SMU SINGAPORE MANAGEMENT UNIVERSITY | School of Computing and Information Systems

IJCAI 2025 MONTREAL

L2M2: A Hierarchical Framework Integrating Large Language Model and Multi-agent Reinforcement Learning

Minghong Geng, Shubham Pateria, Budhitama Subagdja, Lin Li, Xin Zhao, Ah-Hwee Tan

IJCAI 2025 Technical Session

Agent-based and Multi-agent Systems (2/3)

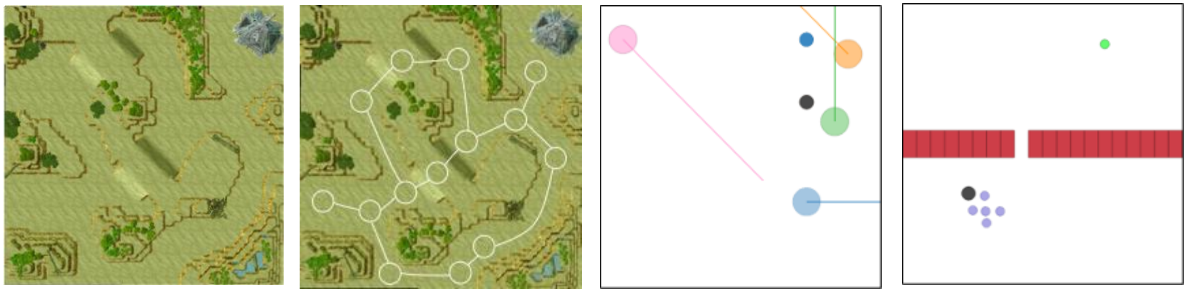
Speaker: Minghong Geng

Date & Time: Aug 20, 2025, 14:00 PM

Location: 520A, Palais des congrès, Montreal, Canada

In Proc. of the 34th International Joint Conference on Artificial Intelligence (IJCAI 2025). Main Track.

Motivation: The Challenge of Long-Horizon Multi-Agent Tasks



Problem Illustration: Multi-agent navigation in complex environments. Agents must coordinate to avoid obstacles and reach goals. Long-horizon planning required for strategic pathfinding. MARL methods generally underperform in our test in such scenarios.

Core Problem

MARL agents struggle with long-horizon sequential planning and coordination tasks that require sustained strategic thinking and temporal abstraction.

Current MARL Limitations

- **Sample Inefficiency**
  Requires millions of steps to learn complex behaviours
- **Exploration Challenges**
  Large state-action spaces are hard to explore
- **Temporal Credit Assignment**
  Difficulty linking actions to distant rewards
- **Non-Stationarity**
  Environment changes as other agents learn

Existing Hierarchical Approaches

- **Domain Knowledge Dependency**
  Require manual subtask definition
- **Limited Transferability**
  Task-specific policies don't generalize
- **Costly Retraining**
  Need to train high-level controllers from scratch
- **Scalability Issues**
  Struggle with large agent population

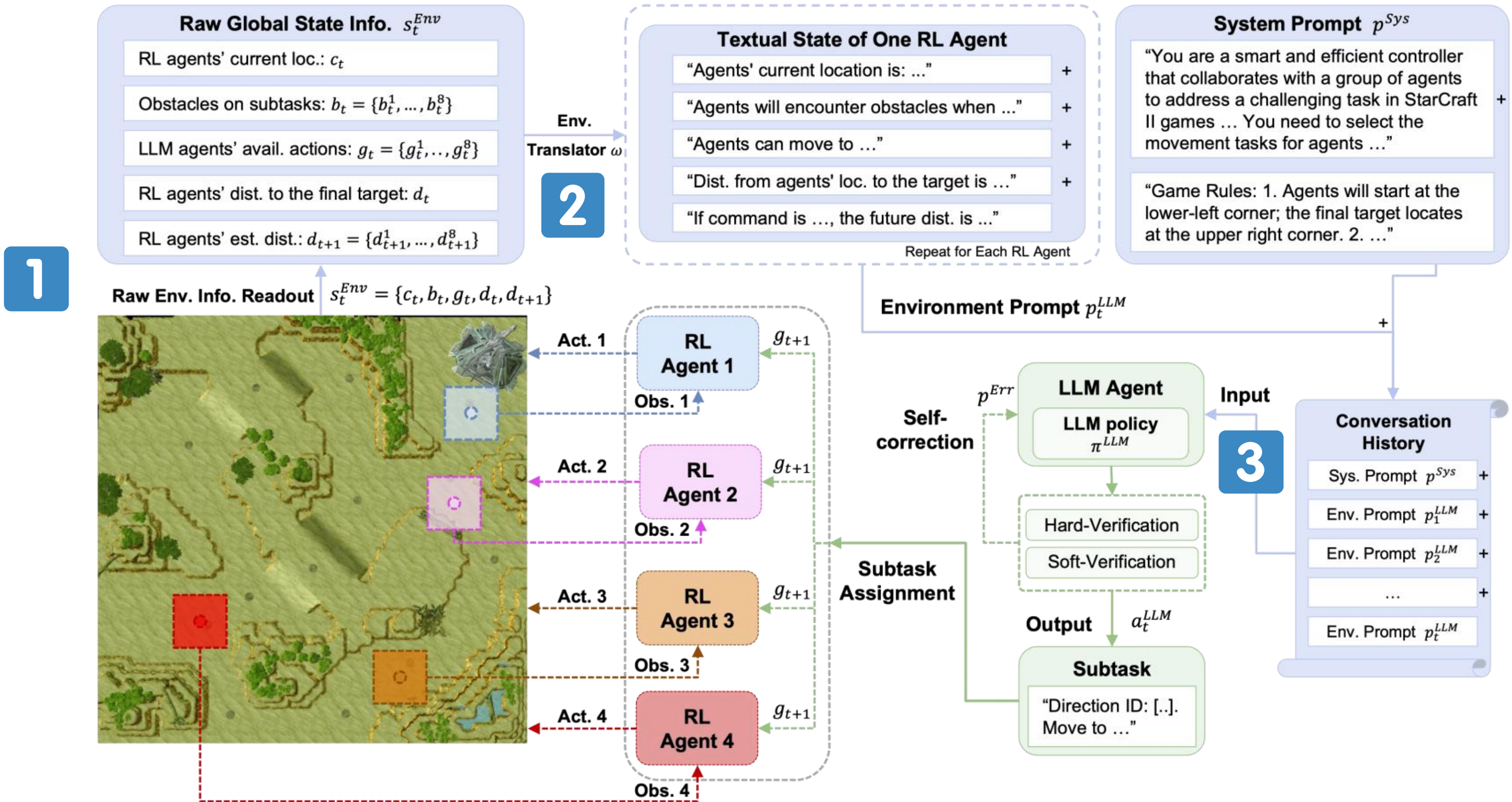
SMU SINGAPORE MANAGEMENT UNIVERSITY

School of Computing and Information Systems

IJCAI 2025 MONTREAL

L2M2 Architecture: LLM + MARL Integration

L2M2 integrates the strategic planning strengths of Large Language Models (LLM) with the accurate execution skills provided by Multi-Agent Reinforcement Learning (MARL).



LLM Agent: State Representation: Environmental prompts p^{LLM}; Action Space: Discrete subtasks G; Feedback Mechanism: Hard/soft verification

RL Agents: Observation Space: Environment + Subtask info.; Action Space: Primitive actions; Reward: Environment + Subtask rewards

The Large Language Model Agent

The *environment translator ω* enables robust communication between LLM and RL agents, which process natural language and numerical signals separately.

1

Environmental State:

$$s_t^{Env} = (c_t, b_t, g_t, d_t, d_{t+1})$$

To extract key information from the simulation environment as environmental states.

2

Environment Translator:

$$\omega: S^{Env} \rightarrow P^{LLM}$$

To map numerical environmental states into environmental prompts.

3

Environmental prompt:

To construct inputs that incorporate system prompts and existing environmental prompts utilized for LLM's inferencing.

LLM's Decision-making:

$$a_t^{LLM} = \{g_{t+1}^i \in G \mid i \in \{1, \dots, n\}\}$$

LLM agent generates temporally abstracted subtasks from the set of available subtasks G for n RL agents.

Verification on output format and action validity.
Self-correction with error descriptions if error occurs.

SMU SINGAPORE MANAGEMENT UNIVERSITY | School of Computing and Information Systems

IJCAI 2025 MONTREAL

The Reinforcement Learning Agents

The reinforcement learning (RL) agents operate under the centralized training decentralized execution framework, *taking subtask g as part of observation*.

Observation: $o_t^i = (o_t^{e,i}, o_t^{g,i})$

RL agents perceive environments partially, observing general local environment information and subtask-related information.

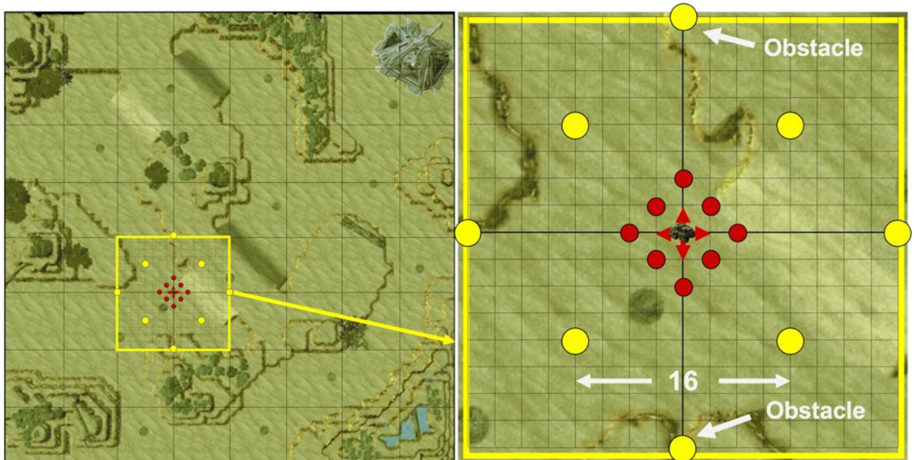
Action:

RL agents take actions follow the default settings of the benchmark environments. For example, actions in MOSMAC are no-op, movement in four directions and stop.

Reward:

$$r_t^i = r_t^{e,i} + r_t^{g,i}$$

RL agents balance immediate environmental reward with subtask-related reward towards completing their subtasks.

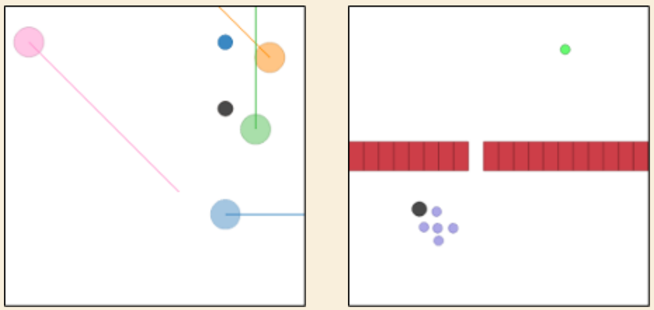


RL agents' reference points of terrain info. RL agents' mvt. dist. per step
LLM agent's subtasks targets and reference points of terrain info.

SMU Singapore Management University | School of Computing and Information Systems | IJCAI 2025 MONTREAL

Experiments: VMAS and MOSMAC

VMAS Environment [1]



The VMAS navigation (four RL agents) and passage (five RL agents) scenarios implemented in this study.

MOSMAC Environment [2]

MOSMAC scenarios implemented in this study. In each scenario, four units perform navigate tasks.

Baseline Comparisons

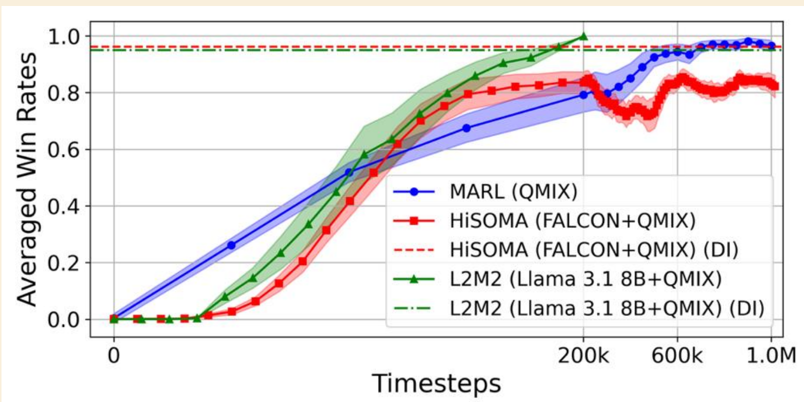
- Non-Hierarchical MARL methods (End-to-end training)
- Hierarchical methods (end-to-end training and direct integration):
- Rule-Based Controller + MARL [3]
- HiSOMA [3] (FALCON + MARL)
- L2M2 (LLM + MARL)

[1] Matteo Bettini, Ryan Kortvelesy, Jan Blumenkamp, and Amanda Prorok. 2022. VMAS: A Vectorized Multi-agent Simulator for Collective Robot Learning. In Distributed Autonomous Robotic Systems.
[2] Minghong Geng, Shubham Pateria, Budhitama Subagdja, and Ah-Hwee Tan. 2025. MOSMAC: A Multi-agent Reinforcement Learning Benchmark on Sequential Multi-Objective Tasks. AAMAS '25.
[3] Minghong Geng, Shubham Pateria, Budhitama Subagdja, and Ah-Hwee Tan. 2024. HiSOMA: A hierarchical multi-agent model integrating self-organizing neural networks with multi-agent deep reinforcement learning. ESwA.

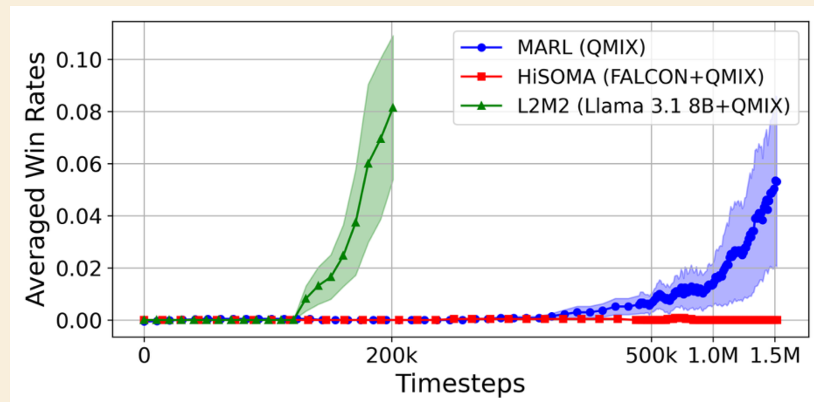
Results

L2M2 demonstrates superior *performance* and *sample efficiency (20%)*.

VMAS Scenarios



VMAS Navigation



VMAS Passage

MOSMAC Scenarios

Metrics	Rule-based Control	HiSOMA (DI)	L2M2 (DI)
	RBC w. QMIX	FALCON w. QMIX	Llama 3.1 8B w. QMIX
Avg. Win Rates (%) (mean and std)	83.13 ± 10.96	94.38 ± 1.40	98.75 ± 2.80
Avg. Returns (mean and std)	9.45 ± 0.85	10.82 ± 0.13	9.92 ± 0.21

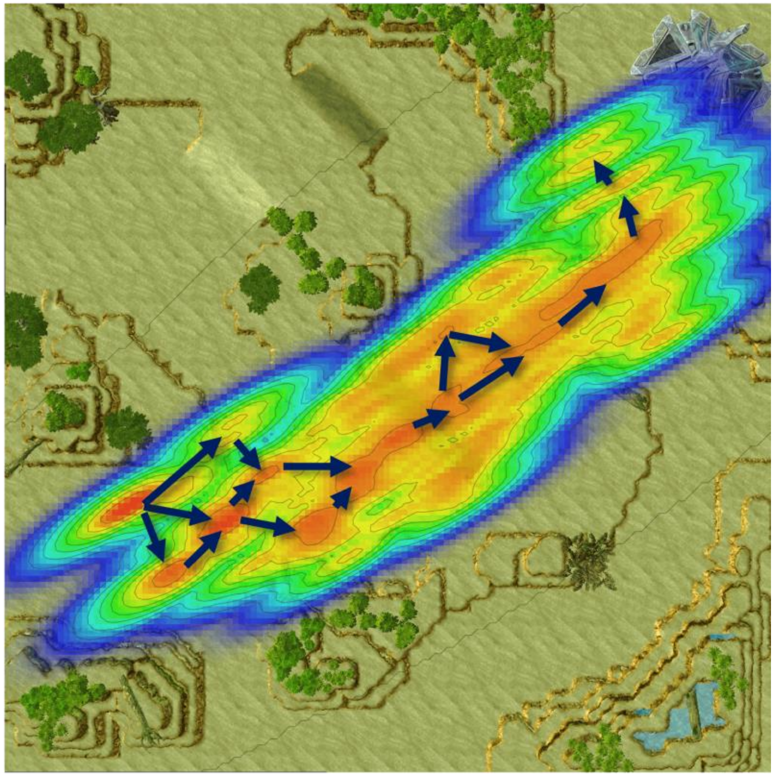
MOSMAC with subgoals

Metrics	End-to-End	Policy Transfer	
	MARL	HiSOMA (DI)	L2M2 (DI)
Avg. Win Rates (%) (mean and std)	0.00 ± 0.00	14.68 ± 14.14	68.13 ± 25.14
Avg. Returns (mean and std)	0.73 ± 0.09	7.65 ± 2.49	9.39 ± 2.06

MOSMAC without subgoals

Analysis on LLM Agent Behaviours

Kernel density estimation reveals that L2M2's LLM agent automatically generates *strategic navigation paths* that avoid challenging terrain features.



LLM Action Density Map

Heat map showing spatial distribution of LLM's action selections using kernel density estimation

Key Observations: LLM perform strategic path selection with zero-shot planning:

- High density in central regions with short path
- Low density near cliffs and ramps

Conclusion

L2M2 is an efficient and novel method for addressing challenging multi-agent problems, benefiting from the power of pre-trained language model.

Key Benefits of L2M2 Framework

Zero-Shot Planning: Immediate strategic guidance from pre-trained LLMs

Sample Efficiency: 80-85% reduction in training samples

Generalizability : Adaptable to different MARL algorithms and LLMs

Future Extensions of L2M2

Multi-Level Hierarchy: Extend to 3+ level hierarchies for complex task decomposition

Dynamic Subtask Generation: LLM automatically create new subtasks

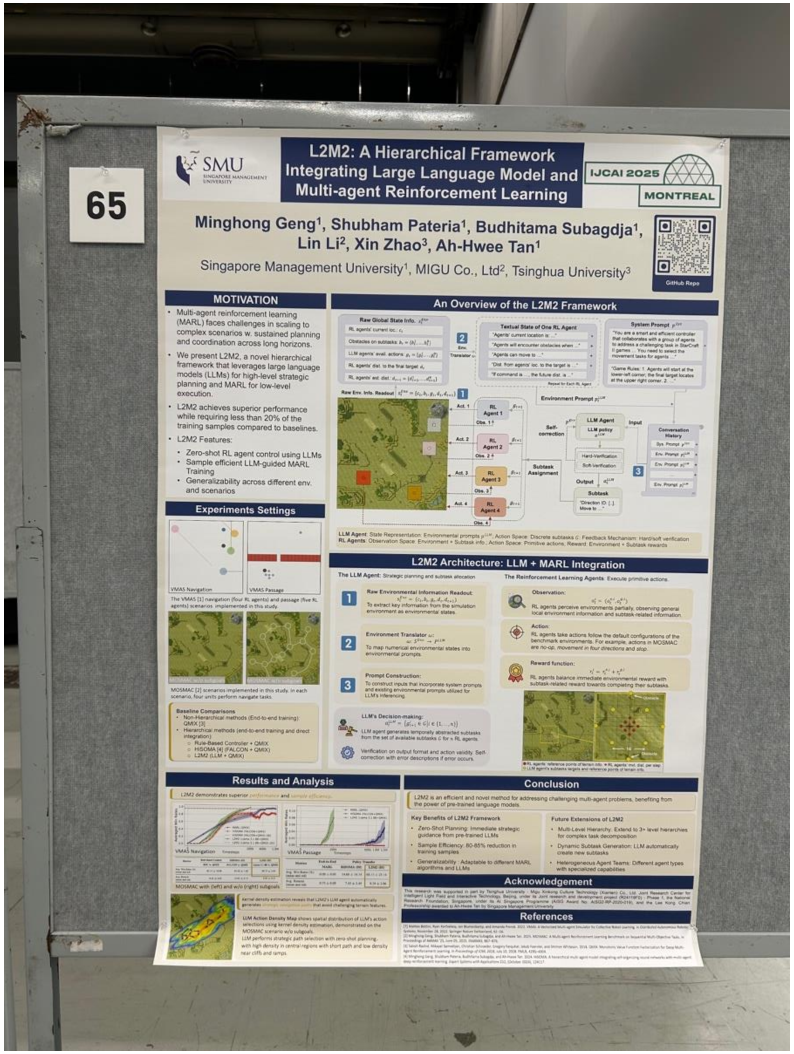
Heterogeneous Agent Teams: Different agent types with specialized capabilities

Contact Information

Thank You!

📧 Minghong Geng: mhgeng.2021@phdcs.smu.edu.sg
🌐 https://gengminghong.github.io
💻 Code: Available upon publication at https://github.com/smu-ncc
🔗 Neural and cognitive computing group
https://sites.google.com/smu.edu.sg/neural-and-cognitive-computing

Poster Presentation Information



Poster location:

Broad **65**

See you at the poster!