AdaSociety: An Adaptive Environment with Social Structures for Multi-Agent Decision-Making

Yizhe Huang*[2,1] Xingbo Wang*[2] Hao Liu[3] Fanqi Kong[2,1] Aoyang Qin[4,1] Min Tang[5,1] Xiaoxi Wang[1]
Song-Chun Zhu[1,2] Mingjie Bi[1] Siyuan Qi[1] Xue Feng[1]✉

[1]State Key Laboratory of General Artificial Intelligence, BIGAI [2]Peking University [3]New York University
[4]Tsinghua University [5]University of Science and Technology of China

The code is available at https://github.com/bigai-ai/AdaSociety

Motivation

Physical Complexity OR **Social Complexity**

An agent traverses various tasks | Multiple agents collaborate to complete a task

Multi-Agent Diverse Tasks

WE WANT THEM ALL!

Adaptive Social Connections Expanding Physical Surroundings

Novelty

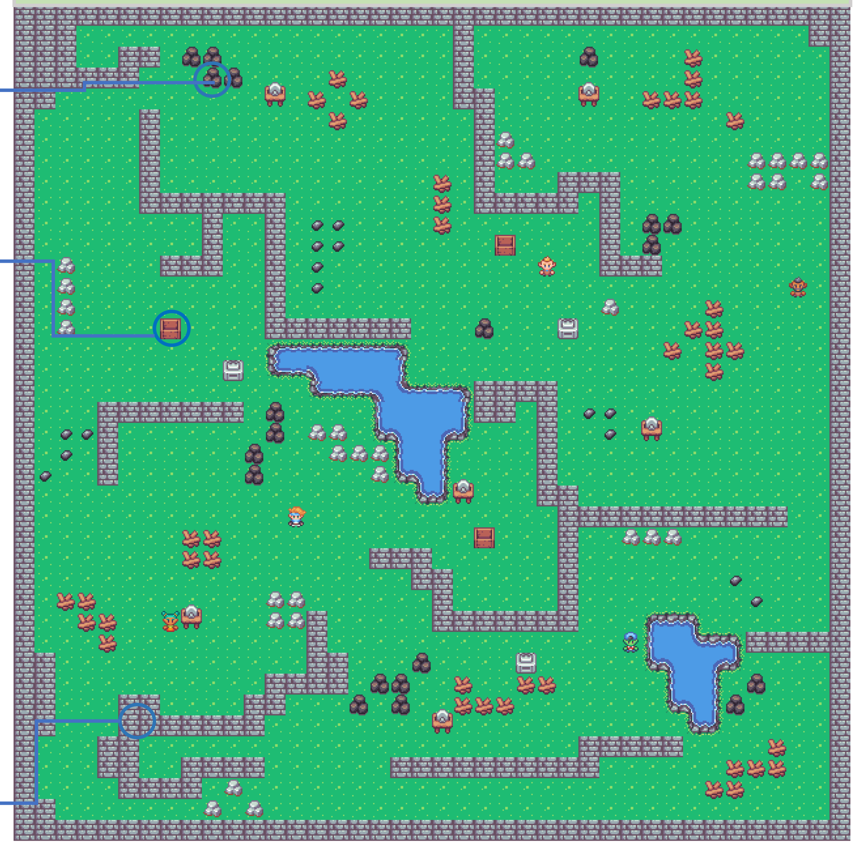
- Introduce an environment featuring **expanding physical surroundings** and **adaptive social connections**
- Introduce **social states** to explicitly and quantitatively describe the dynamic connections between entities
- Offer a **customizable environment** with three built-in minigames supporting **tensor- and LLM-based** methods

Environment

Physical Component

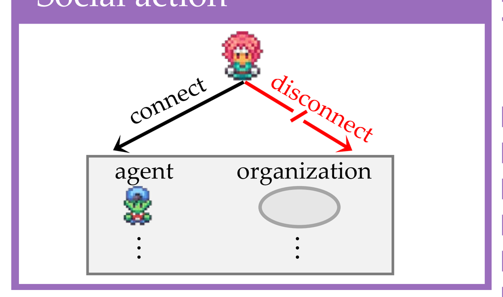
Natural resources … Synthetic resources …	Resource
resources —event→ resource e.g. + →	Event grid
Heterogeneous inv. capacity Heterogeneous preference for resources in inv.	Agent's inventory
Movement (…) Pick (…), Dump (…) Synthesize, Communicate	Physical action
Impassable wall or water	Block

Map

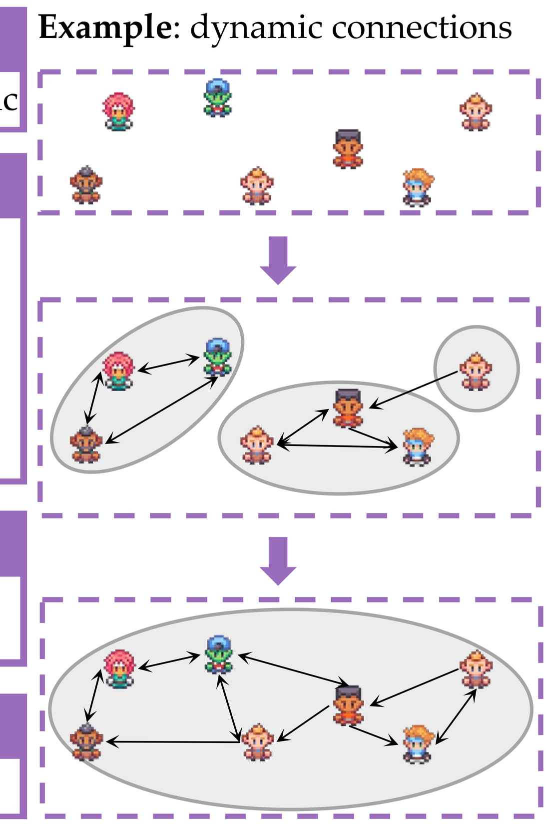


Social Component

Social connection
non-deterministic & dynamic
Social action
Hierarchy
agent-agent & agent-organization connections
Connection semantics
share information /reward

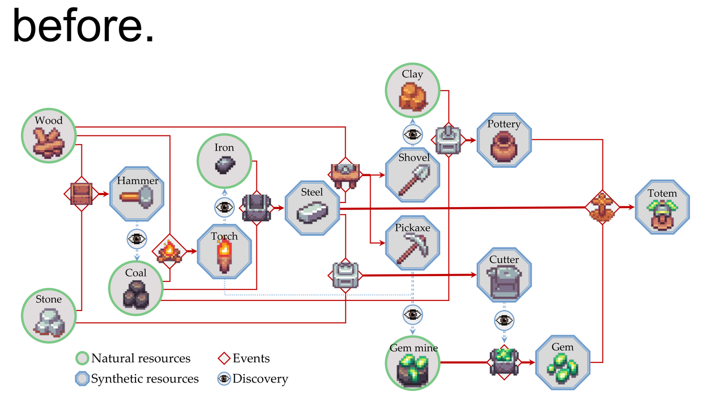


Example: dynamic connections



Agents can:
- move
- pick up / discard resources
- synthesize new resources in event grids

As the synthesis progresses, agents will gradually discover resources that have not appeared before.



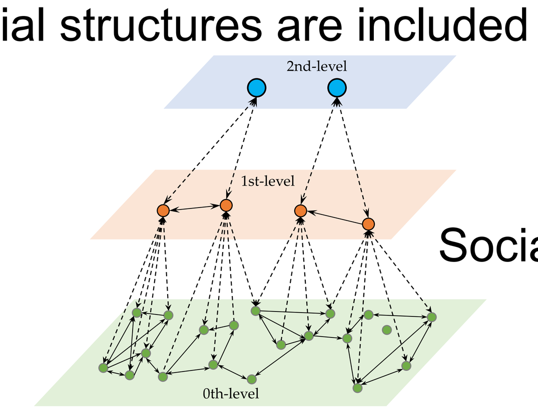
Synthesis tree

Characteristics

✓ Multi-agent general-sum game
✓ Dynamically expanding state and action spaces
✓ Dynamic social connection
✓ Massive and diverse tasks
✓ Partial observation
✓ Communication
✓ Customization
✓ LLM- and tensor-based APIs
✓ …

Agents can:
- share observation by links to other agent nodes
- share reward by links to organization nodes
- autonomously connect / disconnect with other nodes
- establish new organization nodes

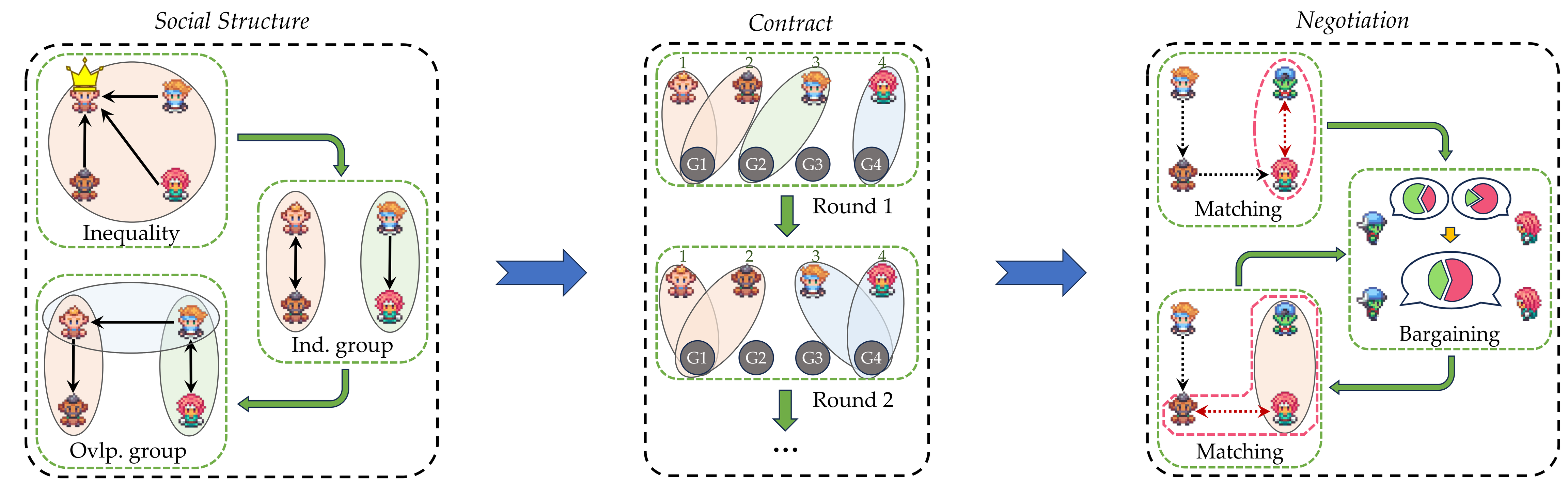
Social structures are included in observations.



Social structure

Mini-games

The three mini-games are arranged in ascending order of the complexity of decision-making. All of the three mini-games share the same physical component, containing two basic events: *HammerCraft* and *TorchCraft*.



Social Structure	Contract	Negotiation
Pre-defined connection	Adaptive connection	Adaptive connection
Pre-defined connection semantics	Pre-defined connection semantics	Negotiated connection semantics

Experiments

☐ Setup and Baseline

The **Easy** mode involves a single event, while the **Hard** mode is more complex, including six resources and two events. We take *PPO*, *RecurrentPPO(RecPPO)*, *MAPPO* and *Rainbow* as our baselines. We also design a *curriculum learning(CL)* algorithm and a *large language model planner + rule-based controller (LLM-C)* framework.

☐ Results

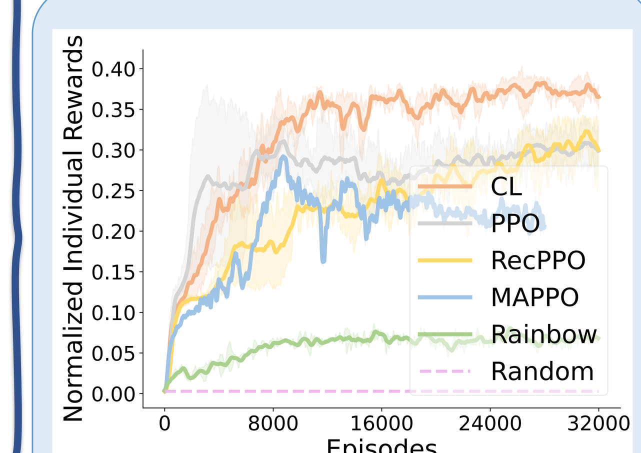


Figure 1: The training curves of tensor-based agents in ***Social Structure - Dynamic.*** *CL* agents demonstrates better performance than popular RL methods. *Rainbow* performs the worst, likely because of its general ineffectiveness in exploration.

Table 1: The normalized rewards of tensor-based methods in ***Contract*** and ***Negotiation***. Popular RL methods often fall into local optima. While *CL* enhances tensor-based agents, its training efficiency and effectiveness decline as the scale of the environment increases.

		CL	PPO	RecPPO	MAPPO	Rainbow
Con.	Easy	0.9136± 0.0023	0.2286± 0.0003	0.2276± 0.0015	0.2271± 0.0003	0.1987± 0.0127
	Hard	0.2773± 0.0466	0.1151± 0.0002	0.1149± 0.0000	0.1137± 0.0005	0.0868± 0.0033
Nego.	Easy	0.3543± 0.0229	0.2276± 0.0006	0.2278± 0.0004	0.2147± 0.0001	0.1969± 0.0105
	Hard	0.1945± 0.0109	0.1093± 0.0027	0.1107± 0.0019	0.0946± 0.0032	0.0905± 0.0024

Table 2: The normalized rewards of *LLM-C* in ***Contract*** and ***Negotiation***. Due to the rich prior knowledge, *LLM-C* achieves significantly high rewards. However, these results still fall short of the optimal rewards achieved by an Oracle.

	Social Structure	*Contract*	*Negotiation*
Easy	-	0.8433± 0.1312	0.8733± 0.1116
Hard	0.7894± 0.0444	0.6499± 0.1716	0.6862± 0.1027

AdaSociety maintains a rational complexity level for popular decision-making methods.

These methods fails to utilize social structure provided in ***AdaSociety*** to form effective cooperation.

Prior knowledge can be of great help to the algorithm.

Contact: Xue Feng <fengxue@bigai.ai>