MARS-RA: Rank Aggregation for Credit Assignment via Multimodal Comparisons in Embodied Multi-Agent Cooperation

Dawei Wang[1], Di Zhao[2], Xinyuan Liu[1], Marci Chi Ma[1], Xiaoyang Liu[1], Chengming Zhou[1], Gary Ushaw[1], Richard Davison[1],
[1]Newcastle University, United Kingdom
[2]University of Auckland, New Zealand

Abstract

Credit assignment is a fundamental challenge in cooperative multi-agent reinforcement learning, particularly in embodied AI settings characterized by limited and delayed feedback as well as dynamically changing numbers of active agents. We propose MARS-RA, a framework that reformulates credit assignment as a rank aggregation problem using contribution-based pairwise comparisons among agents generated by large multimodal models. This shift from absolute to relative estimation ensures robustness against noise and dynamic agent participation, converting comparison results into contribution scores for potential-based reward shaping. We provide theoretical justification for the convergence and robustness of the proposed framework, and show that Shapley values can be used as an interpretive reference. Experimental results on challenging tasks of different types indicate that MARS-RA can guide agents toward effective cooperation.

1 Introduction

In Embodied Artificial Intelligence (AI) (Turing, 2021; Clark, 1998), multi-agent reinforcement learning (MARL) has emerged as the canonical solution for enabling multiple agents to cooperate in dynamic environments (Zhang et al., 2021). Credit assignment is a fundamental challenge in cooperative MARL, where an agent struggles to disentangle its own contribution to the global reward signal from the simultaneous actions of other agents in the environment (Minsky, 2007). The credit assignment problem can lead to suboptimal policies and unpredictable behaviors (Wong et al., 2023), which in turn result in concrete negative outcomes for cooperative embodied agents, such as wasted resources (Patel et al., 2023) or increased collision risks (Serra-Gómez et al., 2023). Therefore, addressing the credit assignment problem is a prerequisite for ensuring the safe and effective operation of cooperative embodied AI system.

However, the credit assignment problem in embodied AI is fundamentally exacerbated by two inherent characteristics: (1) Embodied agents perceive the world through partial and noisy multimodal egocentric sensors (e.g., RGB, thermal, LiDAR) (Feng et al., 2025), and they often operate in tasks with sparse rewards and long-horizon dependencies, resulting in limited and delayed feedback; (2) Embodied AI systems are open, where agents may leave or enter mid-task due to hardware failures or task demands, violating the standard fixed-agent-set assumption (Tang et al., 2023; Abadi and Soh, 2025). Existing credit assignment methods, such as VDN (Sunehag et al., 2017), QMIX (Rashid et al., 2020), and COMA (Foerster et al., 2017), already suffer performance degradation under insufficient feedback conditions alone (Papoudakis et al., 2020). Consequently, the credit assignment problem in cooperative embodied AI systems remains an unresolved challenge.

To address this challenge, we propose a novel approach that reformulates the credit assignment problem through the lens of rank aggregation (Debreu, 1960), a probabilistic framework for inferring latent scores of n items from multiple partially ordered samples (e.g., pairwise comparisons) (Ma et al., 2022). This reformulation stems from viewing an agent's credit as its latent contribution score, allowing us to estimate credits via rank aggregation over pairwise comparisons of *"which agent contributes more"*. In complex multi-agent embodied cooperation tasks, precisely quantifying an individual agent's contribution is not only mathematically ill-posed (Bakushinsky and Goncharsky, 2012) but also practically tenuous. The key advantage of this formulation is transforming absolute score estimation into relative pairwise comparisons, which makes credit assignment more tractable (Christiano et al., 2017). Furthermore, this formulation intrinsically aligns with the inherent characteristics of embodied AI: (1) pairwise

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 29270–29289

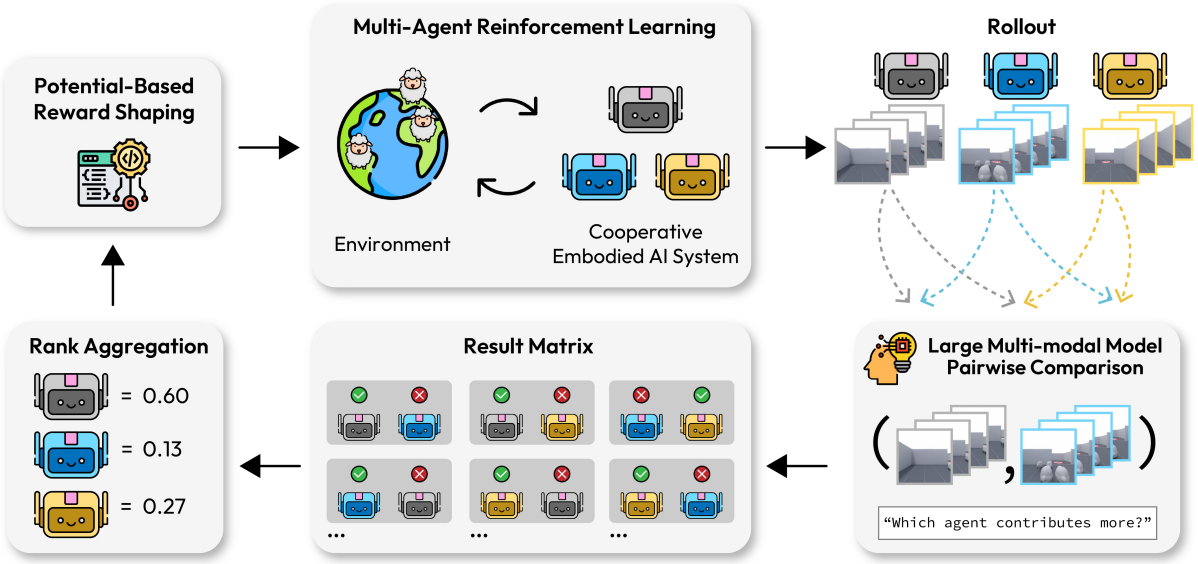


Figure 1: The MARS-RA framework for MARL credit assignment in cooperative embodied AI systems. Agents' visual observations are processed by an LMM to conduct pairwise comparisons based on team contribution; the results are consolidated via rank aggregation to obtain contribution-aligned relevance scores, which are incorporated into MARL training through potential-based reward shaping.

comparisons naturally accommodate dynamically changing numbers of active agents; (2) the inherent robustness of rank aggregation allows it to tolerate comparison noise arising either from biases in the pairwise-comparison generator itself or from embodied AI's partial observability and long-horizon dependencies; and (3) the per-agent contribution scores produced by rank aggregation can serve as dense rewards that complement the environment's sparse reward signals.

We propose **MARS-RA** (**M**ulti-**A**gent **R**eward **S**ystems via **R**ank **A**ggregation), a credit assignment framework that performs pairwise comparisons of agents' contributions toward the team objective and employs rank aggregation to derive contribution-aligned relevance scores for each agent. These scores are then transformed into a potential function (Ng et al., 1999), which is integrated into the MARL training loop to provide dense reward signals for learning. We utilize Large Multimodal Models (LMMs) (Team et al., 2023; Yang et al., 2025; Liu et al., 2023a; Dong et al., 2025; Zhao et al., 2026) as the automatic generator for the required pairwise comparisons. This automated pipeline is capable of meeting the computational demands of MARL training by eliminating the need for costly human annotation. LMMs possess advanced multimodal perception (Huang et al., 2023; Li et al., 2024, 2026, 2025a), are capable of directly ingesting high-dimensional visual observations to perform spatio-temporal inference (Wang et al., 2026; Rocamonde et al., 2023; Zhao et al., 2024), and excel at making pairwise comparisons (Shi et al., 2024; Di Zhao et al., 2025). The overall architecture of the MARS-RA is shown in Figure 1. Since existing embodied AI benchmarks lack settings where the number of active agents can change dynamically, we instantiate a challenging embodied multi-agent task suite in ManiSkill3 (Tao et al., 2024) and construct **MARS-Bench**, where 2 to 4 decentralized embodied agents collaborate to complete three representative tasks: *Pass Gate*, *Herd Sheep*, and *Collect Ball*. Agents in these tasks operate under partial egocentric observations and sparse rewards, and may enter or exit during task execution, closely mirroring the characteristics of embodied AI scenarios.

The main contributions of this work are as follows: (1) We pioneer a reformulation of the credit assignment problem in MARL as a rank aggregation problem. This formulation is well-suited to embodied AI settings and naturally integrates with LMM-based automated comparisons and potential-based reward shaping. (2) We propose MARS-RA, which outperforms strong baselines in our experiments. Further analysis shows that its performance improves with higher accuracy and increased number of LMM-based pairwise comparisons, suggesting that MARS-RA can benefit from continued advances in LMM reasoning capability. (3) We build

MARS-Bench, a benchmark that captures key characteristics and challenges of embodied AI settings, particularly by introducing openness in the number of active agents, encouraging the community to move beyond static team assumptions and develop more resilient algorithms.

2 Related Work

Multi-Agent Cooperation in Embodied AI. Embodied AI (Brooks, 1991) refers to intelligent agents equipped with physical bodies or virtual embodiments that can perceive, act, and adapt through continuous interaction with their environment. Embodied AI tasks often involve cooperation among multiple agents, which can exhibit capabilities beyond those of individual agents. Existing benchmarks for multi-agent embodied AI, such as MQE (Xiong et al., 2024) and TDW-Cook (Zhang et al., 2024), typically exhibit characteristic challenges like partial observability and sparse rewards. However, none of the currently available benchmarks support a dynamic number of active agents, representing a notable gap in the field.

Credit Assignment. Credit assignment is a fundamental challenge in MARL. To address this challenge, various approaches have been proposed to improve credit assignment: value-decomposition methods such as VDN (Sunehag et al., 2017) and QMIX (Rashid et al., 2020) factorize a centralized Q-function into individual agent Q-functions; COMA (Foerster et al., 2017) applies a counterfactual advantage baseline to estimate the individual contributions of each agent; SQDDPG (Wang et al., 2020) utilizes Shapley values as a principled mechanism for credit assignment. Large language models (LLMs) have opened new directions for credit assignment through language-based reasoning and coordination, with works such as LLM-MCA (Nagpal et al., 2025), LCA (Lin et al., 2025), SAMA (Li et al., 2025b), and LERO (Wei et al., 2025) leveraging LLMs for credit evaluation, task decomposition, and hybrid reward design. However, these methods often rely on simplified environments, handcrafted rules, and strong assumptions, limiting their applicability to cooperative embodied AI scenarios.

Rank Aggregation. Rank aggregation is an important task across a wide range of disciplines, including sports (Herbrich et al., 2006), psychology (Critchlow et al., 1991), and bioinformatics (Kolde et al., 2012). In essence, rank aggregation methods treat pairwise comparisons as a means of estimating the latent 'quality' or 'score' of the compared items, such as the popularity of books or the skill levels of athletes. In the machine learning domain, rank aggregation has been applied in various areas. In this work, we pioneer the introduction of rank aggregation in MARL, reinterpreting the credit assignment problem through the lens of rank aggregation.

3 Preliminaries

Embodied multi-agent cooperation can be modeled within the framework of an open decentralized partially observable Markov decision process (Open Dec-POMDP) (Cohen et al., 2017), defined by $(\mathcal{N}, \mathcal{I}, \mathcal{S}, \mathcal{A}, \mathcal{O}, \phi, r, \gamma, T)$, where: $\mathcal{N} = \{1, 2, \ldots, n\}$ is a finite population of n agents. $\mathcal{I} \subseteq \mathcal{P}(\mathcal{N})$ is a finite set of coalitions formed from the agent population $\mathcal{N}$. The coalition $I^t \in \mathcal{I}$ at time t is designated as the *operating coalition*, and an agent i is defined as *active* at time t if $i \in I^t$. $\mathcal{S}$ is the finite set of states. $\mathcal{A} = \{\mathcal{A}^i \mid i \in N\}$ represents the set of action spaces, where $\mathcal{A}^i$ is the finite set of actions for agent i, $\mathcal{A}^I = \times_{i \in I} \mathcal{A}^i$ denotes the set of joint actions a^I available to coalition I, $\boldsymbol{a}^I = (a^i)_{i \in I}$. $\mathcal{O} = \{\mathcal{O}^i \mid i \in \mathcal{N}\}$ represents the set of observation spaces, where $\mathcal{O}^i$ is the finite set of observations for agent i, $\mathcal{O}^I = \times_{i \in I} \mathcal{O}^i$ denotes the set of joint observations $\boldsymbol{o}^I$ available to coalition I, $\boldsymbol{o}^I = (o^i)_{i \in I}$. ϕ is the dynamics model, where $\phi(s', \boldsymbol{o}^{I'}, I' \mid s, I, \boldsymbol{a}^I) \in [0, 1]$ specifies the probability of transitioning to state s' and coalition configuration I', while receiving joint observation $\boldsymbol{o}^{I'}$, given that coalition I took joint action $\boldsymbol{a}^I$ in state s. $r : \mathcal{S} \times \mathcal{A}^I \to \mathbb{R}$ is the reward function, defining the reward $r(s, \boldsymbol{a}^I)$ received after coalition I executes joint action $\boldsymbol{a}^I$ in state s. $\gamma \in [0, 1]$ is the discount factor. T is the planning horizon. Our objective is to learn the joint policy π that maximizes the expected discounted sum of rewards, where the rewards $(R_t)_{t \in 0,1,\ldots,T-1}$ are random variables distributed according to the reward function r: $\arg\max_{\pi} \mathbb{E}\{\sum_{t=0}^{T-1} \gamma^t R_t \mid \pi\}$.

4 Assumptions

The following assumptions are imposed on the LMMs employed in this work: (1) We assume that the LMMs are trained on diverse text and image corpora, providing a reasonable basis for generalization across embodied AI scenarios and tasks. (2) We assume that LMMs can process multiple images concurrently and follow textual prompts to perform reasoning. (3) MARS-RA is intended for

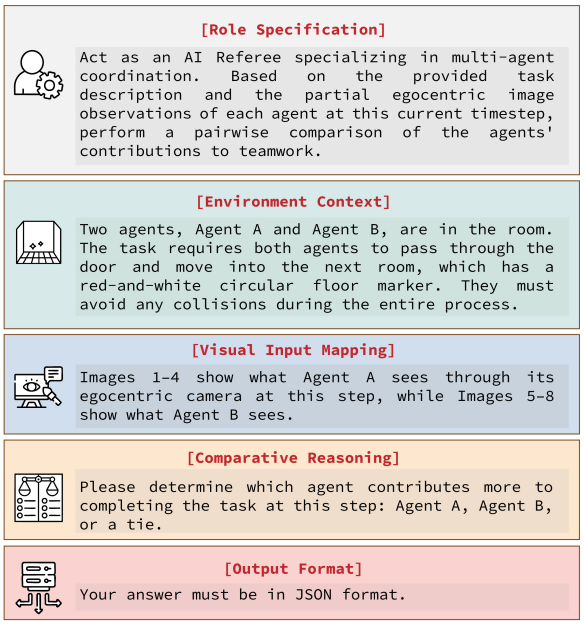


Figure 2: Example of the prompt used in the Pass Gate task. The prompt consists of five components: role specification, environmental context, visual input mapping, comparative reasoning, and output format.

tasks whose states can be evaluated using image-based observations and textual information.

5 Method

As shown in Figure 1, MARS-RA is a framework that integrates rank aggregation and LMMs to enable automatic credit assignment in MARL training. It consists of three steps: (1) LMM-based pairwise comparison, simplifying the contribution evaluation task into relative judgments, thereby providing the essential supervisory signal required for subsequent rank aggregation; (2) rank aggregation, serving to transform the discrete comparison matrix into continuous contribution credits, providing a fine-grained numeric basis for downstream MARL; and (3) potential function, transforming static contribution credits into dynamic shaping rewards, seamlessly embedding the ranking outcomes as dense rewards into the training loop.

5.1 LMM-based Pairwise Comparison.

Pairwise comparison is inherently independent of population size, making it an ideal mechanism for extracting reliable information in scenarios with a dynamic number of active agents. LMMs are especially well suited for performing these comparisons in MARS-RA: they can reason directly over high-dimensional visual observations that are difficult to evaluate using low-level states or hand-crafted rules, and they enable a fully automated assessment pipeline that naturally scales with compute-intensive MARL training.

During training, at each step t, we construct a pairwise comparison matrix $\boldsymbol{M}_t \in \mathbb{R}^{n\times n}$ initialized to zero, where n represents the total number of agents. We then query the LMM to perform pairwise comparisons only among the active agents in the set I^t. For every ordered pair (i, j) where $i \neq j$ and $i, j \in I^t$, the respective egocentric image observations o_t^i and o_t^j are provided as inputs to the LMM. Thus, each unordered agent pair corresponds to two ordered comparisons, which helps mitigate position bias (Tian et al., 2025) in LMM-based judgments. The resulting preferences (win for i, win for j, or tie) are assigned to the corresponding entries in $\boldsymbol{M}_t$, while entries involving inactive agents remain zero:

$$(\boldsymbol{M}_t)_{i,j} = f_{\text{LMM}}(o_t^i, o_t^j, \Theta), \tag{1}$$

where Θ denotes the textual prompt instructing the model to judge which agent contributes more to the team. $\boldsymbol{M}_t$ is structured as a comparison outcome matrix (2d array) where each entry $(\boldsymbol{M}_t)_{i,j}$ represents the number of times agent i is preferred over agent j. To handle ties, we assign a score of 0.5 to both $(\boldsymbol{M}_t)_{i,j}$ and $(\boldsymbol{M}_t)_{j,i}$ for each neutral judgment. It is important to note that, since pairwise comparison requires at least two entities, when the number of active agents satisfies $|I^t| < 2$, we simply skip the query process.

Figure 2 illustrates the structure of the prompt, which consists of five components: (1) role specification: assign the LMM the persona of a referee; (2) environmental context: supply the LMM with information about the task and its rules; (3) visual input mapping: indicate to the LMM the agent identity associated with each image; (4) comparative reasoning: instruct the LMM to determine which agent made a greater contribution to the team's success; (5) output format: enforce that the LMM outputs a format that is easy to parse (e.g., JSON).

5.2 Rank Aggregation.

We adopt the Bradley–Terry model (Bradley and Terry, 1952) to synthesize local pairwise comparisons into global contribution scores. This probabilistic framework serves two critical purposes: First, it performs Maximum Likelihood Estimation (MLE) (Bishop and Nasrabadi, 2006) aggregation under a scalar latent-score model, which can

mitigate noise or inconsistency by best-fit ranking. Second, it transforms ordinal preferences into continuous cardinal values, capturing not just the rank order but the magnitude of capability differences between agents. This fine-grained quantification is essential for generating smooth and informative potential-based rewards.

We fit the Bradley–Terry model to infer a latent contribution score $\boldsymbol{c}_t \in \mathbb{R}^{|I^t|}$ for all active agents at step t. The probability that agent i is preferred over agent j is modeled as:

$$\mathbb{P}(i \succ j) = \frac{\exp(c_t^i)}{\exp(c_t^i) + \exp(c_t^j)}. \quad (2)$$

Given $\boldsymbol{M}_t$, we estimate $\boldsymbol{c}_t$ by minimizing the negative log-likelihood:

$$\hat{\boldsymbol{c}}_t = \arg\min_{\boldsymbol{c}_t} \sum_{i,j} [-(\boldsymbol{M}_t)_{i,j} \log \mathbb{P}(i \succ j)] \quad (3)$$

The resulting estimate $\hat{\boldsymbol{c}}_t$ serves as the credit assigned to each agent at step t.

5.3 Potential Function.

To incorporate the derived contribution scores without biasing the optimization objective, we adopt potential-based reward shaping (PBRS) (Ng et al., 1999). This is crucial because the contribution scores evolve during training. Prior work shows that PBRS preserves policy invariance and Nash equilibria even under dynamic shaping (Devlin and Kudenko, 2012), and can safely embed arbitrary rewards by expressing them as dynamic potentials (Harutyunyan et al., 2015).

At this point, the reward function becomes $r : \mathcal{S} \times \mathcal{A}^I \times \mathcal{S} \to \mathbb{R}$, with $r(s, \boldsymbol{a}^I, s')$ denoting the reward received when coalition I executes the joint action $\boldsymbol{a}^I$ in state s and transitions to the next state s'. And the shaping reward function in this mechanism is defined as $F(s, t, s', t') = \gamma\,\psi(s', t') - \psi(s, t)$, where t denotes the time at which the agent was in the previous state s, and t' is the time when it reaches the current state s', with $t < t'$. We define our potential function as:

$$\psi(s_t, t) = \begin{cases} 0, & \text{if } s_t \text{ is terminal,} \\ \text{softmax}(\hat{\boldsymbol{c}}_t), & \text{otherwise.} \end{cases} \quad (4)$$

This assigns a zero potential to the terminal state (Wierstra et al., 2008), and applies softmax normalization at all non-terminal steps to map the contribution scores into a normalized representation of relative strength. We employ the following shaping reward under this mechanism: $F(s_t, t, s_{t+1}, t+1) = \gamma\psi(s_{t+1}, t+1) - \psi(s_t, t)$. The final shaped reward used for training is:

$$\tilde{\boldsymbol{r}}_t = r(s_t, \boldsymbol{a}_t^I, s_{t+1}) + \rho F(s_t, t, s_{t+1}, t+1), \quad (5)$$

where ρ is a scalar weighting factor that modulates the contribution of the potential-based reward relative to the environmental reward.

6 Theoretical Properties

In this section, we provide the theoretical justification for MARS-RA. We analyze two key properties: (1) **Convergence and Robustness**, showing that our rank aggregation framework can recover agents' underlying contribution scores and effectively mitigate noise and inaccuracies in LMM-generated pairwise comparisons, including those induced by LMM hallucinations as well as by partial observability and long-horizon dependencies in embodied AI scenarios; and (2) **Shapley Values as an Interpretive Reference**, demonstrating the consistency of our derived scores with Shapley values (Shapley et al., 1953) under ideal conditions. We follow the notation established in the preceding two sections. The detailed proof is provided in the Appendix B.

6.1 Convergence and Robustness Analysis

We assume that the LMM possesses a latent ground-truth preference vector $\mathbf{c}_t^* \in \mathbb{R}^{|I^t|}$ that represents the relative contributions of the active agents at step t. The LMM does not output $\mathbf{c}_t^*$ directly; instead, it acts as a comparator that follows the Bradley–Terry model and uses Maximum Likelihood Estimation (MLE) to obtain an estimator $\hat{\mathbf{c}}_t$ that best explains the observed pairwise comparisons. We now characterize the error bound of this estimator.

Proposition 1 (Convergence and Robustness)**.** *Suppose the comparison graph formed by $\boldsymbol{M}_t$ is connected. Let $\hat{\mathbf{c}}_t$ be the MLE estimate derived from K pairwise comparisons. With probability at least $1 - n^{-2}$, the estimation error relative to the latent preference $\mathbf{c}_t^*$ satisfies:*

$$\frac{1}{\sqrt{n}} \|\hat{\mathbf{c}}_t - \mathbf{c}_t^*\|_2 \leq C_0 \sqrt{\frac{n \log n}{K}} \quad (6)$$

where C_0 is a constant depending on the graph topology and $\|\cdot\|_2$ denotes the Euclidean norm.

Sketch of Proof. This result relies on the analysis of regularized MLE for pairwise comparisons (Negahban et al., 2012a). The Hessian of the negative log-likelihood behaves similarly to the Laplacian

of the comparison graph (Shah et al., 2016). Under the assumption of algebraic connectivity ($\lambda_2 > 0$), the objective function satisfies Restricted Strong Convexity (Negahban et al., 2012b). Combined with the concentration of measure for the gradient (bounded by $O(\sqrt{K \log n})$ via Hoeffding's inequality), standard convex optimization analysis yields the convergence rate of $O(1/\sqrt{K})$. □

Proposition 1 provides a mathematical justification for the convergence and robustness of our framework. It guarantees that as long as we perform sufficient pairwise comparisons (K), the aggregated scores $\hat{\mathbf{c}}_t$ will converge to the LMM's stable latent preference $\mathbf{c}_t^*$, thereby filtering out the influence of noise and inaccuracies.

6.2 Shapley Values as an Interpretive Reference

Having established that our method robustly recovers the LMM's latent preference $\mathbf{c}_t^*$, a natural question arises: *What does $\mathbf{c}_t^*$ represent conceptually?* We posit that a rational LMM, when provided with sufficient context, judges agents based on their marginal contributions.

Proposition 2 (Interpretability via Shapley Values)**.** *If the LMM acts as a rational probabilistic comparator where the latent preference is determined by the agents' true Shapley values (i.e., $\mathbf{c}_t^* = \mathbf{v}_t^*$), then the scores $\hat{\mathbf{c}}_t$ derived by MARS-RA are consistent estimators of the true Shapley values (up to a translation constant).*

This proposition serves as an interpretability bridge: it links the statistically robust scores derived in Section 6.1 to the game-theoretic concept of contribution assignment. This result is not intended to suggest that MARS-RA recovers true Shapley values in practice, but rather to provide an interpretive lens for understanding the semantics of the aggregated scores under idealized assumptions.

7 MARS-Bench

Existing embodied AI benchmarks are limited by their fixed-agent settings, failing to account for scenarios with a dynamic number of active agents. We implement a challenging set of embodied cooperative tasks with 2 to 4 agents in ManiSkill3 using XLeRobot (Wang and Lu, 2025), and construct MARS-Bench, as illustrated in Figure 3. It uses agents' egocentric camera views as pixel-based observations and adopts a discrete action space.

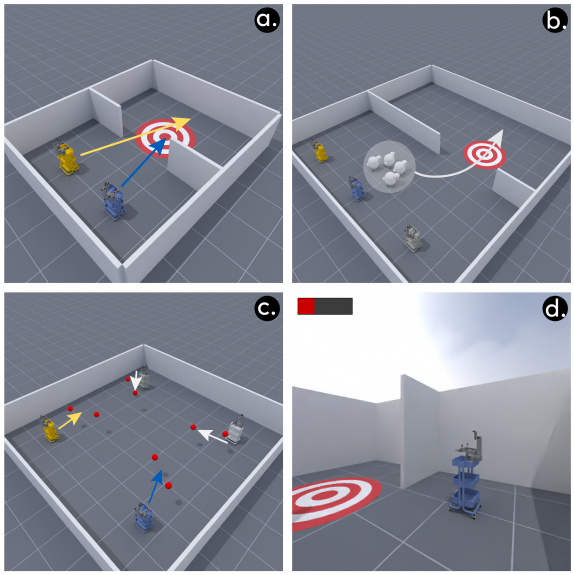
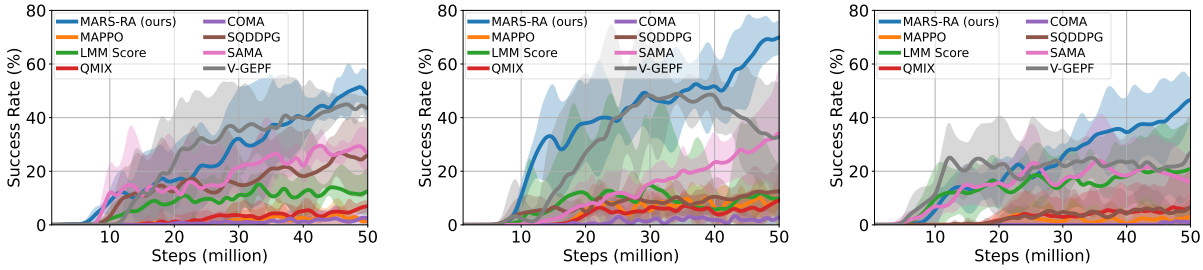


Figure 3: (a–c) Visualizations of the Pass Gate, Herd Sheep, and Collect Ball tasks in MARS-Bench, respectively. Colored arrows depict the intended movements of agents and objects. (d) An example of an agent's egocentric observation. The icon in the upper-left corner denotes the agent's current energy level; once depleted, the agent is removed from the environment and re-enters with full energy after a random number of steps.

Two reward modes are provided: sparse rewards for training to closely reflect real-world conditions, and dense rewards for analytical evaluation and debugging. At the start of each episode, agents are assigned random initial battery levels, which are depleted by actions. Agents with depleted batteries are temporarily removed and later respawn with full charge after a random delay, introducing agent-number openness. It consists of three tasks: **Pass Gate**, two agents are required to traverse a doorway between rooms without collisions; **Herd Sheep**, three agents cooperatively herd sheep from one room to another, where the sheep follow predefined movement dynamics; **Collect Ball**, four agents collect all red balls in the room. These tasks correspond to three fundamental multi-agent cooperation scenarios, namely spatio-temporal movement, cooperation, and divide and conquer (Wu et al., 2021). See Appendix A for more details.

8 Experiment

We evaluate MARS-RA on MARS-Bench, using the task success rate as the primary metric. Specifically, success is defined by task completion, whereas collisions and timeouts are failures. Beyond MARS-Bench, we further evaluate the gen-



(a) Pass Gate (b) Herd Sheep (c) Collect Ball

Figure 4: Learning curves of all compared methods on the three tasks in MARS-RA, trained for 50 million environment steps. Results are averaged over five random seeds, and error bars indicate 95% confidence intervals.

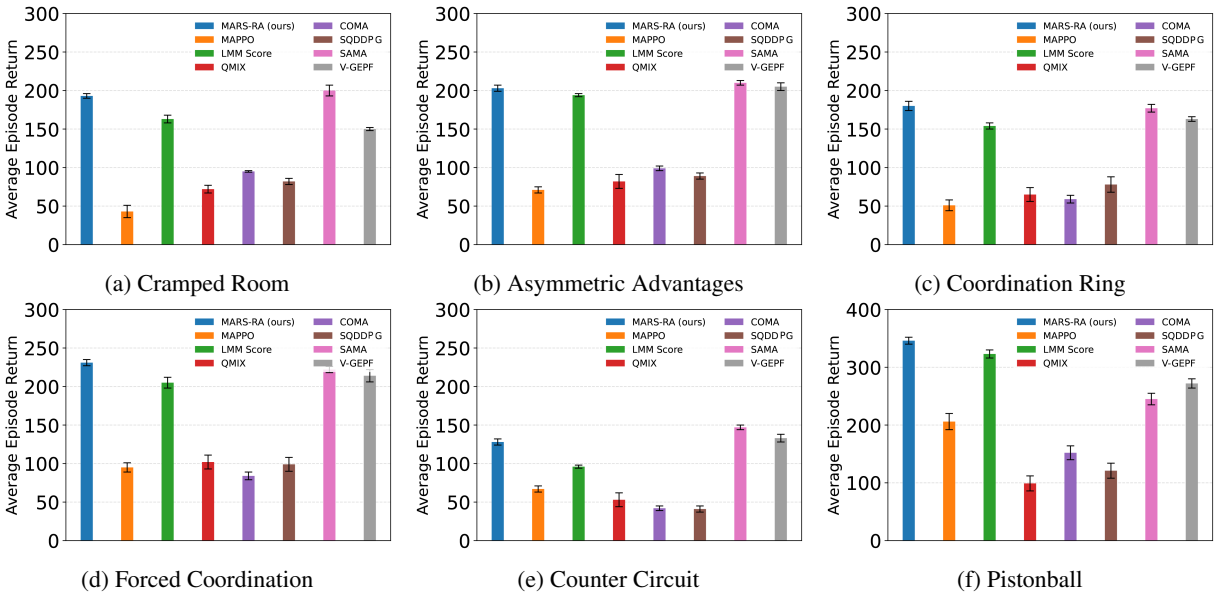
eralization of MARS-RA on Overcooked (Carroll et al., 2019) and Pistonball (Terry et al., 2021), with detailed descriptions of both environments provided in Appendix D. MARS-RA adopts MAPPO (Yu et al., 2022) as the backbone algorithm, while the required pairwise comparisons are generated by Gemini-2.5-Pro (Comanici et al., 2025), with a single query per comparison. For comparison, we include MAPPO and LMM Score as ablation variants alongside the baseline methods. See Appendix C for experiment details.

- **MAPPO**. It is a widely used policy-gradient algorithm for cooperative MARL and is adopted as the backbone of our method.
- **LMM Score**. This baseline queries the LMM using the task description and all agents' observations, generating per-agent contribution scores in $[0, 1]$ that are incorporated into learning via potential-based reward shaping.
- **QMIX** (Rashid et al., 2020), **COMA** (Foerster et al., 2017) and **SQDDPG** (Wang et al., 2020). These widely used methods address the credit assignment problem through value decomposition, counterfactual advantage estimation, and Shapley value approximation, respectively.
- **SAMA** (Li et al., 2025b). This is a subgoal-based framework designed to address the credit assignment problem in cooperative MARL. It leverages the commonsense priors embedded in LMMs to guide agent coordination, using MAPPO as its underlying backbone algorithm.
- **V-GEPF** (Ma et al., 2025). It is a hierarchical reward-shaping framework built on MAPPO for cooperative MARL. It employs a potential function derived from a vision language model for semantic guidance, alongside a LMM that selects cooperative skills from a predefined pool.

8.1 Overall Performance on MARS-Bench

Figure 4 displays the learning curves for all baselines. MARS-RA surpasses all baselines on three tasks, maintaining success rates above 47%. In particular, it attains a success rate of exceeding 70% on the highly collaborative Herd Sheep task. These results indicate that MARS-RA guides effective policy learning across embodied AI cooperative tasks involving 2 to 4 agents.

All selected baselines obtain success rates below 50% across the three MARS-Bench tasks, highlighting the challenging nature of the benchmark. QMIX and COMA do not exhibit meaningful cooperative behavior during training. SQDDPG shows stronger performance in the Pass Gate task with fewer agents, but its effectiveness diminishes in tasks with more agents, likely due to increased estimation error in Shapley-based credit assignment. Both SAMA and V-GEPF, as state-of-the-art methods, achieve competitive performance but remain consistently inferior to MARS-RA. This performance gap may partly arise from the design assumptions underlying these approaches. Specifically, SAMA's subgoal formulation is better suited to environments with clearly discretized task structures, whereas embodied AI tasks often lack such explicit decompositions. In contrast, V-GEPF relies on an Adaptive Skill Selection module that assumes timely and informative environmental feedback, while embodied AI tasks typically exhibit delayed feedback and long-horizon dependencies. In this experiment, we computed a 3-cycle rate of 8.95%, which is substantially lower than the theoretical random baseline of 25.00%. In addition, the held-out 3-way NLL (win/loss/tie) is 0.41, substantially below the random baseline of 1.10.



(a) Cramped Room (b) Asymmetric Advantages (c) Coordination Ring

(d) Forced Coordination (e) Counter Circuit (f) Pistonball

Figure 5: Performance comparison on five Overcooked tasks and Pistonball after training for 1 million environment steps, averaged over 10 random seeds, with standard error.

8.2 Results on Overcooked and Pistonball

Figures 5 (a–e) show the evaluation results of the baselines after training on the five Overcooked tasks. MARS-RA achieves performance comparable to the state-of-the-art method SAMA in the Overcooked environment, and outperforms SAMA on the Coordination Ring and Forced Coordination tasks. These results indicate that MARS-RA exhibits stable performance in standard MARL settings such as Overcooked. We also observe that the LMM Score variant performs comparably to MARS-RA on these five tasks, further confirming that its effectiveness improves on tasks with explicit objectives. Figure 5 (f) illustrates the evaluation performance of the baselines following training on the Pistonball task which involves 10 agents. MARS-RA consistently outperforms all compared baselines, suggesting that the proposed rank-aggregation formulation remains effective beyond small team sizes. We observe that SAMA and V-GEPF achieve relatively modest performance, which may be attributed to the simplicity of the Pistonball task, where there is limited scope for subtask decomposition or the exploitation of more complex strategies.

8.3 Ablation Study

As shown in Figure 4, compared to MARS-RA, the backbone algorithm MAPPO alone shows limited ability to learn effective cooperative policies across the three tasks. In contrast, the LMM Score variant learns cooperative behaviors to a limited extent but exhibits substantial performance oscillations, with inferior overall performance. This suggests that the spatio-temporal inference capabilities of LMMs can inform MARL training, whereas direct scoring is less effective than rank aggregation in enabling this translation. Notably, on the Collect Ball task with its explicit objectives, the LMM Score variant achieves relatively competitive performance.

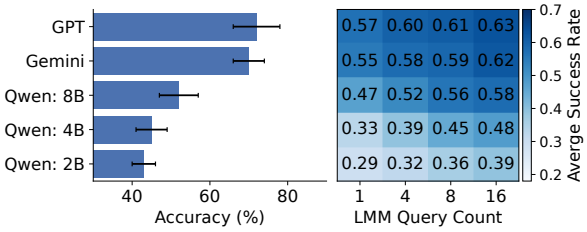


Figure 6: Left: Pairwise comparison accuracy of the selected LMMs, where GPT refers to GPT-5.1, Gemini to Gemini-2.5-Pro, and Qwen to Qwen3-VL. Right: Average success rates of MARS-RA on MARS-Bench under different LMMs and query counts.

8.4 Further Analysis

Accuracy of LMM-based Pairwise Comparisons. We further analyze the accuracy of LMM-based pairwise comparisons across a range of popular commercial and open-source LMMs, including Gemini-2.5-Pro, GPT-5.1 (OpenAI, 2025), and Qwen3-VL (2B, 4B, and 8B) (Yang et al., 2025). Accuracy is calculated by comparing the LMMs'

pairwise comparison results with the ground truth defined by the dense reward function of MARS-Bench, averaged over the three benchmark tasks. As shown in the left panel of Figure 6, both Gemini-2.5-Pro and GPT-5.1 achieve accuracy levels above 70%, while the three smaller-scale Qwen3-VL models also reach accuracies exceeding 43%. When an agent's egocentric observation is severely limited and lacks sufficient visual information (e.g., teammate positions or goal locations) to assess team contribution, our evaluation shows that LMMs can still make correct judgments as long as at least one agent's observation contains informative visual cues. Pairwise comparison errors most frequently occur when all agents face a wall and lack informative visual reference cues. See Appendix E for more details.

Model Selection and Number of Queries. We investigate the impact of LMM selection with different pairwise comparison accuracies and the number of LMM queries used for pairwise comparisons on MARS-RA performance. We train MARS-RA on the three MARS-Bench tasks using Gemini-2.5-Pro, GPT-5.1, and Qwen3-VL (2B, 4B, and 8B) under different numbers of pairwise comparison queries. The average success rates across the three tasks for each configuration are reported in the right panel of Figure 6. We observe that both employing LMMs with higher pairwise comparison accuracy and increasing the number of pairwise comparison queries lead to improved MARS-RA performance. However, when using high-accuracy LMMs, the performance gains from increasing the query count are less pronounced compared to those achieved with lower-accuracy LMMs. These results suggest a trade-off and complementary relationship between LMM pairwise comparison accuracy and the number of comparison queries. Further details can be found in Appendix E.

Comparison of Rank Aggregation Methods. We implemented Rank Centrality (Negahban et al., 2012a) as an alternative rank aggregation method under the same experimental settings as in this section, and report the results in Table 1. Each value represents the success rate on the corresponding MARS-Bench task achieved by the final trained model. The experimental results show that MARS-RA with Rank Centrality performs slightly worse than with Bradley–Terry model, although the performance gap is small.

Real-World Validation. We conduct a real-world validation of MARS-RA, as illustrated in Figure 7.

Method	Pass Gate	Herd Sheep	Collect Ball
RC	0.50 ± 0.03	0.67 ± 0.02	0.46 ± 0.02
BT	0.52 ± 0.01	0.68 ± 0.03	0.46 ± 0.02

Table 1: Comparison of different rank aggregation methods based on average performance across different levels in MARS-Bench. RC denotes Rank Centrality, and BT denotes the Bradley–Terry model.

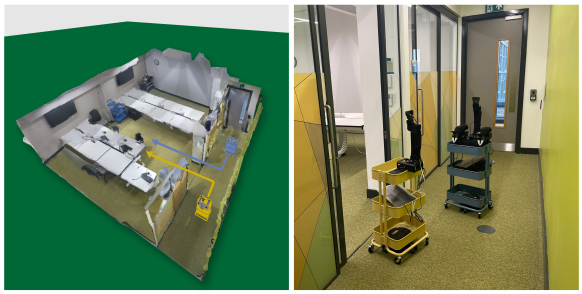


Figure 7: Left: A real-world 3D-scanned indoor environment instantiated as a MARS-Bench Pass Gate task, where two robots enter the room from a corridor without collisions. Right: Real-world deployment after virtual-environment training.

We deploy two XLeRobot robots to perform the Pass Gate task in a real-world indoor environment. The entire room is first captured via 3D scanning and reconstructed as a virtual 3D environment, which is then instantiated as a task in MARS-Bench. MARS-RA is trained in simulation and then deployed on real robots for evaluation in a physical environment, achieving a success rate of 64% over 25 runs. This result preliminarily suggests the potential of MARS-RA to guide agents toward effective cooperative policies in scenarios with real-world–level complexity. Additional details are provided in Appendix F.

9 Conclusion

In this paper, we propose MARS-RA, a framework that addresses the credit assignment problem in MARL for cooperative embodied AI systems through a rank aggregation perspective. The framework outperforms strong baselines across different tasks. Beyond empirical validation, we present theoretical analysis to substantiate the proposed framework. This work suggests a new direction for incorporating prior knowledge from foundation models into MARL training.

10 Limitations

There exist several avenues for improving this work and mitigating the limitations discussed below: (1) MARS-RA depends on LMMs for pairwise agent comparisons. While rank aggregation helps mitigate noise in these comparisons, the accuracy of LMM-generated judgments remains an important factor affecting overall performance. In MARS-RA, ordered pairwise comparisons are employed to alleviate position bias, and future work will investigate additional intermediate mechanisms to further reduce this reliance. (2) MARS-RA is designed for cooperative tasks whose states and outcomes can be reasonably assessed through visual observations and textual task descriptions. Tasks that rely on fine-grained physical signals, internal states, or domain-specific metrics that are not visually observable may fall outside the current scope of the framework. Future work will explore extensions or alternative formulations better suited to such tasks. (3) Non-stationarity is inherent in MARL, and the pairwise comparisons in MARS-RA do not eliminate this issue. However, the denoising effect of rank aggregation mitigates the impact of non-stationarity on the resulting contribution credits. Moreover, the potential-based reward shaping mechanism ensures that introducing these credits into MARL training preserves policy invariance under stationary transitions. In future work, we will continue to explore new approaches to address non-stationarity.

References

Alireza Saleh Abadi and Leen-Kiat Soh. 2025. Challenges in credit assignment for multi-agent reinforcement learning in open agent systems. *arXiv preprint arXiv:2510.27659*.

Anatoly Bakushinsky and A Goncharsky. 2012. *Ill-posed problems: theory and applications*, volume 301. Springer Science & Business Media.

Christopher M Bishop and Nasser M Nasrabadi. 2006. *Pattern recognition and machine learning*, volume 4. Springer.

Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

Rodney A Brooks. 1991. New approaches to robotics. *Science*, 253(5025):1227–1232.

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

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

Andy Clark. 1998. *Being there: Putting brain, body, and world together again*. MIT press.

Jonathan Cohen, Jilles-Steeve Dibangoye, and Abdel-Illah Mouaddib. 2017. Open decentralized pomdps. In *2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 977–984. IEEE.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Douglas E Critchlow, Michael A Fligner, and Joseph S Verducci. 1991. Probability models on rankings. *Journal of mathematical psychology*, 35(3):294–318.

Gerard Debreu. 1960. Individual choice behavior: A theoretical analysis.

Sam Michael Devlin and Daniel Kudenko. 2012. Dynamic potential-based reward shaping. In *11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, pages 433–440. IFAAMAS.

Jingfeng Zhang Di Zhao, Hongsheng Hu, Philippe Fournier-Viger, Gillian Dobbie, and Yun Sing Koh. 2025. Balancing invariant and specific knowledge for domain generalization with online knowledge distillation. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, pages 2440–2448.

Shuai Dong, Siyuan Wang, Xingyu Liu, Chenglin Li, Haowen Hou, and Zhongyu Wei. 2025. Interleaved latent visual reasoning with selective perceptual modeling. *arXiv preprint arXiv:2512.05665*.

Tongtong Feng, Xin Wang, Yu-Gang Jiang, and Wenwu Zhu. 2025. Embodied ai: From llms to world models. *arXiv preprint arXiv:2509.20021*.

Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. 2017. Counterfactual multi-agent policy gradients (2017). *arXiv preprint arXiv:1705.08926*.

Anna Harutyunyan, Sam Devlin, Peter Vrancx, and Ann Nowé. 2015. Expressing arbitrary reward functions as potential-based advice. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29.

Ralf Herbrich, Tom Minka, and Thore Graepel. 2006. Trueskill™: a bayesian skill rating system. *Advances in neural information processing systems*, 19.

Shaohan Huang, Li Dong, Wenhui Wang, Yaru Hao, Saksham Singhal, Shuming Ma, Tengchao Lv, Lei Cui, Owais Khan Mohammed, Barun Patra, and 1 others. 2023. Language is not all you need: Aligning perception with language models. *Advances in Neural Information Processing Systems*, 36:72096–72109.

Raivo Kolde, Sven Laur, Priit Adler, and Jaak Vilo. 2012. Robust rank aggregation for gene list integration and meta-analysis. *Bioinformatics*, 28(4):573–580.

Chenglin Li, Qianglong Chen, Feng Han, Yikun Wang, Xingxi Yin, Yan Gong, Ruilin Li, Yin Zhang, and Jiaqi Wang. 2026. Videothinker: Building agentic videollms with llm-guided tool reasoning. *arXiv preprint arXiv:2601.15724*.

Chenglin Li, Feng Han, Yikun Wang, Ruilin Li, Shuai Dong, Haowen Hou, Haitao Li, Qianglong Chen, Feng Tao, Jingqi Tong, and 1 others. 2025a. Videopro: Adaptive program reasoning for long video understanding. *arXiv preprint arXiv:2509.17743*.

Wenhao Li, Dan Qiao, Baoxiang Wang, Xiangfeng Wang, Wei Yin, Hao Shen, Bo Jin, and Hongyuan Zha. 2025b. Multi-agent credit assignment with pretrained language models. In *International Conference on Artificial Intelligence and Statistics*, pages 1945–1953. PMLR.

Yunxin Li, Baotian Hu, Xinyu Chen, Lin Ma, Yong Xu, and Min Zhang. 2024. Lmeye: An interactive perception network for large language models. *IEEE Transactions on Multimedia*.

Muhan Lin, Shuyang Shi, Yue Guo, Vaishnav Tadiparthi, Behdad Chalaki, Ehsan Moradi Pari, Simon Stepputtis, Woojun Kim, Joseph Campbell, and Katia Sycara. 2025. Speaking the language of teamwork: Llm-guided credit assignment in multi-agent reinforcement learning. *arXiv preprint arXiv:2502.03723*.

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

Wenzhang Liu, Wenzhe Cai, Kun Jiang, Guangran Cheng, Yuanda Wang, Jiawei Wang, Jingyu Cao, Lele Xu, Chaoxu Mu, and Changyin Sun. 2023b. Xuance: A comprehensive and unified deep reinforcement learning library. *arXiv preprint arXiv:2312.16248*.

Hao Ma, Shijie Wang, Zhiqiang Pu, Siyao Zhao, and Xiaolin Ai. 2025. Vision-based generic potential function for policy alignment in multi-agent reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 19287–19295.

Ke Ma, Qianqian Xu, Jinshan Zeng, Guorong Li, Xiaochun Cao, and Qingming Huang. 2022. A tale of hodgerank and spectral method: Target attack against rank aggregation is the fixed point of adversarial game. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4090–4108.

Marvin Minsky. 2007. Steps toward artificial intelligence. *Proceedings of the IRE*, 49(1):8–30.

Kartik Nagpal, Dayi Dong, Jean-Baptiste Bouvier, and Negar Mehr. 2025. Leveraging large language models for effective and explainable multi-agent credit assignment. *arXiv preprint arXiv:2502.16863*.

Sahand Negahban, Sewoong Oh, and Devavrat Shah. 2012a. Iterative ranking from pair-wise comparisons. *Advances in neural information processing systems*, 25.

Sahand N Negahban, Pradeep Ravikumar, Martin J Wainwright, and Bin Yu. 2012b. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. *Statistical Science*, 27(4):538–557.

Andrew Y Ng, Daishi Harada, and Stuart Russell. 1999. Policy invariance under reward transformations: Theory and application to reward shaping. In *Icml*, volume 99, pages 278–287. Citeseer.

OpenAI. 2025. Gpt-5 system card. https://cdn.openai.com/gpt-5-system-card.pdf.

Georgios Papoudakis, Filippos Christianos, Lukas Schäfer, and Stefano V Albrecht. 2020. Benchmarking multi-agent deep reinforcement learning algorithms in cooperative tasks. *arXiv preprint arXiv:2006.07869*.

Dipam Patel, Phu Pham, Kshitij Tiwari, and Aniket Bera. 2023. Dream: Decentralized reinforcement learning for exploration and efficient energy management in multi-robot systems. *arXiv preprint arXiv:2309.17433*.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. 2020. Monotonic value function factorisation for deep multi-agent reinforcement learning. *Journal of Machine Learning Research*, 21(178):1–51.

Juan Rocamonde, Victoriano Montesinos, Elvis Nava, Ethan Perez, and David Lindner. 2023. Vision-language models are zero-shot reward models for reinforcement learning. *arXiv preprint arXiv:2310.12921*.

Álvaro Serra-Gómez, Hai Zhu, Bruno Brito, Wendelin Böhmer, and Javier Alonso-Mora. 2023. Learning scalable and efficient communication policies for multi-robot collision avoidance. *Autonomous Robots*, 47(8):1275–1297.

Nihar B Shah, Sivaraman Balakrishnan, Joseph Bradley, Abhay Parekh, Kannan Ramchandran, and Martin J Wainwright. 2016. Estimation from pairwise comparisons: Sharp minimax bounds with topology dependence. *Journal of Machine Learning Research*, 17(58):1–47.

Lloyd S Shapley and 1 others. 1953. A value for n-person games.

Lin Shi, Chiyu Ma, Wenhua Liang, Xingjian Diao, Weicheng Ma, and Soroush Vosoughi. 2024. Judging the judges: A systematic study of position bias in llm-as-a-judge. *arXiv preprint arXiv:2406.07791*.

Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, and 1 others. 2017. Value-decomposition networks for cooperative multi-agent learning. *arXiv preprint arXiv:1706.05296*.

Xuting Tang, Jia Xu, and Shusen Wang. 2023. Roma: Resilient multi-agent reinforcement learning with dynamic participating agents. In *2023 IEEE 12th International Conference on Cloud Networking (CloudNet)*, pages 247–255. IEEE.

Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse-kai Chan, and 1 others. 2024. Maniskill3: Gpu parallelized robotics simulation and rendering for generalizable embodied ai. *arXiv preprint arXiv:2410.00425*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

J Terry, Benjamin Black, Nathaniel Grammel, Mario Jayakumar, Ananth Hari, Ryan Sullivan, Luis S Santos, Clemens Dieffendahl, Caroline Horsch, Rodrigo Perez-Vicente, and 1 others. 2021. Pettingzoo: Gym for multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 34:15032–15043.

Xinyu Tian, Shu Zou, Zhaoyuan Yang, and Jing Zhang. 2025. Identifying and mitigating position bias of multi-image vision-language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10599–10609.

Alan M Turing. 2021. Computing machinery and intelligence (1950). *Mind*, 59(236):33–60.

Dawei Wang, Chengming Zhou, Di Zhao, Xinyuan Liu, Marci Chi Ma, Gary Ushaw, and Richard Davison. 2026. Towermind: A tower defence game learning environment and benchmark for llm as agents. *arXiv preprint arXiv:2601.05899*.

Gaotian Wang and Zhuoyi Lu. 2025. Xlerobot: A practical low-cost household dual-arm mobile robot design for general manipulation. https://github.com/Vector-Wangel/XLeRobot.

Jianhong Wang, Yuan Zhang, Yunjie Gu, and Tae-Kyun Kim. 2022. Shaq: Incorporating shapley value theory into multi-agent q-learning. *Advances in Neural Information Processing Systems*, 35:5941–5954.

Jianhong Wang, Yuan Zhang, Tae-Kyun Kim, and Yunjie Gu. 2020. Shapley q-value: A local reward approach to solve global reward games. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7285–7292.

Yuan Wei, Xiaohan Shan, Ran Miao, and Jianmin Li. 2025. Lero: Llm-driven evolutionary framework with hybrid rewards and enhanced observation for multi-agent reinforcement learning. In *International Conference on Intelligent Computing*, pages 15–26. Springer.

Daan Wierstra, Tom Schaul, Jan Peters, and Juergen Schmidhuber. 2008. Episodic reinforcement learning by logistic reward-weighted regression. In *International Conference on Artificial Neural Networks*, pages 407–416. Springer.

Annie Wong, Thomas Bäck, Anna V Kononova, and Aske Plaat. 2023. Deep multiagent reinforcement learning: Challenges and directions. *Artificial Intelligence Review*, 56(6):5023–5056.

Sarah A Wu, Rose E Wang, James A Evans, Joshua B Tenenbaum, David C Parkes, and Max Kleiman-Weiner. 2021. Too many cooks: Bayesian inference for coordinating multi-agent collaboration. *Topics in Cognitive Science*, 13(2):414–432.

Ziyan Xiong, Bo Chen, Shiyu Huang, Wei-Wei Tu, Zhaofeng He, and Yang Gao. 2024. Mqe: Unleashing the power of interaction with multi-agent quadruped environment. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5918–5924. IEEE.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

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

Hongxin Zhang, Zeyuan Wang, Qiushi Lyu, Zheyuan Zhang, Sunli Chen, Tianmin Shu, Behzad Dariush, Kwonjoon Lee, Yilun Du, and Chuang Gan. 2024. Combo: compositional world models for embodied multi-agent cooperation. *arXiv preprint arXiv:2404.10775*.

Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. 2021. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of reinforcement learning and control*, pages 321–384.

Di Zhao, Yun Sing Koh, Gillian Dobbie, Hongsheng Hu, and Philippe Fournier-Viger. 2024. Symmetric self-paced learning for domain generalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 16961–16969.

Di Zhao, Jingfeng Zhang, Hongsheng Hu, Philippe Fournier-Viger, Gillian Dobbie, and Yun Sing Koh. 2026. Unlearning during training: Domain-specific gradient ascent for domain generalization. In *The Fourteenth International Conference on Learning Representations*.

A Additional Details of MARS-Bench

Each agent receives an egocentric pixel-based observation. The action space is discrete and consists of five actions: *no-operation*, *move forward*, *move backward*, *turn left*, and *turn right*. The reward signal is defined under two settings: a sparse and a dense mode. In the sparse reward setting, the agent receives a reward of $+1$ upon task completion and a penalty of -1 for collisions or timeouts. In the dense reward setting, agents receive per-agent, step-wise shaping rewards that reflect progress toward task completion. The sparse reward mode is designed to closely reflect real-world conditions during training. In contrast, the dense reward mode leverages extensive environment-internal variables that would be unavailable in real-world settings, and is used solely for analysis and debugging purposes. Each episode has a maximum length of 2000 steps and terminates immediately upon either success or failure. Below we provide detailed descriptions of the three tasks:

- **Pass Gate**: This task involves 2 agents. It presents a bottleneck challenge in which a narrow gate divides the environment. Two agents starting on the same side must resolve contention for access to this shared passage, necessitating policies that can manage spatio-temporal conflicts.
- **Herd Sheep**: This task involves 3 agents. We introduce sheep as non-player characters whose autonomous behavior combines avoidance of agents, attraction to the flock's centroid, and stochastic wandering. The agents must collaborate to drive all sheep through the gate into the next room. This task tests the agents' ability to understand the sheep's movement dynamics and to cooperate in executing the herding task.
- **Collect Ball**: This task involves 4 agents. Agents must collect all balls in the room, aiming to maximize efficiency via parallel execution.

B Detailed Proofs of Theoretical Properties

In this appendix, we provide the step-by-step mathematical derivations for the theoretical properties presented in Section 6.

B.1 Proof of Proposition 1 (Convergence and Robustness)

Proposition Restatement: *Suppose the comparison graph is connected with algebraic connectivity* $\lambda_2 > 0$. *Let* $\hat{\mathbf{c}}_t$ *be the MLE estimate derived*

from K pairwise comparisons. With probability at least $1 - n^{-2}$, the estimation error satisfies $\|\hat{\mathbf{c}}_t - \mathbf{c}_t^*\|_2 \leq \mathcal{O}(\sqrt{\frac{n \log n}{K}})$.

Proof. We analyze the error bound using the framework of M-estimation in convex optimization. The proof proceeds in three main steps: (1) establishing the strong convexity of the loss function, (2) bounding the gradient at the optimum, and (3) deriving the parameter error bound.

Step 1: Negative Log-Likelihood and Convexity. The negative log-likelihood (loss function) for the Bradley–Terry model with K samples is:

$$\mathcal{L}(\mathbf{c}_t) = \sum_{k=1}^{K} \big[\log(1 + e^{c_{t,i_k} - c_{t,j_k}}) - y_k(c_{t,i_k} - c_{t,j_k})\big] \tag{7}$$

where $y_k = 1$ if $i_k \succ j_k$ and 0 otherwise. The Hessian matrix of the loss function, $\mathbf{H}(\mathbf{c}_t) = \nabla^2 \mathcal{L}(\mathbf{c}_t)$, corresponds to the Laplacian of the comparison graph weighted by the variances of the logistic distribution. For a connected graph, the smallest non-zero eigenvalue of the Laplacian, denoted as λ_2 (algebraic connectivity), is strictly positive. Thus, restricted to the subspace orthogonal to the constant vector $\mathbf{1}$ (to handle translation invariance), the loss function is μ-strongly convex locally around $\mathbf{c}_t^*$:

$$\Delta^T \nabla^2 \mathcal{L}(\mathbf{c}_t) \Delta \geq \mu \|\Delta\|_2^2, \quad \forall \Delta \in \mathbb{R}^n, \Delta \perp \mathbf{1} \tag{8}$$

where μ scales linearly with the number of samples K (assuming uniform sampling of pairs).

Step 2: Taylor Expansion. Since $\hat{\mathbf{c}}_t$ minimizes $\mathcal{L}(\mathbf{c}_t)$, the gradient $\nabla \mathcal{L}(\hat{\mathbf{c}}_t) = 0$. We verify the error $\Delta = \hat{\mathbf{c}}_t - \mathbf{c}_t^*$ using the first-order Taylor expansion of the gradient around $\mathbf{c}_t^*$:

$$\nabla \mathcal{L}(\hat{\mathbf{c}}_t) \approx \nabla \mathcal{L}(\mathbf{c}_t^*) + \nabla^2 \mathcal{L}(\mathbf{c}_t^*)(\hat{\mathbf{c}}_t - \mathbf{c}_t^*) \tag{9}$$

Setting $\nabla \mathcal{L}(\hat{\mathbf{c}}_t) = 0$ and multiplying by Δ^T:

$$0 \approx \Delta^T \nabla \mathcal{L}(\mathbf{c}_t^*) + \Delta^T \nabla^2 \mathcal{L}(\mathbf{c}_t^*) \Delta \tag{10}$$

Using the strong convexity property (Eq. 8) and the Cauchy-Schwarz inequality:

$$\begin{aligned} \mu \|\Delta\|_2^2 \leq \Delta^T \nabla^2 \mathcal{L}(\mathbf{c}_t^*) \Delta &= -\Delta^T \nabla \mathcal{L}(\mathbf{c}_t^*) \\ &\leq \|\Delta\|_2 \|\nabla \mathcal{L}(\mathbf{c}_t^*)\|_2 \end{aligned} \tag{11}$$

Dividing both sides by $\mu \|\Delta\|_2$:

$$\|\Delta\|_2 \leq \frac{1}{\mu} \|\nabla \mathcal{L}(\mathbf{c}_t^*)\|_2 \tag{12}$$

Step 3: Bounding the Gradient Norm (Concentration). The gradient at the true parameter $\mathbf{c}_t^*$ is given by:

$$\nabla \mathcal{L}(\mathbf{c}_t^*) = \sum_{k=1}^{K} (P^*_{i_k j_k} - y_k) \mathbf{x}_k \tag{13}$$

where $\mathbf{x}_k$ is the indicator vector for the pair (i_k, j_k). Since $E[y_k] = P^*_{i_k j_k}$, the expected gradient is $\mathbb{E}[\nabla \mathcal{L}(\mathbf{c}_t^*)] = \mathbf{0}$. The term $(P^* - y_k)$ is a bounded random variable in $[-1, 1]$. By Hoeffding's inequality (or Azuma-Hoeffding for martingales), the norm of the sum of these random variables is bounded with high probability. Specifically, for n dimensions:

$$\mathbb{P}\left(\|\nabla \mathcal{L}(\mathbf{c}_t^*)\|_2 \geq \tau\right) \leq 2n \exp\left(-\frac{\tau^2}{CK}\right) \tag{14}$$

Setting the probability to n^{-2}, we get the bound $\|\nabla \mathcal{L}(\mathbf{c}_t^*)\|_2 \leq \mathcal{O}(\sqrt{K \log n})$.

Final Assembly. Substituting the gradient bound ($\sqrt{K}$) and the strong convexity parameter ($\mu \propto K$) into Eq. (12):

$$\|\hat{\mathbf{c}}_t - \mathbf{c}_t^*\|_2 \leq \frac{\mathcal{O}(\sqrt{K \log n})}{\mathcal{O}(K)} = \mathcal{O}\left(\sqrt{\frac{\log n}{K}}\right) \tag{15}$$

Normalizing by dimension (as per the theorem statement) yields the stated result. □

B.2 Proof of Proposition 2 (Alignment with Shapley Values)

Proposition Restatement: *If $\mathbf{c}_t^* = \boldsymbol{v}_t^*$, then $\hat{\mathbf{c}}_t$ derived by MLE are consistent estimators of the true Shapley values.*

Proof. This proof relies on the consistency of Maximum Likelihood Estimators and the Law of Large Numbers.

The derivative of the log-likelihood function with respect to the score $c_{t,i}$ is:

$$\frac{\partial \ell}{\partial c_{t,i}} = \sum_{j \neq i} \left(n_{ij} - N_{ij} \frac{e^{c_{t,i}}}{e^{c_{t,i}} + e^{c_{t,j}}}\right) \tag{16}$$

where n_{ij} is the observed number of times i beats j, and N_{ij} is the total comparisons. The MLE solution $\hat{\mathbf{c}}_t$ satisfies $\frac{\partial \ell}{\partial c_{t,i}} = 0$ for all i.

Dividing the equation by the total number of samples K and taking the limit as $K \to \infty$:

$$\lim_{K \to \infty} \frac{n_{ij}}{N_{ij}} = \mathbb{P}(i \succ j \mid \mathbf{c}_t^*) = \frac{e^{c^*_{t,i}}}{e^{c^*_{t,i}} + e^{c^*_{t,j}}} \tag{17}$$

The optimality condition for the estimator $\hat{\mathbf{c}}_t$ in the limit becomes:

$$\sum_{j \neq i} \pi_{ij} \left(\frac{e^{c^*_{t,i}}}{e^{c^*_{t,i}} + e^{c^*_{t,j}}} - \frac{e^{\hat{c}_{t,i}}}{e^{\hat{c}_{t,i}} + e^{\hat{c}_{t,j}}} \right) = 0 \quad (18)$$

where π_{ij} is the sampling probability of pair (i, j). Since the logistic function $\sigma(x) = \frac{1}{1+e^{-x}}$ is strictly monotonic, the only solution to this system of equations (assuming a connected graph) implies:

$$\hat{c}_{t,i} - \hat{c}_{t,j} = c^*_{t,i} - c^*_{t,j}, \quad \forall i, j \quad (19)$$

Thus, $\hat{\mathbf{c}}_t = \mathbf{c}^*_t + C$. Since we assume $\mathbf{c}^*_t = \boldsymbol{v}^*$, the estimated scores converge to the Shapley values. □

C Experimental Details

We employ action repeat during training, where agents make decisions once every 32 environment steps. From each action-repeat window, we uniformly sample four visual frames, convert them to grayscale, and stack them to form a $128 \times 128 \times 4$ observation space for each agent. All training models in our experiments adopt a unified multi-layer CNN-based architecture, as illustrated in Figure 8. MAPPO, QMIX, and COMA are implemented using the XuanCe (Liu et al., 2023b) framework, while the implementations of SQDDPG, SAMA, and V-GEPF are obtained from their respective open-source code repositories. All MAPPO-based methods, including MARS-RA, MAPPO, LMM Score, SAMA, and V-GEPF, share the same MAPPO hyperparameters, as reported in Table 2. The hyperparameters for QMIX and COMA follow those used in Papoudakis et al. (2020), the hyperparameters for SQDDPG follow those used in Wang et al. (2022). For a fair comparison, SAMA and V-GEPF employ the same LMM as MARS-RA (Gemini-2.5-Pro), and all other configurations follow the original implementations.

All experiments were conducted on a desktop computer running Ubuntu 24.04, equipped with an Intel(R) Core(TM) i7-14700K CPU (20 cores), an NVIDIA GeForce RTX 3090 GPU with 24 GB of VRAM, and 128 GB of RAM. For GPT-5.1 and Gemini-2.5-Pro, we use the official API services provided on their respective websites. In contrast, Qwen3-VL (2B, 4B, and 8B) is deployed locally. To accelerate LMM-based pairwise comparisons during training, we deploy Qwen3-VL across 4 desktop machines, each equipped with

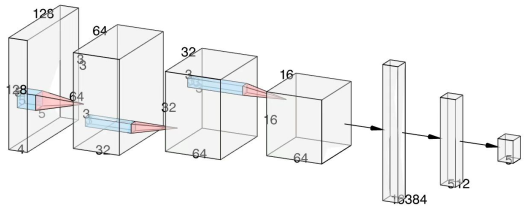


Figure 8: Model architectures used by all MARL algorithms in this study, a CNN followed by MLP.

an NVIDIA GeForce RTX 3090 GPU. Under our experimental setup and computational resources, training MAPPO for 50 million environment steps completes within 16 hours, while training MARS-RA for 50 million steps completes within 30 hours.

Moreover, we conduct an additional hyperparameter sweep over ρ on MARS-Bench to investigate how different ρ values affect the performance of MARS-RA. The results are reported in Table 3, where each value denotes the success rate achieved by the final trained model on the corresponding MARS-Bench task. The hyperparameter sweep results indicate that MARS-RA achieves its best performance when $\rho \in \{0.1, 0.5, 1.0\}$. When ρ takes a smaller value (e.g., 0.03) or a larger value (e.g., 5), the performance of MARS-RA declines and the standard error increases. When $\rho = 0$, MARS-RA reduces to MAPPO, and its performance is consistent with that of MAPPO.

Hyperparameter	**MAPPO**
Optimizer	Adam
Learning rate	0.001
Training Batch Size	10240
Minibatch Size	512
Discount factor (γ)	0.99
GAE (λ)	0.95
Policy clip ratio	0.20
Epochs	10

Table 2: Key hyperparameters for MAPPO in our experiments.

D Experiments on Overcooked and Pistonball

We evaluate MARS-RA and the selected baselines on the widely used MARL environments Overcooked (Carroll et al., 2019) and Pistonball (Terry et al., 2021). This evaluation aims to verify that

ρ	Pass Gate	Herd Sheep	Collect Ball
$\rho = 0$	0.02 ± 0.00	0.06 ± 0.02	0.03 ± 0.01
$\rho = 0.03$	0.35 ± 0.03	0.52 ± 0.04	0.33 ± 0.06
$\rho = 0.1$	0.50 ± 0.03	0.66 ± 0.02	0.44 ± 0.03
$\rho = 0.5$	0.53 ± 0.02	0.66 ± 0.01	0.47 ± 0.04
$\rho = 1.0$	0.52 ± 0.01	0.68 ± 0.03	0.46 ± 0.02
$\rho = 5.0$	0.40 ± 0.07	0.51 ± 0.05	0.39 ± 0.04

Table 3: Effect of different ρ values on performance across levels in MARS-Bench.

MARS-RA can guide agents to learn effective cooperative policies not only in embodied AI scenarios, but also in classic MARL tasks. We retain the original observation spaces, discrete action spaces, and reward functions provided by these environments, without introducing dynamically varying numbers of active agents or using action repeat. In both environments, models are trained for 1 million environment steps. All other experimental settings follow those in Section 8 and Appendix C.

D.1 Overcooked

In this kitchen cooking game, two agents must collaboratively prepare and deliver soup to obtain a shared team reward, as illustrated in Figure 9. In this environment, agents receive a team reward of 20 upon each successful soup delivery, and the episode return is used as the evaluation metric. It includes five tasks:

- **Cramped Room**: The setting is a restrictive room where two agents must share a single pot and serving point. The challenge encourages the agents to fully exploit the limited resources to cook and serve soup, achievable through simple cooperative strategies.
- **Asymmetric Advantages**: Players operate in two separate, isolated kitchens with an asymmetric layout. On the left side, onions are far from the pots, but serving points are centrally located. On the right side, the setup is reversed: onions are near the center, while serving points are far away.
- **Coordination Ring**: This circular design forces players to keep moving to avoid collisions, especially at the choke points in the top-right and bottom-left corners housing the ingredients and pots. Success depends on the simultaneous use of both pots.
- **Forced Coordination**: This layout creates a dependency loop by isolating the agents. Since the left side lacks cooking facilities and the right side lacks ingredients, the pair must verify their actions are synchronized. The workflow dictates that the left player prepares ingredients and plates, which allows the right player to complete the cooking and serving.
- **Counter Circuit**: This larger ring-shaped map places pots, ingredients, and serving stations on four different sides. Narrow paths make blocking frequent and teamwork difficult. A key strategy for success is placing onions on the central counters to allow for quick hand-offs between players.

D.2 Pistonball

In this physics-based cooperative environment, agents control vertically actuating pistons to propel a ball toward the left boundary, as illustrated in Figure 10. Developing an optimal policy for this task requires the agents to learn and execute highly coordinated joint behaviors. The environment contains 10 pistons, each of which is controlled by an individual agent. All agents receive a shared global reward at each timestep, consisting of a movement-based term and a fixed time penalty. The movement reward is calculated as the net displacement of the ball towards the left, normalized as a percentage of the total initial distance to the wall. Conversely, rightward movement incurs a negative reward. We evaluate performance using the episode return as the metric.

E Experimental Details on LMM-Based Pairwise Comparisons

E.1 Setup

We train MARS-RA on the three MARS-Bench tasks using Gemini-2.5-Pro, GPT-5.1, and Qwen3-VL (2B, 4B, and 8B) under varying numbers of pairwise comparison queries, and concurrently measure the accuracy of LMM-based pairwise comparisons. The accuracy of LMM-based pairwise comparisons is computed by measuring the agreement between LMM judgments generated during training and the corresponding per-agent dense reward signals of each task, and then averaging across comparisons. All other experimental settings follow Section 8 and Appendix C.

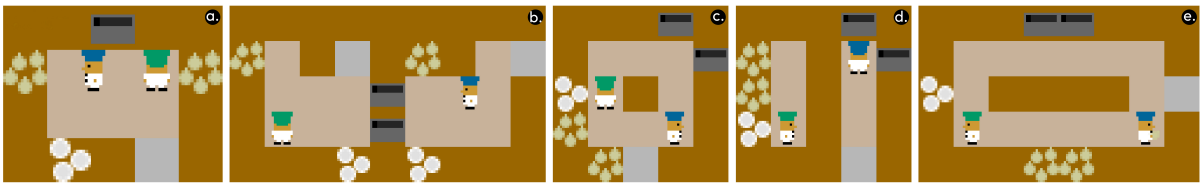


Figure 9: Screenshots of five tasks in the Overcooked environment: (a) Cramped Room, (b) Asymmetric Advantages, (c) Coordination Ring, (d) Forced Coordination, and (e) Counter Circuit.

LMMs	Pass Gate	Herd Sheep	Collect Ball	LMM Average
GPT-5.1	0.65 ± 0.09	0.83 ± 0.06	0.68 ± 0.04	0.72 ± 0.06
Gemini-2.5-Pro	0.63 ± 0.04	0.77 ± 0.10	0.70 ± 0.03	0.70 ± 0.04
Qwen3-VL: 8B	0.48 ± 0.04	0.62 ± 0.08	0.46 ± 0.15	0.52 ± 0.05
Qwen3-VL: 4B	0.42 ± 0.11	0.54 ± 0.02	0.40 ± 0.06	0.45 ± 0.04
Qwen3-VL: 2B	0.37 ± 0.18	0.49 ± 0.13	0.42 ± 0.10	0.43 ± 0.03
Task Average	0.51 ± 0.06	0.65 ± 0.06	0.53 ± 0.07	0.56 ± 0.06

Table 4: Pairwise comparison accuracy of different LMMs on the three MARS-Bench tasks, with standard error.

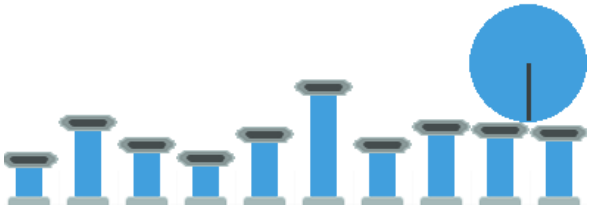


Figure 10: Screenshot of the Pistonball environment.

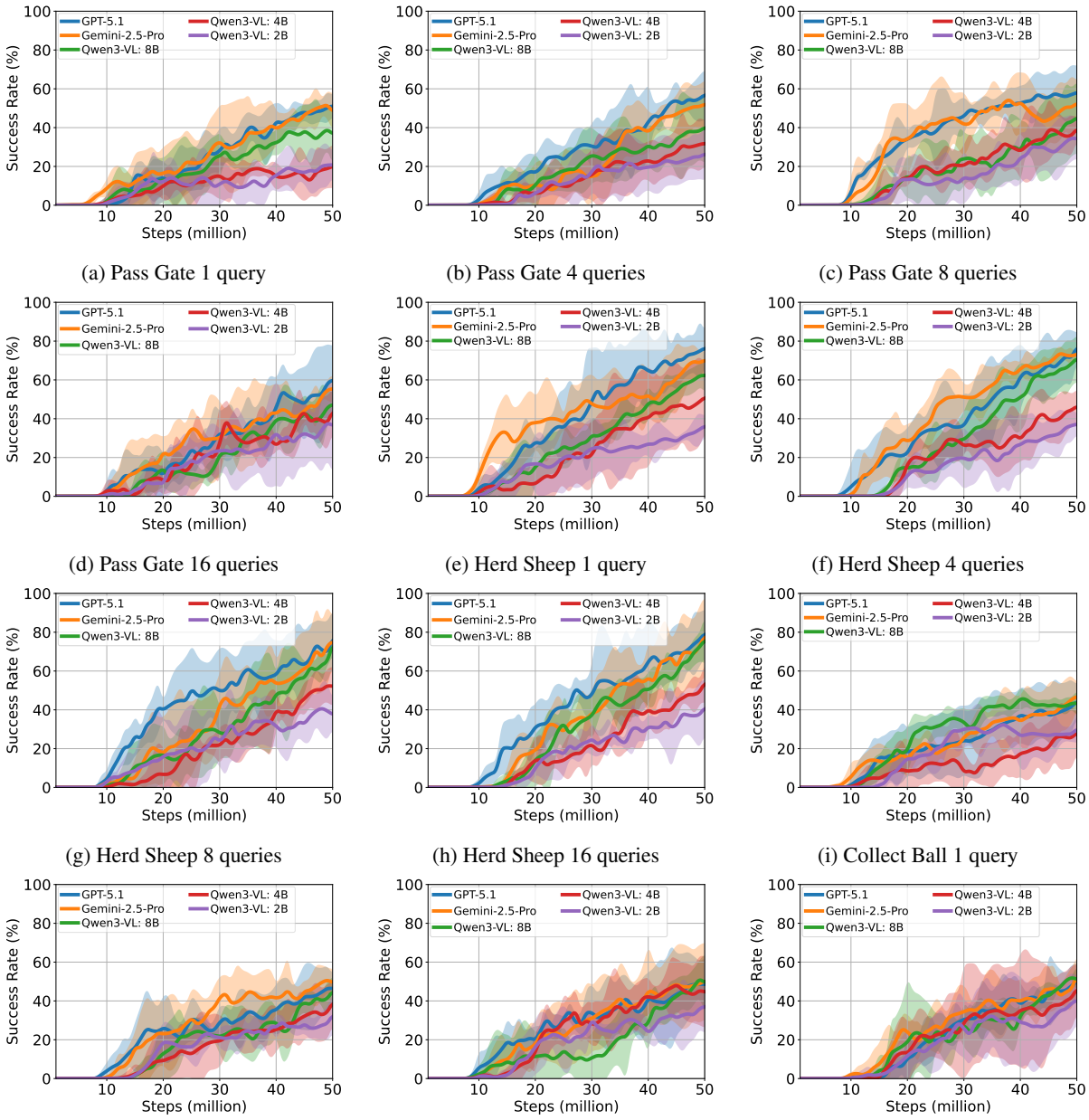
E.2 Results

Figure 11 shows the training curves of different LMMs under varying numbers of pairwise comparison queries. We observe that both employing LMMs with higher pairwise comparison accuracy and increasing the number of pairwise comparison queries lead to improved MARS-RA performance. However, when using high-accuracy LMMs, the performance gains from increasing the query count are less pronounced compared to those achieved with lower-accuracy LMMs. These results suggest a trade-off and complementary relationship between LMM pairwise comparison accuracy and the number of comparison queries. Table 4 reports the accuracy of LMM-based pairwise comparisons. We observe that commercial models generally outperform open-source models, and models with larger parameter scales tend to achieve higher accuracy. However, the accuracy of all models still leaves room for improvement. Figure 12 illustrates typical cases observed during LMM-based pairwise comparisons. When an agent's egocentric observation is severely limited and lacks sufficient visual information (e.g., teammate positions or goal locations), LMMs can still make correct judgments as long as at least one agent's observation contains informative visual cues. In contrast, pairwise comparison errors most frequently occur when all agents face a wall and lack informative visual reference cues.

F Real-World Validation Details

F.1 Task Setup

We conduct a real-world validation of MARS-RA, as illustrated in Figure 7. We deploy two XLeRobot robots to perform the Pass Gate task in a real-world indoor environment. The entire room is first captured via 3D scanning and reconstructed as a virtual 3D environment, which is then instantiated as a task in MARS-Bench. MARS-RA is trained in simulation for 70 million environment steps, using Qwen3-VL (8B) to generate pairwise comparisons, with 16 comparisons per decision. We additionally measure the pairwise comparison accuracy of Qwen3-VL (8B) on the Pass Gate task during training. The success rate is used as the evaluation metric for this experiment. The success criterion in simulation is consistent with that described in Section 8. In the real-world setting, success is defined as both agents passing through the gate into the adjacent room within two minutes without collisions; otherwise, the trial is considered a failure. All other settings follow Section 8, Appendix C



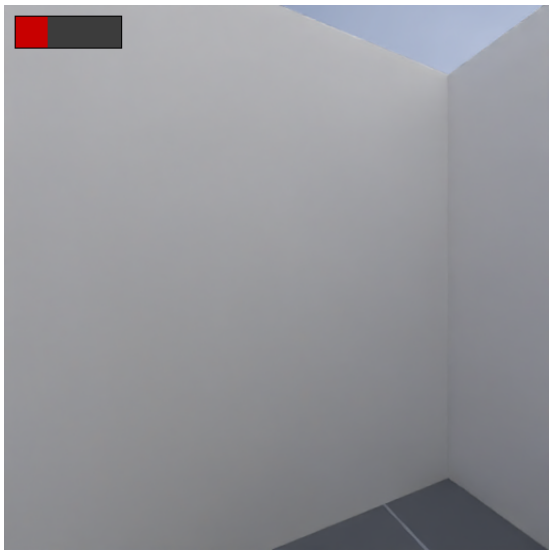
(a) Pass Gate 1 query (b) Pass Gate 4 queries (c) Pass Gate 8 queries

(d) Pass Gate 16 queries (e) Herd Sheep 1 query (f) Herd Sheep 4 queries

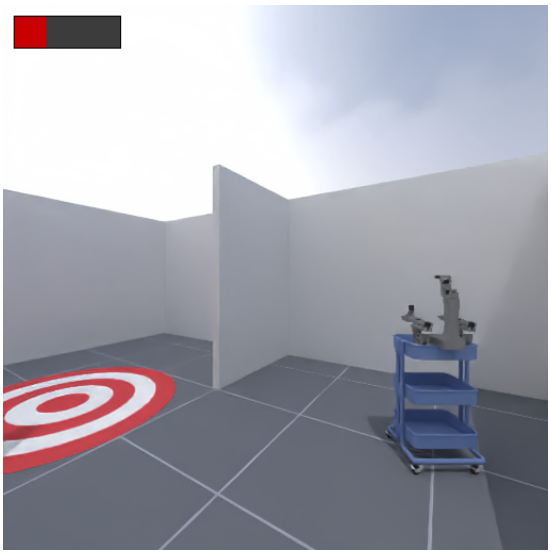
(g) Herd Sheep 8 queries (h) Herd Sheep 16 queries (i) Collect Ball 1 query

(j) Collect Ball 4 queries (k) Collect Ball 8 queries (l) Collect Ball 16 queries

Figure 11: Training Curves of MARS-RA with Different LMMs and Numbers of Queries for Pairwise Comparisons. Results are averaged over five random seeds, and error bars indicate 95% confidence intervals.



(a) Limited visual observations.



(b) Informative visual observations.

Figure 12: Pairwise comparison errors by LMMs mainly occur when both agents' egocentric observations are limited, as shown in the left image. In contrast, correct judgments can be made as long as at least one agent provides an informative observation with rich visual cues, as shown in the right image.

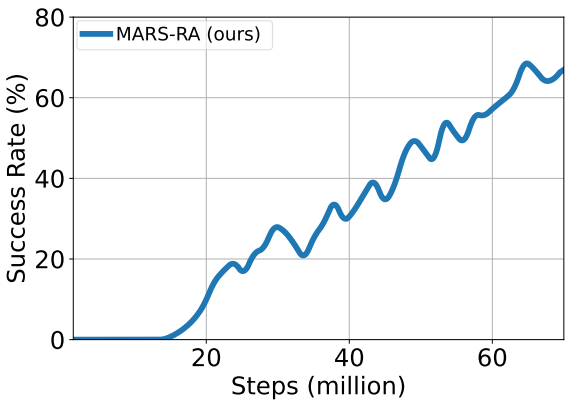


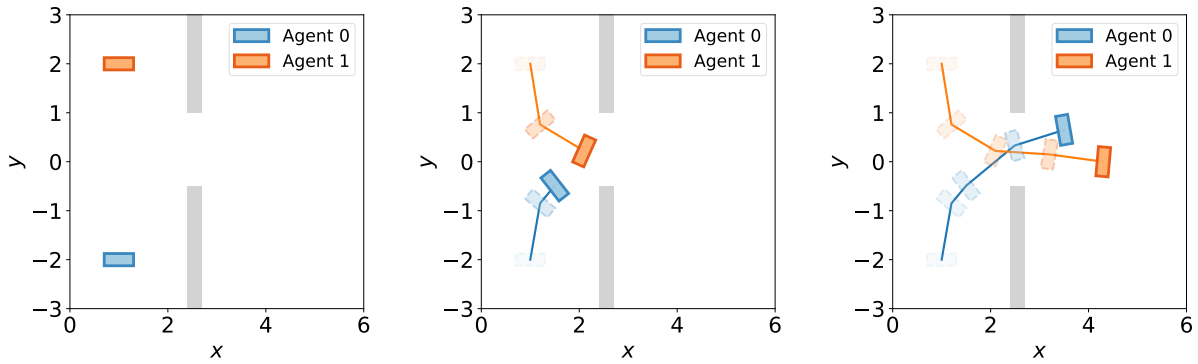
Figure 13: The learning curve obtained during training in the simulated environment for the real-world validation.

and Appendix E.

F.2 Results

The training results in simulation are shown in Figure 13. On the Pass Gate task, the success rate exceeds 67% after training. The pairwise comparison accuracy of Qwen3-VL (8B) is 71%. This result preliminarily suggests the potential of MARS-RA to guide agents toward effective cooperative policies in scenarios with real-world–level complexity.

We deploy the trained policy in the real world and conduct 25 trials, of which 16 are successful, resulting in a success rate of 64%. Notably, LMM-based pairwise comparisons are only used during training in MARS-RA, and are not required at deployment time. Figure 14 shows a representative successful trajectory of the trained policy in a real-world environment. Both agents initially move toward the gate simultaneously. The yellow agent passes through the gate first, while the blue agent maintains a safe distance and waits. After the yellow agent clears the gate and moves forward to create sufficient space, it comes to a stop, allowing the blue agent to pass through the gate once a safe clearance is available.



(a) Agent positions at t = 0.0 s. (b) Agent positions at t = 16.6 s. (c) Agent positions at t = 31.7 s.

Figure 14: A representative successful real-world trajectory of XLeRobots in the Pass Gate task, where a policy trained with MARS-RA enables coordinated, sequential gate traversal without collisions. Blue and yellow rectangles represent Agent 0 and 1, respectively; opaque rectangles indicate current positions, and transparent rectangles denote historical positions.