If the above two early terminal conditions are not reached and the deck is exhausted, the game continues for one additional round, and the return is the sum of the ranks of the fireworks.

Firework return. Since many VLM agents often consume all the life tokens and result in a zero return, we also consider another return that relaxes the "zero-out" penalty upon losing all life tokens. More specifically, if all life tokens are consumed before the fireworks are completed, the return is the sum of the ranks of the firework piles. This return measures the progress of the fireworks and is reported in the raw decision-making results without normalization in Table 6.

For each model, we perform 10 self-play games and report the average standard return and the firework return. These results are then normalized and compared with a random baseline and an optimal policy as reported in Multi-Agent PPO (MAPPO) Yu et al. (2022).

E.2 OVERCOOKED

In *Overcooked*, each episode is limited to 50 steps. Within these steps, two chefs cooperatively cook soup and deliver the cooked soup to the service desk. The two agents share a common return, which includes three process rewards and a delivery reward.

Process rewards. To measure the progress of making a soup, the agents get 2 reward for each of the three events: (1) An agent successfully adds an onion to a cooking pot; (2) An agent picks up a dish

when a pot contains onions or cooking is in progress; (3) An agent successfully plates a finished soup using a dish. Completing a three-onion soup leads to a total of 10 process rewards.

Delivery reward. To align with the game's goal of successfully delivering specified dishes, the agents get 10 reward upon successful delivery to the service desk.

For the 3-onion soup recipe, successfully completing and delivering one soup yields a total return of 20, consisting of a process reward of 10 and a final reward of 10. We evaluate each VLM over 10 episodes of self-play, in which both agents employ the same model. The accumulated reward across an episode is reported as the episode return, and scores are further normalized with respect to the returns achieved by random and optimal agents. Notably, the optimal agents are able to complete two deliveries within a single episode.

E.3 KNIGHTS ARCHERS ZOMBIES

In *Knights Archers Zombies* (KAZ), the return is defined as the total number of zombies eliminated by the end of the game. The game terminates under two possible conditions: (i) reaching the pre-defined maximum horizon of 200 steps, or (ii) when either of the two players is contacted by a zombie and dies. Although the environment inherently supports multi-agent settings, we adopt a two-agent configuration in the decision-making context considered in this work, with one archer and one knight. Moreover, a new zombie is spawned every 10 steps, implying that optimal agents can achieve up to 20 points in a single episode.

E.4 BREAKTHROUGH

In *Breakthrough*, the return is determined by the final outcome: the winning agent gets a reward of $+1$, the losing agent gets a reward of -1, and draws are not possible. We selected a moderately strong MCTS agent as our baseline, configured with an exploration constant $c = 2.0$, a maximum of 100 simulations per move, and a rollout count of 10. Each model is evaluated for 20 games against this MCTS agent, with 10 games as the first agent and 10 games as the second agent. For the optimal policy, we use a minimax agent with alpha–beta pruning and a maximum search depth of 5, using a state evaluation function as described in Appendix D.4. Although minimax is not guaranteed to be optimal for *Breakthrough*, it achieved a perfect win rate against the MCTS agent in our experiment, making it a reasonable choice as the optimal policy in this study.

E.5 KUHN POKER

In *Kuhn Poker*, the return of each agent is the net chips won or lost at the end of the game. We consider a Nash equilibrium (NE) strategy as described in Appendix D.5 with $\alpha = 0$. Each model is evaluated for ten runs of 120 games against this NE agent, with 60 games as the first agent and 60 games as the second agent. We then calculate the mean and standard deviation of the ten runs and normalize the result. The optimal agent in *Kuhn poker* is the agent with the NE strategy.

E.6 MPE

In the *MPE* environment, there are two types of players: a *charger* and a *blocker*. Since it is a competitive game, we designate the VLMs as the blocker, competing against a built-in agent that plays as the charger. The built-in agent is rule-based, with a strategy that always moves directly toward the target.

The reward function of the blocker consists of two components:

Push Charger: defined as the distance between the charger and the target.

Near Target: defined as the negative distance between the blocker and the target.

In summary, the blocker is incentivized to stay as close to the target as possible while simultaneously keeping the charger as far away from the target as possible. In our setting, we adopt a heuristic value of 0 as the optimal baseline.

E.7 ATARI PONG

In *Atari Pong*, we adopt frame stacking of 4 frames to pass dynamic information to the VLM agent. We also employ a sticky action probability of 0.25 and perform a random number (between 1 and 30) of <STAY> steps at the beginning of an episode to achieve randomness. These settings have been common practice in related works, such as DQN Mnih et al. (2013). We also consider two kinds of reward.

Score reward. The agent scores one point when the ball passes the opponent's paddle and gets a score reward of $+1$. The game ends when an agent scores three points.

Step reward. As many VLMs fail to score even one point against the built-in bot, the score reward alone becomes too sparse to distinguish the performance of different models. Therefore, we design a continuous reward proportional to the number of steps that VLMs survived against the built-in bot.

We evaluate each VLM for 10 episodes against the built-in bot, and the overall return is the weighted sum of the normalized score reward with a 0.9 weight and the normalized step reward with a 0.1 weight. The optimal agent is an RL agent that scores three points against the built-in bot.

E.8 COIN DILEMMA

In *Coin Dilemma*, the agents receive rewards on different game events:

1. **Red agent collects a red coin**: red agent gets $+1$ reward.
2. **Red agent collects a blue coin**: red agent gets $+1$ reward, blue agent get -2 reward.
3. **Blue agent collects a blue coin**: blue agent gets $+1$ reward.
4. **Blue agent collects a red coin**: blue agent gets $+1$ reward, red agent get -2 reward.

We evaluate the return of each VLM for 10 episodes of self-play, where the red and blue agents use the same model. We further normalize these returns with respect to the returns of the random agents and the optimal agents that always move towards the coin of their own color.

E.9 MONSTER HUNT

In *Monster Hunt*, the agents receive rewards on different game events:

1. **Red agent eats an apple**: red agent gets $+2$ reward.
2. **Blue agent eats an apple**: blue agent gets $+2$ reward.
3. **Red agent encounters the monster alone**: red agent gets -2 reward.
4. **Blue agent encounters the monster alone**: blue agent gets -2 reward.
5. **Both agents defeat the monster together**: both agents get $+5$ reward.

We evaluate the return of each VLM for 10 episodes of self-play, where the red and blue agents use the same model. We further normalize these returns with respect to the returns of the random agents and the optimal agents that always move to the middle block and stay there to wait for the monster.

E.10 BATTLE OF THE COLORS

In *Battle of the Colors*, the agents receive rewards on different game events:

1. **Both players on the red block**: red agent gets $+2$ reward, blue agent gets $+1$ reward.
2. **Both players on the blue block**: blue agent gets $+2$ reward, red agent gets $+1$ reward.
3. **Two players on different blocks**: both agents get 0 reward.

We evaluate the return of each VLM for 10 episodes of self-play, where the red and blue agents use the same model. We further normalize these returns with respect to the returns of the random agents and the optimal agents that always move to the closest block to the two players.

Table 6: Raw decision-making results without normalization. The *Hanabi* standard[1] and firework[2] returns are described in Appendix E.1, while the *Pong* score[3] and step[4] rewards are described in Appendix E.7.

Models	Cooperative						Competitive		Mixed-Motive			
	Hanabi		*Overcooked*	*KAZ*	*Board*	*Poker*	*Pong*		*MPE*	*Dilemma*	*Hunt*	*Battle*
	standard[1]	firework[2]					score[3]	step[4]				
Optimal	24.0	24.0	40.0	20.0	1.0	0.0	3.0	398.0	0.0	27.9	176.2	59.7
o3	$13.4_{\pm 5.0}$	$13.6_{\pm 4.5}$	$6.4_{\pm 1.2}$	$6.6_{\pm 3.1}$	$0.6_{\pm 0.8}$	$-0.0_{\pm 0.0}$	$0.2_{\pm 0.4}$	$219.4_{\pm 113.4}$	$-6.5_{\pm 9.3}$	$-0.3_{\pm 4.9}$	$27.0_{\pm 6.3}$	$10.2_{\pm 3.1}$
o4-mini	$10.3_{\pm 7.3}$	$13.3_{\pm 2.9}$	$7.0_{\pm 2.7}$	$3.7_{\pm 2.5}$	$-0.4_{\pm 0.9}$	$-0.0_{\pm 0.0}$	$0.3_{\pm 0.5}$	$205.2_{\pm 91.0}$	$-6.6_{\pm 8.3}$	$-1.5_{\pm 6.0}$	$28.6_{\pm 16.1}$	$2.4_{\pm 3.2}$
gemini-2.5-pro	$7.9_{\pm 8.5}$	$11.6_{\pm 5.2}$	$7.0_{\pm 2.2}$	$2.6_{\pm 1.6}$	$0.1_{\pm 1.0}$	$-0.1_{\pm 0.1}$	$0.1_{\pm 0.3}$	$246.7_{\pm 65.2}$	$-6.1_{\pm 8.0}$	$-2.9_{\pm 5.5}$	$22.0_{\pm 11.7}$	$20.4_{\pm 3.8}$
gemini-2.5-flash	$6.5_{\pm 8.6}$	$10.7_{\pm 5.3}$	$3.6_{\pm 2.1}$	$3.7_{\pm 1.7}$	$-0.6_{\pm 0.5}$	$-0.1_{\pm 0.1}$	$0.0_{\pm 0.0}$	$194.4_{\pm 53.2}$	$-7.7_{\pm 8.9}$	$2.6_{\pm 7.2}$	$31.2_{\pm 11.4}$	$19.8_{\pm 5.1}$
claude-3.7-sonnet	$1.6_{\pm 5.1}$	$9.7_{\pm 3.9}$	$4.2_{\pm 1.4}$	$2.0_{\pm 1.4}$	$-0.6_{\pm 0.8}$	$-0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$133.7_{\pm 27.8}$	$-5.9_{\pm 8.5}$	$1.1_{\pm 4.3}$	$18.8_{\pm 6.9}$	$1.8_{\pm 2.8}$
ui-tars-1-5	$0.0_{\pm 0.0}$	$4.9_{\pm 1.7}$	$1.0_{\pm 1.3}$	$0.1_{\pm 0.3}$	$-0.9_{\pm 0.3}$	$-0.1_{\pm 0.0}$	$0.0_{\pm 0.0}$	$121.4_{\pm 8.4}$	$-11.2_{\pm 12.9}$	$9.1_{\pm 4.0}$	$28.6_{\pm 6.5}$	$8.4_{\pm 1.7}$
doubao-1.5-thinking-pro	$13.6_{\pm 5.5}$	$14.1_{\pm 4.0}$	$4.2_{\pm 1.9}$	$3.7_{\pm 1.7}$	$-0.8_{\pm 0.4}$	$-0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$230.5_{\pm 72.6}$	$-6.2_{\pm 10.3}$	$0.0_{\pm 0.9}$	$13.6_{\pm 22.3}$	$2.7_{\pm 2.8}$
qvq-max	$0.0_{\pm 0.0}$	$4.9_{\pm 2.9}$	$1.0_{\pm 1.3}$	$0.7_{\pm 0.9}$	$-0.9_{\pm 0.3}$	$-0.1_{\pm 0.0}$	$0.0_{\pm 0.0}$	$158.2_{\pm 46.9}$	$-6.1_{\pm 8.3}$	$-0.2_{\pm 0.6}$	$-18.8_{\pm 8.8}$	$0.0_{\pm 0.0}$
gemini-2.5 w/o thinking	$0.0_{\pm 0.0}$	$3.8_{\pm 1.6}$	$1.0_{\pm 1.6}$	$1.0_{\pm 0.8}$	$-1.0_{\pm 0.0}$	$-0.1_{\pm 0.0}$	$0.0_{\pm 0.0}$	$175.9_{\pm 41.5}$	$-10.0_{\pm 7.7}$	$-0.4_{\pm 1.2}$	$-18.8_{\pm 17.4}$	$1.8_{\pm 2.0}$
claude-3.7 w/o thinking	$0.0_{\pm 0.0}$	$2.9_{\pm 0.9}$	$1.0_{\pm 1.6}$	$1.7_{\pm 1.7}$	$-0.9_{\pm 0.3}$	$-0.1_{\pm 0.1}$	$0.0_{\pm 0.0}$	$121.4_{\pm 8.4}$	$-14.8_{\pm 14.0}$	$0.2_{\pm 2.6}$	$-19.8_{\pm 16.0}$	$0.9_{\pm 1.4}$
gpt-4.1	$0.0_{\pm 0.0}$	$3.6_{\pm 1.4}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$-1.0_{\pm 0.0}$	$-0.1_{\pm 0.0}$	$0.0_{\pm 0.0}$	$151.8_{\pm 41.5}$	$-8.0_{\pm 6.6}$	$4.8_{\pm 1.9}$	$1.8_{\pm 10.9}$	$0.6_{\pm 1.2}$
qwen-vl-max	$0.3_{\pm 0.5}$	$0.3_{\pm 0.5}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$-1.0_{\pm 0.0}$	$-0.1_{\pm 0.1}$	$0.0_{\pm 0.0}$	$139.7_{\pm 29.3}$	$-12.5_{\pm 12.0}$	$-0.3_{\pm 0.8}$	$5.8_{\pm 39.6}$	$0.0_{\pm 0.0}$
doubao-1.5-vision-pro	$0.0_{\pm 0.0}$	$4.6_{\pm 1.1}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$-1.0_{\pm 0.0}$	$-0.1_{\pm 0.1}$	$0.0_{\pm 0.0}$	$121.4_{\pm 8.4}$	$-7.7_{\pm 9.3}$	$-0.8_{\pm 1.5}$	$-4.8_{\pm 16.1}$	$0.0_{\pm 0.0}$
grok-2-vision	$0.0_{\pm 0.0}$	$1.6_{\pm 1.0}$	$0.8_{\pm 1.3}$	$1.0_{\pm 1.4}$	$-1.0_{\pm 0.0}$	$-0.1_{\pm 0.1}$	$0.2_{\pm 0.4}$	$152.6_{\pm 45.9}$	$-31.6_{\pm 24.7}$	$0.1_{\pm 2.0}$	$-21.0_{\pm 11.4}$	$0.6_{\pm 1.2}$
Qwen2.5-VL-72B-Ins.	$0.2_{\pm 0.4}$	$0.2_{\pm 0.4}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$-1.0_{\pm 0.0}$	$-0.1_{\pm 0.1}$	$0.0_{\pm 0.0}$	$123.8_{\pm 4.6}$	$-11.1_{\pm 10.5}$	$-0.2_{\pm 0.8}$	$18.2_{\pm 50.4}$	$0.0_{\pm 0.0}$
InternVL3-78B	$0.0_{\pm 0.0}$	$2.4_{\pm 1.0}$	$0.2_{\pm 0.6}$	$0.3_{\pm 0.5}$	$-1.0_{\pm 0.0}$	$-0.1_{\pm 0.1}$	$0.0_{\pm 0.0}$	$121.4_{\pm 8.4}$	$-14.8_{\pm 14.8}$	$1.2_{\pm 2.0}$	$-20.6_{\pm 10.8}$	$1.2_{\pm 1.5}$
Llama-3.2-90B-Vision-Ins.	$0.0_{\pm 0.0}$	$1.2_{\pm 1.6}$	$0.8_{\pm 1.3}$	$1.0_{\pm 0.8}$	$-1.0_{\pm 0.0}$	$-0.1_{\pm 0.1}$	$0.0_{\pm 0.0}$	$121.4_{\pm 8.4}$	$-24.1_{\pm 17.6}$	$-0.1_{\pm 0.9}$	$-13.2_{\pm 9.6}$	$0.9_{\pm 1.4}$
Random	0.0	1.2	0.2	1.0	−1.0	−0.1	0.0	147.2	−16.3	−0.2	−20.2	0.3

Table 7: Strategic reasoning results on multimodal input and CoT prompting.

Model	*Hanabi*			*Breakthrough*			*Monster Hunt*		
	Text-Only	Multimodal	CoT	Text-Only	Multimodal	CoT	Text-Only	Multimodal	CoT
Optimal	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
o3	58.2	61.2	N/A	31.0	29.0	N/A	50.8	57.8	N/A
o4-mini	53.8	58.2	N/A	27.5	26.8	N/A	47.8	43.8	N/A
gemini-2.5-pro	54.2	51.2	N/A	28.2	26.8	N/A	55.0	35.8	N/A
gemini-2.5-flash	42.0	37.0	N/A	23.5	23.2	N/A	49.0	50.2	N/A
claude-3-7-sonnet	45.0	39.0	N/A	25.0	24.2	N/A	56.0	42.5	N/A
doubao-1-5-thinking-pro	34.5	32.8	N/A	23.5	19.8	N/A	57.2	45.2	N/A
ui-tars-1-5	24.2	25.5	N/A	27.8	23.3	N/A	34.0	41.5	N/A
qvq-max	41.0	32.2	N/A	27.5	21.8	N/A	48.2	21.5	N/A
gemini-2.5 w/o thinking	24.5	21.5	24.0	20.5	14.8	21.5	40.2	30.0	45.0
claude-3-7 w/o thinking	19.2	9.8	32.8	19.2	18.0	19.0	37.0	26.0	43.8
gpt-4.1	40.0	23.0	49.8	20.5	22.5	27.5	41.0	36.8	34.5
qwen-vl-max	17.0	26.5	20.0	19.0	19.5	17.2	31.8	23.5	35.5
doubao-1-5-vision-pro	19.5	15.0	25.2	17.2	15.8	16.8	33.2	36.0	39.2
grok-2-vision	23.8	12.8	22.5	14.0	10.8	18.2	42.5	31.5	33.2
Qwen2.5-VL-72B-Ins.	18.5	26.8	22.2	19.2	23.8	16.5	32.2	27.2	37.5
InternVL3-78B	26.8	25.2	20.5	17.5	14.0	16.0	34.2	30.0	37.8
Llama-3.2-90B-Vision-Ins.	26.8	20.0	14.8	6.5	11.8	14.0	36.8	26.2	27.8
Random	8.8	8.8	8.8	4.3	4.3	4.3	29.3	29.3	29.3

Table 8: Decision-making results on multimodal input and CoT prompting.

Model	*Hanabi*			*Breakthrough*			*Monster Hunt*		
	Text-Only	Multimodal	CoT	Text-Only	Multimodal	CoT	Text-Only	Multimodal	CoT
Optimal	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
o3	$62.9_{\pm 7.1}$	$55.8_{\pm 20.9}$	N/A	$95.0_{\pm 15.0}$	$80.0_{\pm 40.0}$	N/A	$45.3_{\pm 7.7}$	$24.0_{\pm 3.2}$	N/A
o4-mini	$37.1_{\pm 26.1}$	$42.9_{\pm 30.5}$	N/A	$30.0_{\pm 47.0}$	$30.0_{\pm 47.0}$	N/A	$23.2_{\pm 4.2}$	$24.9_{\pm 8.2}$	N/A
gemini-2.5-pro	$22.5_{\pm 36.7}$	$32.9_{\pm 35.5}$	N/A	$55.0_{\pm 49.5}$	$55.0_{\pm 49.5}$	N/A	$50.4_{\pm 9.0}$	$21.5_{\pm 6.0}$	N/A
gemini-2.5-flash	$40.8_{\pm 21.9}$	$27.1_{\pm 36.0}$	N/A	$30.0_{\pm 42.3}$	$20.0_{\pm 25.7}$	N/A	$15.63_{\pm 4.6}$	$26.2_{\pm 5.8}$	N/A
claude-3-7-sonnet	$33.8_{\pm 35.8}$	$6.7_{\pm 21.1}$	N/A	$45.0_{\pm 50.0}$	$20.0_{\pm 39.8}$	N/A	$27.3_{\pm 4.6}$	$19.9_{\pm 3.5}$	N/A
doubao-1-5-thinking-pro	$37.5_{\pm 32.9}$	$56.7_{\pm 22.8}$	N/A	$15.0_{\pm 37.0}$	$10.0_{\pm 21.0}$	N/A	$27.6_{\pm 6.1}$	$17.2_{\pm 11.3}$	N/A
ui-tars-1-5	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	N/A	$20.0_{\pm 40.0}$	$30.0_{\pm 47.0}$	N/A	$12.6_{\pm 5.1}$	$24.8_{\pm 3.3}$	N/A
qvq-max	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	N/A	$5.0_{\pm 15.8}$	$5.0_{\pm 15.8}$	N/A	$15.1_{\pm 5.7}$	$0.7_{\pm 4.5}$	N/A
gemini-2.5 w/o thinking	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$3.3_{\pm 10.5}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$20.0_{\pm 39.8}$	$5.1_{\pm 4.9}$	$0.7_{\pm 8.9}$	$6.3_{\pm 9.8}$
claude-3-7 w/o thinking	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$5.0_{\pm 15.8}$	$5.0_{\pm 15.8}$	$10.0_{\pm 31.5}$	$15.0_{\pm 5.8}$	$0.2_{\pm 8.2}$	$12.4_{\pm 8.6}$
gpt-4.1	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$10.0_{\pm 31.5}$	$13.0_{\pm 6.8}$	$11.2_{\pm 5.6}$	$18.5_{\pm 10.9}$
qwen-vl-max	$0.0_{\pm 0.0}$	$1.2_{\pm 2.0}$	$0.0_{\pm 0.0}$	$5.0_{\pm 15.8}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$14.1_{\pm 7.3}$	$13.2_{\pm 20.2}$	$-0.6_{\pm 8.3}$
doubao-1-5-vision-pro	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$5.0_{\pm 5.0}$	$10.0_{\pm 21.0}$	$0.0_{\pm 0.0}$	$5.0_{\pm 31.5}$	$7.8_{\pm 8.8}$	$8.1_{\pm 6.3}$	$16.2_{\pm 15.0}$
grok-2-vision	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$5.1_{\pm 5.3}$	$-0.4_{\pm 5.8}$	$3.0_{\pm 3.9}$
Qwen2.5-VL-72B-Ins.	$6.2_{\pm 6.6}$	$0.8_{\pm 1.8}$	$2.9_{\pm 6.2}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$16.2_{\pm 9.6}$	$19.6_{\pm 25.7}$	$23.3_{\pm 22.9}$
InternVL3-78B	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$1.7_{\pm 5.2}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$4.8_{\pm 3.9}$	$-1.8_{\pm 9.2}$	$8.2_{\pm 7.6}$
Llama-3.2-90B-Vision-Ins.	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$	$5.5_{\pm 3.7}$	$3.6_{\pm 4.9}$	$3.0_{\pm 8.8}$
Random	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

E.11 Raw results without normalization

The decision-making results in all environments without normalization are listed in Table 6. We also report the firework return for *Hanabi* and the score reward and step reward for *Atari Pong*.

<BACK> to Section 3.3 Decision-making.

F Additional experiment results

F.1 Multimodal observation results

We select a cooperative card game *Hanabi*, a competitive board game *Breakthrough*, and a mixed-motive video game *Monster Hunt*, and evaluate the performance of fifteen VLMs with multimodal observations and text-only observations. The evaluation results of strategic reasoning are shown in Table 7 and the evaluation results of decision-making are shown in Table 8. In general, most

VLMs achieve better results with text-only observations in both strategic reasoning and decision-making. However, multimodal observations provide richer information and should lead to at least the same performance. These results demonstrate that VLMs can fail to extract visual information in multimodal observations and improve strategic reasoning and decision-making performance.

<BACK> to Section 4.1 Multimodal observations.

F.2 Test-time scaling results

We consider the same three games as in multimodal observation experiments and evaluate the performance of six chat VLMs and three open-source VLMs with IO prompting and Chain-of-Thought (CoT) prompting Wei et al. (2022). We do not consider CoT prompting for reasoning models because they already generate step-by-step reasoning by themselves. The evaluation results of strategic reasoning are shown in Table 7, and the evaluation results of decision-making are shown in Table 8. In general, CoT prompting leads to much better performance, which shows that test-time scaling methods like reasoning and CoT prompting can substantially improve VLMs' performance.

<BACK> to Section 4.2 Test-time scaling.

F.3 Social behaviors and personas results

We evaluated the social behavior of all models in three mixed-motive games. Additionally, we selected three well-performing reasoning models to test their behavior under different persona settings.

F.3.1 Social behaviors

For *Coin Dilemma*, the social behaviors of different models are shown in Fig. 19a, 19d, and 19g. Compared to reasoning models, chat models exhibit inferior performance in coin collection, leading to fewer instances of both cooperation and defection. The best-performing chat model, `GPT-4.1`, demonstrates more defections than cooperative actions, suggesting a behavioral tendency that prioritizes self-interest over collective welfare. In contrast, the open-source model `InternVL3-78B` displays the opposite behavioral pattern.

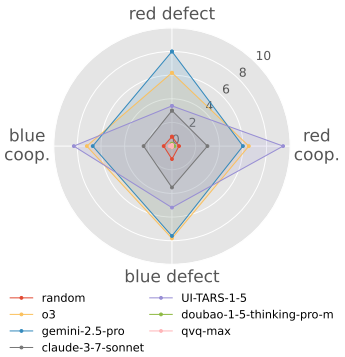
For *Monster Hunt*, the social behaviors of different models are shown in Fig. 19b, 19e, and 19h. For reasoning models, `o3`, `gemini-2.5-pro`, and `claude-3-7-sonnet` prefer eating apples alone; `doubao-1-5-thinking-pro-m` prefers cooperatively defeating the monster; `qvq-max` performs a little better than random. For chat models, `gpt-4.1` prefer eating apples alone; `doubao-1-5-vision-pro` and `qwen-vl-max` prefer cooperatively defeating the monster; the other models perform a little better than random. For open-source models, `Qwen2.5-VL-72B-Instruct` prefers cooperatively defeating the monster; the other models perform a little better than random.

For *Battle of the Colors*, the social behaviors of different models are shown in Fig. 19c, 19f, and 19i. Both `gemini-2.5-pro`, `gemini-2.5` w/o thinking, `o3`, and `ui-tars-1-5` achieve a considerable number of game events, demonstrating strong decision-making capabilities. Beyond this, `gemini-2.5-pro` and `gemini-2.5` w/o thinking exhibit a pronounced willingness to cooperate with others, even when such cooperation yields higher rewards for their counterparts.

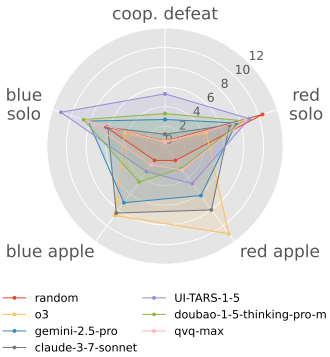
F.3.2 Persona

To investigate the influence of different prompts on the outputs of VLMs, we conducted additional experiments on "personas" across three mixed-motive games. Specifically, we selected three reasoning models—`o3`, `gemini-2.5-pro`, and `doubao-1-5-thinking-pro`—and prompted them to adopt either a *selfish persona* (aiming to maximize their own return) or a *cooperative persona* (aiming to maximize the group's total return). The complete prompt instructions are listed below.

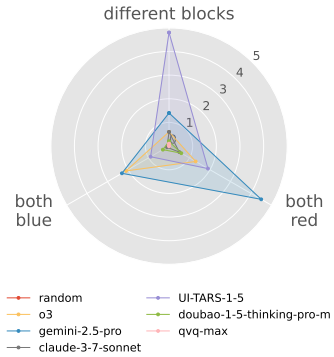
1. *Coin Dilemma*
   - **Common Welfare:** "Consider the common welfare and collect only coins of your own color to avoid harming the other player's score."



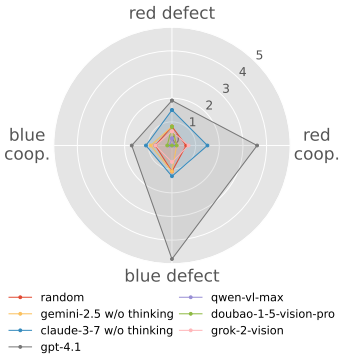
(a) Reasoning VLMs in *Dilemma*.



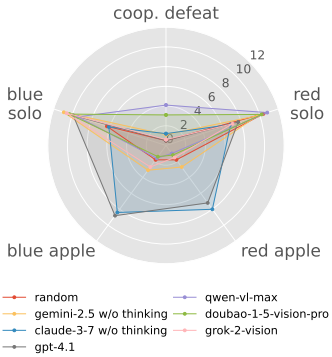
(b) Reasoning VLMs in *Hunt*.



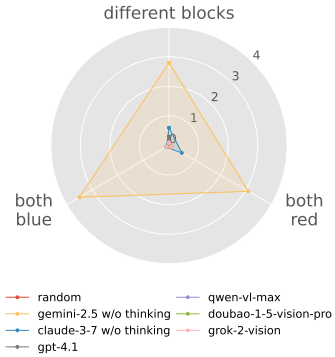
(c) Reasoning VLMs in *Battle*.



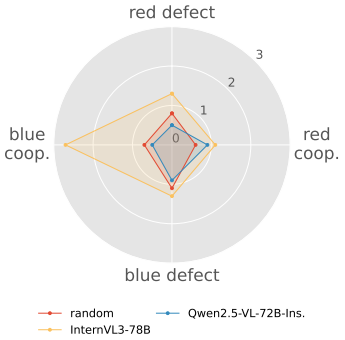
(d) Chat VLMs in *Dilemma*.



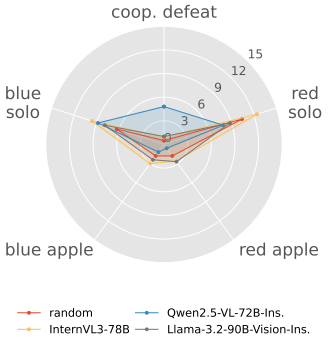
(e) Chat VLMs in *Hunt*.



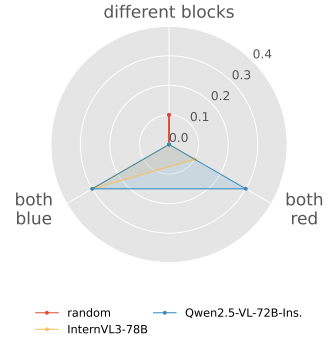
(f) Chat VLMs in *Battle*.



(g) Open-source VLMs in *Dilemma*.



(h) Open-source VLMs in *Hunt*.



(i) Open-source VLMs in *Battle*.

Figure 19: Social behaviors of all models in mixed-motive social dilemma games. Dimensions are agents' behaviors described in Sec. 4.3.

- **Self Interest:** "Consider only your own welfare and collect coins of all colors to maximize your score."

2. *Monster Hunt*

- **Common Welfare:** "To maximize the common welfare, go to the bottom-left corner and wait for the other player to join you." "Stay there and let the monster come to you so that you can defeat the monster together."
- **Self Interest:** "To minimize your risk on behalf of your own welfare, do not cooperate with the other player and only collect apples to maximize your own score."

Table 9: Raw decision-making results under different personas in three mixed-motive games across three reasoning models.

Models	*Hanabi*		*Breakthrough*		*Monster Hunt*	
	Common Welfare	Self Interest	Common Welfare	Self Interest	Common Welfare	Self Interest
`o3`	$11.2_{\pm 1.8}$	$-0.1_{\pm 5.9}$	$54.3_{\pm 5.7}$	$27.5_{\pm 5.8}$	$13.1_{\pm 4.4}$	$0.3_{\pm 0.6}$
`gemini-2.5-pro`	$11.2_{\pm 2.1}$	$-0.7_{\pm 5.7}$	$51.0_{\pm 6.6}$	$26.6_{\pm 4.3}$	$14.7_{\pm 4.2}$	$0.2_{\pm 0.4}$
`doubao-1.5-thinking-pro-m`	$9.9_{\pm 4.5}$	$-3.5_{\pm 9.1}$	$13.1_{\pm 41.1}$	$6.7_{\pm 5.6}$	$5.0_{\pm 4.8}$	$0.1_{\pm 1.2}$

Table 10: Decision-making results for the human baseline. For comparison, we also report the performance of the best reasoning, chat, and open-source models. All values are normalized scores. For each environment, the best result is highlighted in green.

Models	Overall	Cooperative			Comptitive				Mixed-Motive		
		Hanabi	*Overcooked*	*KAZ*	*Board*	*Poker*	*Pong*	*MPE*	*Dilemma*	*Hunt*	*Battle*
Oracle	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
Human	62.7	72.9	87.9	84.2	100.0	7.6	66.5	22.4	38.1	66.0	81.7
`o3`	31.4	55.8	15.6	29.5	65.0	61.8	8.6	37.2	−0.4	24.0	16.7
`gpt-4.1`	4.8	0.0	−0.5	−5.3	0.0	−7.1	0.2	31.5	17.8	11.2	0.5
`Qwen2.5-VL-72B-Ins.`	3.0	0.8	−0.5	−5.3	0.0	−3.1	−0.8	19.6	0.0	19.6	−0.5
Random	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

3. *Battle of the Colors*

- **Common Welfare:** "To maximize the common welfare, cooperate with the other player and go to the nearest color block to both players (regardless of the color)."
- **Self Interest:** "To maximize your own welfare, do not cooperate with the other player and only go to the color block of your own color."

The complete experimental results are presented in Table 9. It can be clearly observed that, across all three mixed-motive games, the choice of prompt significantly shapes the action decisions of VLMs, thereby exerting a substantial impact on the final scores.

<BACK> to Section 4.3 Social behaviors.

F.4 HUMAN EXPERIMENTS

In order to better evaluate the capabilities of VLMs, we additionally conducted a human study. A total of 26 college students were participated in our experiments, where they were grouped to perform all decision-making games. This provides an effective human baseline for VS-BENCH, offering an intuitive reference to measure the performance gap between current VLMs and human-level decision making. The complete results of the human study are reported in Table 10. For a more comprehensive comparison, we also include the best-performing reasoning model `o3`, the chat model `gemini-2.5` w/o thinking, and the open-source model `Qwen2.5-VL-72B-Ins`. The results clearly show that, at the current stage, humans still substantially outperform even the strongest reasoning models. Specifically, the normalized score of human participants is nearly twice as high as that of `o3`.

<BACK> to Section 4.4 Human experiments.

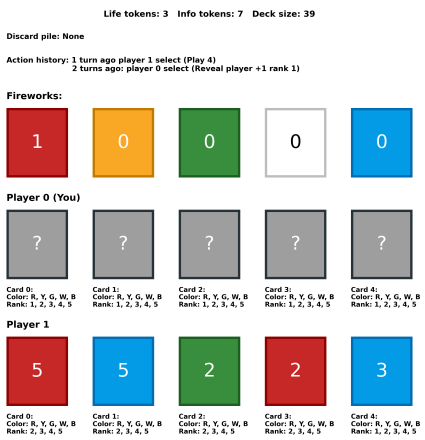
F.5 BEYOND THE TWO-PLAYER SETTING

Although the main body of our work primarily emphasizes the two-player setting—because it is the most canonical and widely studied case in multi-agent research, serving as a foundational step toward more general n-agent evaluation—our benchmark is in fact a general framework that can be readily applied to n-player games with $n > 2$. To substantiate this claim, we further conduct experiments on two additional environments: (1) three-player *Hanabi* and (2) the three-player *Coin Dilemma*. We evaluate several representative models and the corresponding results are reported in Table 11.

Table 11: Decision-making results of 7 representative VLMs on multi-agent games, including three-player *Hanabi* and the three-player *Coin Dilemma*.

Models	Three-Player *Hanabi* standard	Three-Player *Hanabi* firework	Three-Player *Dilemma*
o3	$14.3_{\pm 2.19}$	$14.3_{\pm 2.19}$	$-8.7_{\pm 4.45}$
doubao-1.5-thinking-pro-m	$8.7_{\pm 6.14}$	$10.4_{\pm 3.95}$	$1.07_{\pm 10.37}$
gpt-4.1	$0_{\pm 0}$	$3.8_{\pm 1.40}$	$5.34_{\pm 8.87}$
gemini-2.5-flash	$0_{\pm 0}$	$2.3_{\pm 1.41}$	$-3.56_{\pm 10.89}$
doubao-1.5-vision-pro	$0_{\pm 0}$	$4.3_{\pm 1.41}$	$-2.14_{\pm 4.98}$
Qwen2.5-VL-72B-Instruct	$0.3_{\pm 0.48}$	$0.4_{\pm 0.52}$	$-4.63_{\pm 8.90}$
Llama-3.2-90B-Vision-Instruct	$0_{\pm 0}$	$1_{\pm 1.05}$	$-1.07_{\pm 5.10}$

Visual observation:



VLM response:

```
...
Now, Player 1's possible actions:
play, discard, or reveal. Let's
consider playing first. The fireworks
for Red have 1, so the next needed is
Red 2. Player 1 has R2 (card 3),
which is the next in Red. Playing
that would be valid. Green needs 1
first, but Player 1 has G2, which
can't be played yet since Green's 1
isn't down. The 5s (R5, B5) can't be
played until their sequences are up
to 4. B3 is Blue 3, but Blue needs 1
first. So the valid play here is R2
(card 3) since Red is at 1, and R2 is
the next.
...
```

Figure 20: Failure case example of strategic reasoning in *Hanabi*.

G FAILURE CASE EXAMPLES

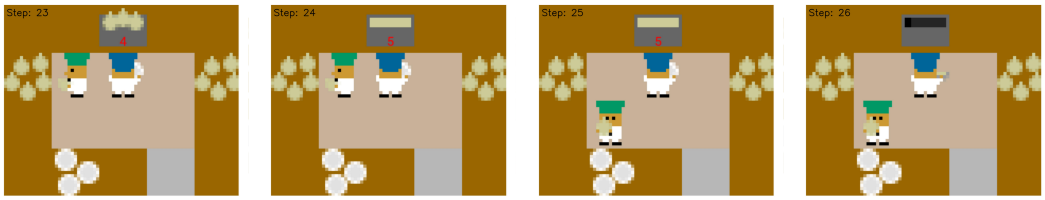
G.1 STRATEGIC REASONING

We present three illustrative failure cases in strategic reasoning from different game environments.

In *Hanabi*, VLM agents only observe the other agent's hands but not their own hands, creating a distinct information asymmetry. An example with visual observation and the VLM's response is shown in Fig. 20. When predicting the other agent's next move, VLMs often immediately identify which visible card could yield points and assume the other agent will play it. This prediction, however, overlooks a critical fact: from the other agent's perspective, their information about that card is incomplete, so they would not risk losing a life token by playing it prematurely. Instead, they would wait for more definitive clues before making that play.

In *Overcooked*, accurately predicting the next action of the other agent requires robust visual perception capabilities and a thorough comprehension of image information. An example with visual observation and the VLM's response is shown in Fig. 21. In this case, the VLM should predict the blue chef's action from the green chef's perspective. Despite correctly identifying from historical frames that the soup was cooked, the VLM's visual perception is inadequate: it fails to recognize that the blue chef was already holding the soup, plated in a dish. In addition, the VLM overlooks a critical game rule, mistakenly believing that soup could be collected from the pot before a dish was acquired. As a result of these combined deficiencies in visual understanding and rule application, the VLM predicts the erroneous action <INTERACT>.

Visual observation:



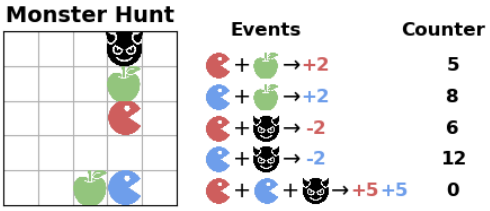
frame 1 frame 2 frame 3 frame 4

VLM response:

```
...
Actually, let me look more carefully at what chef_0 is doing. In
frames 23, 24, and 25, chef_0 is standing in the same position and
hasn't moved. In frame 26, the pot shows a black strip, indicating
the soup is fully cooked.
According to the rules, after the soup is cooked for 5 timesteps,
the next step is to plate it using a dish. Since chef_0 is directly
in front of the pot, the most likely action would be <INTERACT> to
take the soup from the pot.
However, this would require chef_0 to have a dish already. Since I
don't clearly see chef_0 holding a dish, they would first need to
get one.
Given this analysis, I believe chef_0 will most likely choose
<INTERACT> to interact with the pot.
...
```

Figure 21: Failure case example of reasoning in *Overcooked*.

Visual observation:



VLM response:

```
...
If the red player moves <UP>,
it collects an apple, and gets
+2 points. The red player's
best move to maximize their
own score is to go up.
...
```

Figure 22: Failure case example of strategic reasoning in *Monster Hunt*.

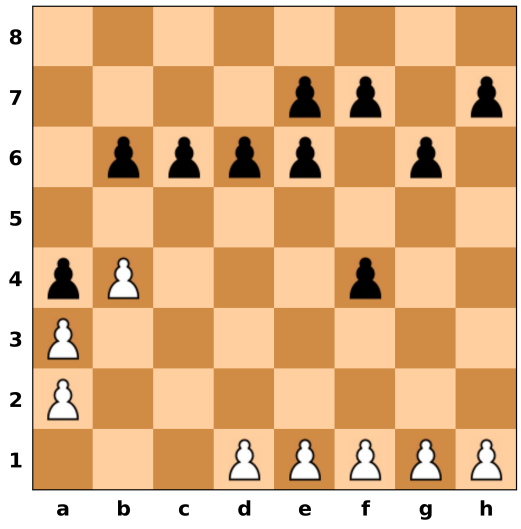
In *Monster Hunt*, in order to predict the next action of the other player, it is important to track the positions and movements of all game elements, rather than focusing on only part of the game mechanics and make short-sighted predictions. An example with visual observation and the VLM's response is shown in Fig. 22. The VLM notices the apple is just one step above the red player, but does not pay attention to the monster on top of the apple. The VLM predicts the red player to go up to collect the apple, while the red player actually goes down to flee from the monster.

G.2 Decision-making

We also present three illustrative failure cases in decision-making from different game environments.

In *Breakthrough*, a strong strategy must excel at both offense and defense. However, VLMs occasionally overlook situations in which the opponent's pieces are on the verge of victory and require immediate defensive intervention. Instead, they persistently advance their most distant piece by one step, thereby neglecting the critical balance between attack and defense. An example with visual observation and the VLM's response is shown in Fig. 23.

Visual observation:



VLM response:

```
...
Now, considering optimal play,
White would want to advance
their pieces towards Black's
side, possibly creating threats
or blocking Black's moves. The
b4 piece is already advanced.
Moving it to c5 (b4c5) would
bring it closer to the center
and maybe set up a capture or
block.
Looking at the starting row
pieces (d1 to h1), moving them
straight forward (like d1d2,
e1e2, f1f2, g1g2, h1h2) or
diagonally. Straight moves are
safe but maybe less aggressive.
Diagonal moves could
potentially set up captures
later.
...
```

Figure 23: Failure case example of decision-making in *Breakthrough*.

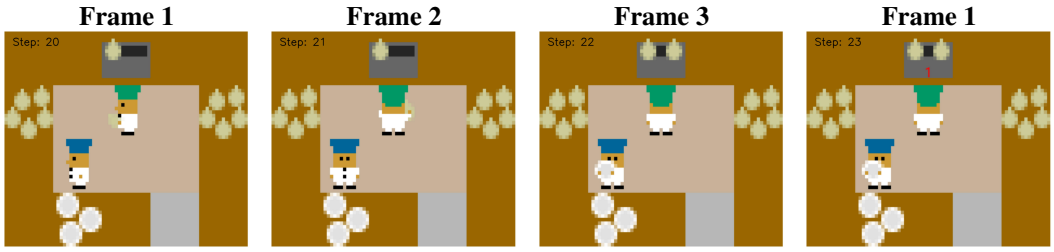


Figure 24: Failure case example of decision-making in *Overcooked*.

In *Overcooked*, to correctly prepare a soup with three onions, the VLM agent needs to accurately manage the number of onions. A common failure mode of VLMs is that they often neglect these precise quantity requirements. As illustrated in Fig. 24, the green chef initiates cooking by performing an <INTERACT> operation after adding only the second onion for this recipe. Concurrently, the blue chef prematurely begins to retrieve a plate. As a consequence, the dish, even when served, fails to earn the delivery reward because the ingredient count requirement is not met.

In *Atari Pong*, in order to successfully intercept the ball, the VLM agent needs to accurately predict the landing point of the ball. A common failure mode of the VLMs is that they may over-adjust their paddle. As illustrated in Fig. 25, the VLM on the right detects that the ball is moving downward and then move the paddle down, but misses the landing point due to over-aggressive adjustment.

<BACK> to Section 4.4 Failure cases.

H ADDITIONAL ENVIRONMENTS

To provide a more comprehensive evaluation, we extend VS-BENCH with four additional environments. Specifically, we introduce two simpler games, a cooperative game *Tiny-Hanabi* and a competitive game *Tic-Tac-Toe*, which serve as lightweight testbeds. In addition, since our benchmark is a general framework that can be readily applied to n-player games with $n > 2$, we further include two three-player extensions, a cooperative game given by the three-player version of *Hanabi* and a

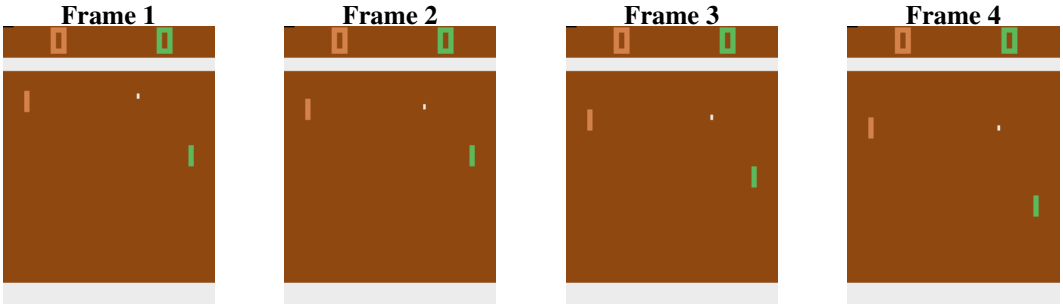


Figure 25: Failure case example of decision-making in *Atari Pong*.

mixed-motive game given by the three-player version of *Coin Dilemma*, which naturally extend the original two-player settings.

1. ***Tiny Hanabi*** is a simplified version of *Hanabi* with only two colors, three ranks, and a hand size of three cards. Detailed rules are provided in Appendix I.2. An example visual observation is shown in Fig. 26.
2. ***Tic-Tac-Toe*** is a competitive two-player board game played on a 3×3 grid. The first player to align three marks horizontally, vertically, or diagonally wins. This game has a relatively small state space, and its optimal strategies are known to guarantee a draw under perfect play. We use it as a preliminary testbed for evaluating VLMs in competitive multi-agent environments. An example observation is shown in Fig. 28. Detailed descriptions can be found at `https://en.wikipedia.org/wiki/Tic-tac-toe`.
3. **Three-player *Hanabi*** extends the original *Hanabi* to three players with other rules unchanged. Detailed rules are provided in Appendix I.2. An example observation is shown in Fig. 27.
4. **Three-player *Coin Dilemma*** extends the original game to a three-player setting with other rules unchanged. Detailed rules are provided in Appendix I.9. An example observation is shown in Fig. 29.

<BACK> to Section 2 VS-BENCH environments.

I ENVIRONMENT DETAILS

In this section, we first introduce the four core abilities that VS-BENCH aims to evaluate in multi-agent settings. We then provide detailed descriptions of the ten games included in VS-BENCH, covering their rules, gameplay dynamics, and relevant implementation details.

I.1 ABILITY DEFINITION

In VS-BENCH, we identify four core abilities that play crucial roles in multi-agent interaction:

1. **Spatial Reasoning.** This ability emphasizes a model's capacity for spatial perception, namely, whether it can interpret the current environment based on visual inputs and the rules of the game. It serves as the foundation for VLMs in interactions. In particular, continuous-space games such as *Pong* and *KAZ* are highly dependent on visual understanding.
2. **Theory of Mind (ToM).** This ability evaluates whether a VLM can accurately infer an opponent's intentions and subsequently make better action choices. For example, in *Hanabi*, players must reason about their teammates' true intentions based on very limited hints.
3. **Long-Term Planning.** This ability reflects a model's competence in long-horizon reasoning, particularly its capacity to anticipate future actions at each decision point in order to maximize cumulative rewards. Both *Hanabi* and *Breakthrough* exemplify this demand for strong planning skills, with the former requiring players to operate under a limited set of cards, while the latter embodies the general characteristics of board games.

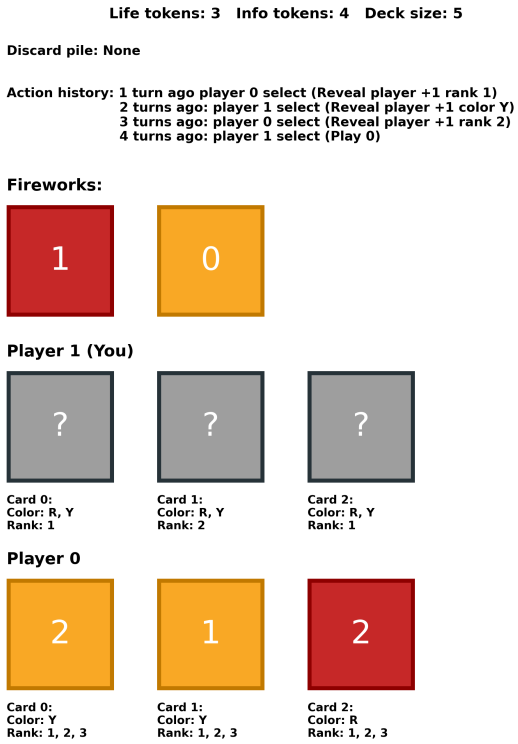


Figure 26: *Tiny Hanabi*.

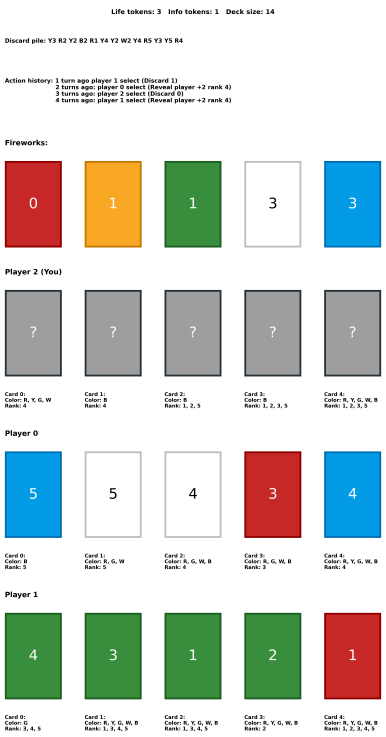


Figure 27: Three-player *Hanabi*.

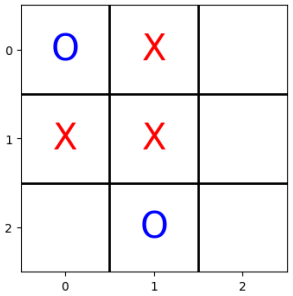


Figure 28: *Tic-Tac-Toe*.

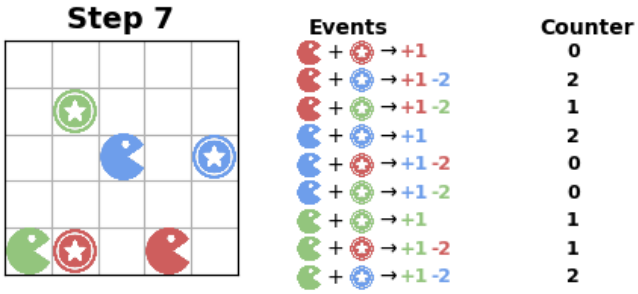


Figure 29: Three-player *Coin Dilemma*.

4. **Team Collaboration.** This ability evaluates how effectively models coordinate with one another. It is not relevant in purely competitive *games*, but becomes crucial in cooperative settings, where improved collaboration directly translates into higher returns. For instance, *Hanabi* presents a considerable challenge by demanding close cooperation between players.

I.2 HANABI

Hanabi (Bard et al., 2020) is a partially observable card game where players can observe others' cards but not their own. Each card has a color and a rank that can only be revealed through hint actions at the cost of an information token. To succeed, agents must coordinate to play cards in rank order for five colors. We consider the two-player full game, which is widely used for research on theory of mind, zero-shot coordination, and ad-hoc teamplay (Hu et al., 2020; Hu & Sadigh, 2023).

Game rules. Detailed game descriptions can be found in `https://en.wikipedia.org/wiki/Hanabi_(card_game)`.

1. *Hanabi* is a cooperative card game for 2 players.
2. The deck consists of 5 colors: R (Red), Y (Yellow), G (Green), W (White), B(Blue), with ranks ranging from 1 to 5. Each color contains 10 cards: three of rank 1, two each of rank 2 through 4, and one of rank 5, for a total of 50 cards.
3. Each player holds 5 cards in hand.
4. There are 8 Info tokens (used to give hints) and 3 Life tokens (penalties for misplays).
5. As in blind man's bluff, players can see each other's cards but they cannot see their own. Play proceeds around the table; each turn, a player must take one of the following actions: <PLAY>, <DISCARD i>, <REVEAL color c>, <REVEAL rank r>.
6. The game ends immediately when either all Life tokens are used up, resulting in a game loss with a score of 0, or when all 5s have been successfully played, resulting in a game win with a score of 25. Otherwise, the game continues until the deck runs out and one final round is completed. At the end of the game, the final score is calculated as the sum of the highest card played in each suit, up to a maximum of 25 points.

Visual observation. An example is shown in Fig. 7. The visual observation has four parts:

1. Basic Information: counts of remaining life tokens, information tokens, and deck size.
2. Historical Information: all discarded cards, together with each player's two most recent actions.
3. Fireworks: current progress of each color's fireworks stack, indicating the highest played rank.
4. Players' Hands: each player's hand, with annotations beneath each card showing all possible colors and ranks deduced from received hints.

I.3 OVERCOOKED

Overcooked (Games, 2016) is a popular video game where two chefs cooperate to cook and serve dishes in a kitchen. Each dish requires multiple operations like navigating, chopping, cooking, plating, and delivering, which are difficult to coordinate even for human players. Our implementation is based on Overcooked-AI (Carroll et al., 2019), a well-known environment for zero-shot coordination and human-AI interactions (Strouse et al., 2021; Zhang et al., 2024).

Game rules. Detailed game descriptions can be found in `https://github.com/HumanCompatibleAI/overcooked_ai`.

1. Overcooked is a cooperative game where two chefs collaborate to cook and serve soups in 50 timesteps.
2. The chefs can move in the available area and cannot move to the counter.
3. The chefs can interact with the object on the tile that they are facing.
4. A soup is cooked in the following steps:
    a. Pick up (interact) 1 onion and place (interact) it in the pot.
    b. After placing 3 onions in the pot, open (interact) the pot and cook for 5 timesteps. The pot will show how long the soup has been cooked.
    c. When the pot shows the number 5, the soup is finished. Pick up (interact) a dish to plate (interact) the soup.
    d. Deliver the soup and put (interact) it on the serving location.

Visual observation. An example is shown in Fig. 8. On the left is the current game state, showing the overall kitchen layout, the positions and orientations of both chefs, and the items they hold. On the right is a legend explaining the visual representations of game elements, such as objects and chef orientations, used in the game state.

I.4 KNIGHTS ARCHERS ZOMBIES

Game rules. Detailed game descriptions can be found in `https://pettingzoo.farama.org/environments/butterfly/knights_archers_zombies/`.

1. *Knights Archers Zombies* (KAZ) is a cooperative survival game played on a 2D battlefield. The objective is to survive as long as possible while maximizing zombie kills and protecting both yourself and your teammate.
2. Zombies spawn from the top and walk downward toward the bottom along unpredictable paths.
3. Each player controls either a Knight (melee fighter) or an Archer (ranged fighter), both starting at the bottom of the field.
4. In the illustration, green units represent zombies, red units represent Archers, and white units represent Knights.
5. The game ends under either of the following conditions: (a) one agent dies, or (b) a zombie reaches the bottom border.
6. Rewards are assigned as follows: each zombie killed yields +1 point.
7. Knights attack with a mace, striking in an arc in front of them, whereas Archers attack by shooting arrows in straight lines.
8. All agents can move forward or backward and rotate left or right to change their direction.

Visual observation. An example is shown in Fig. 9. The environment is a two-dimensional plane where zombies randomly spawn from the top and continuously move downward, with their horizontal movement being stochastic. Both players are initialized at the bottom side of the environment.

I.5 BREAKTHROUGH

Breakthrough (Troyka, 2000) is a chess-like board game with simplified rules and identical pawns. Two players compete to advance their pieces across an 8×8 grid to reach the opponent's back row. The game is deceptively simple, yet it exhibits deep combinatorial complexity and sharp tempo imbalance between attack and defense, making it a suitable environment for studying multi-step lookahead and adversarial decision-making (Lorentz & Horey, 2013; Saffidine et al., 2012).

Game rules. Detailed game descriptions can be found in `https://en.wikipedia.org/wiki/Breakthrough_(board_game)`.

1. Breakthrough is a two-player strategy game played on an 8x8 grid.
2. Each player controls pieces of a color: 'White' or "Black". 'White' starts at the bottom (rows 1 and 2), while "Black" starts at the top (rows 7 and 8).
3. If 'White' moves a piece to row 8, 'White' wins the game. Conversely, if "Black" moves a piece to row 1, "Black" wins the game.
4. Players alternate turns, moving one piece per turn, with "Black" going first.
5. A piece may only move one space straight or diagonally forward, and only if the destination square is empty.
6. A piece may only capture an opponent's piece by moving one space diagonally forward into its square. In this case, the opponent's piece is removed, and your piece takes its place.
7. "Black" moves forward by decreasing row indices (downward), while 'White' moves forward by increasing them (upward).
8. Moves are specified by their start and end positions. For example, 'a2a3' indicates moving a piece from a2 (column a, row 2) to a3 (column a, row 3).
9. The board is labeled with columns a-h and rows 1-8. Thus, h8 is the top-right corner, and a1 is the bottom-left corner.

Visual observation. An example is shown in Fig. 11. The figure illustrates the current positions of both black and white pieces on the board. Row and column indices are annotated on the left and bottom sides of the image, respectively.

I.6 KUHN POKER

Kuhn Poker (Kuhn, 1950) is a simplified variant of Texas Hold'em (Brown & Sandholm, 2019; Moravčík et al., 2017) designed to study imperfect-information games for game-theoretic analysis. It has a three-card deck and a single betting round where two players take turns to either check or bet with limited stakes. The game has been used as a classic environment for counterfactual reasoning and decision-making with imperfect information (Lanctot et al., 2019; Muller et al., 2019).

Game rules. Detailed game descriptions can be found in `https://en.wikipedia.org/wiki/Kuhn_poker`.

1. Kuhn poker is a two-player card game. The deck includes only three cards: King (K) > Queen (Q) > Jack (J).
2. At the start of each game, both player 0 and player 1 place 1 chip into the pot as a blind ante.
3. Each player is dealt a card as private information, and the third card is set aside unseen.
4. The two players take turns acting, starting with player 0. A player can choose to:
    a. <PASS>: place no additional chips into the pot.
    b. <BET>: place 1 additional chip into the pot.
5. If a player chooses to <PASS> after the other player's <BET>, the betting player wins the pot.
6. If both players choose to <PASS> or both players choose to <BET>, the player with the higher card wins the pot.

Visual observation. An example is shown in Fig. 12. Each player receives a visual representation of their private card and the current chips in the pot.

I.7 MPE

Game rules. Detailed game descriptions can be found in `https://pettingzoo.farama.org/environments/mpe/simple_push/`.

1. The environment consists of two moving players (a blocker and a charger) and a target position within a two-dimensional space.
2. The charger's goal is to reach the target location.
3. The blocker's goal is to prevent the charger from reaching the target by blocking and pushing.
4. At each step:
    (a) The charger is rewarded based on its distance to the target — the smaller the distance, the higher the reward.
    (b) The blocker is rewarded when it is close to the target and when the charger remains far from the target (i.e., the difference in distances).
5. The player with the higher cumulative reward at the end of the game is declared the winner.

Visual observation. An example is shown in Fig. 15. The target is represented by a red X. The charger and blocker are represented by green and blue circles, respectively. The cordinate grid on the image indicates the positions and distances between the players and the target.

I.8 ATARI PONG

Atari Pong (Atari, 1972) is a classic arcade video game where two players control paddles to hit a ball across the screen. With raw pixel observations and competitive dynamics, the game has become a canonical environment in the Arcade Learning Environment (ALE) (Bellemare et al., 2013), which requires spatio-temporal reasoning and strategic gameplay (Mnih et al., 2013; 2015).

Game rules. Detailed game descriptions can be found in `https://en.wikipedia.org/wiki/Pong`.

1. Atari Pong is a zero-sum game played on a 2D screen with two players (left and right) and a ball.
2. Each player controls a paddle and receives rewards on different events:
   a. If the ball passes your paddle: the opponent +1 point.
   b. If the ball passes the opponent's paddle: you +1 point.
3. The ball bounces off the top/bottom walls and the paddles.
4. Paddles can only move vertically within the top and bottom walls.
5. First player to score 3 points wins.

Visual observation. An example is shown in Fig. 13. The two players each control a paddle on the side of the screen to hit a ball back and forth with each other. The paddles are vertical rectangles, and the ball is a white square. The players score if the ball passes their opponent's paddle. The built-in bot controls the left paddle, while the VLM agent controls the right paddle. The scores of both players are displayed at the top of the screen.

I.9 COIN'S DILEMMA

Coin Dilemma (Lerer & Peysakhovich, 2017) is a grid-world environment inspired by the classic Prisoner's Dilemma (Rapoport & Chammah, 1965) in game theory. A red player and a blue player move in a 5×5 grid world to collect red and blue coins. A player earns 1 point for collecting any coin. However, if the red player collects a blue coin, the blue player is penalized 2 points, and vice versa. This setup creates a tension between mutual benefit and self-interest: while both players collecting coins of their own color leads to a win-win result, unilateral defection to collect all coins maximizes one's own gains at the other's expense. Therefore, the game has been a common environment for studying rational reasoning, opponent shaping, and social dilemma resolution (Foerster et al., 2017; Lu et al., 2022; Rutherford et al., 2024).

Game rules. Detailed game descriptions can be found in Lerer & Peysakhovich (2017).

1. Coin Dilemma is a general-sum game played on a 5x5 grid board with two players (red and blue) and two types of coins (red and blue).
2. Players receive rewards on different events:
   a. A player collects one coin of its own color: the player +1 point.
   b. A player collects one coin of the other player's color: the player +1 point, the other player -2 points.
3. New coins spawn randomly on the board after each collection.

Visual observation. An example is shown in Fig. 16. On the left of the image is a grid map showing the current positions of all game elements, including two players (red and blue) and two coins (red and blue). The players are each represented by a Pac-Man icon, and the coins are each represented by a coin icon. On the right of the image is a table demonstrating the rewards of each event and a corresponding counter tracking the number of occurrences for that event.

I.10 MONSTER HUNT

Monster Hunt (Peysakhovich & Lerer, 2017) is a grid-world environment inspired by the classic Stag Hunt (Rousseau, 1985) in game theory. Two players move in a 5×5 grid world to individually eat an apple for 2 points or jointly defeat a monster for 5 points. A player who confronts the monster alone, however, is penalized 2 points. This leads to multiple Nash equilibria where agents can both safely eat apples alone or take risks to cooperate for higher rewards. The game is used to investigate trust formation and risk-sensitive decision-making (Leibo et al., 2021; Tang et al., 2021).

Game rules. Detailed game descriptions can be found in Peysakhovich & Lerer (2017).

1. Monster Hunt is a general-sum game played on a 5x5 grid board with two players (red and blue), one monster, and two apples.

2. The monster moves towards the closest player in each step.
3. Players move in the grid-world and receive rewards on different events:
    a. One player eats an apple: the player +2 points and the apple respawns at a random position.
    b. One player encounters the monster alone: the player -2 points and respawns at a random position.
    c. Two players defeat the monster together: both players +5 points and the monster respawns at a random position.

Visual observation. An example is shown in Fig. 17. On the left of the image is a grid map showing the current positions of all game elements, including two players (red and blue), two apples, and a monster. The players are each represented by a Pac-Man icon, the apples are each represented by a green apple icon, and the monster is represented by a black demon icon. On the right of the image is a table demonstrating the rewards of each event and a corresponding counter tracking the number of occurrences for that event.

I.11 Battle of the Colors

Battle of the Colors is a grid-world environment inspired by the classic Battle of the Sexes (Luce & Raiffa, 1957) in game theory. We propose and design this game in a manner similar to the previous two social dilemma games. A red player and a blue player move in a 5×5 grid world with a red block and a blue block. If both players move to the red block, the red player earns 2 points while the blue player earns 1 point, and vice versa. If players move to two blocks of different colors, both players earn 0 points. Therefore, while coordination is mutually beneficial, each player strictly prefers choosing the block of their own color, creating a conflict of interest that produces two payoff-asymmetric Nash equilibria and a mixed equilibrium. This game thus challenges agents to solve conflicting preferences while avoiding coordination failure, making it suitable for studying equilibrium selection, bargaining dynamics, and social fairness.

Game rules.

1. The Battle of the Colors is a general-sum game played on a 5x5 grid board with two players (red and blue) and two types of blocks (red and blue).
2. Players receive rewards on different events:
    a. When both players are on a red block: red player +2 points, blue player +1 point, and the red block will be refreshed to a new random position.
    b. When both players are on a blue block: red player +1 point, blue player +2 points, and the blue block will be refreshed to a new random position.
    c. When players are on different blocks: both players +0 points, and both blocks will be refreshed to new random positions.

Visual observation. An example is shown in Fig. 18. On the left of the image is a grid map showing the current positions of all game elements, including two players (red and blue) and two colored blocks (red and blue). The players are each represented by a Pac-Man icon. On the right of the image is a table demonstrating the rewards of each event and a corresponding counter tracking the number of occurrences for that event.

<BACK> to Section 2.2 Multi-agent environments.

J Use of LLMs

For this manuscript, we use LLMs only to aid and polish the writing. Their use is limited to improving grammar, clarity, and overall readability.