Notes: Building Cooperative Modular Embodied Agents With LLM

Kaiwen Bian

October 18, 2024

Address challenging multi-agent cooperation problems with **decentralized control**, **raw sensory observations**, **costly communication**, and **multi-objective tasks** instantiated in various **embodied environments**. Incorporate them into a **cognitive-inspired** modular framework that integrates with perception, memory, and execution. Thus building a **Cooperative Embodied Language Agent CoELA**, who can plan, communicate, and cooperate with others to accomplish long-horizon tasks efficiently.

Also conducted a user study for human-agent interaction and discovered that CoELA communicating in natural language can earn more trust and cooperate more effectively with humans.

Decentralized Partially Observable Markov Decision Process (DEC-POMDP)

The setting can be defined as an extension of the decentralized partially observable Markov decision process (DEC-POMDP), which can be formalized by:

$$(n, S, \{\Sigma_i\}, \{A_i\}, \{O_i\}, T, G, R, \gamma, h)$$

- n: Number of agents.
- S: Finite set of states.
- $A_i = A_{Wi} \cup A_{Ci}$: Action set for agent i, including:
  - A_{Wi}: Finite set of world actions.
  - A_{Ci}: Communication actions to send a message $\sigma_i \in \Sigma_i$.
- $O_i = O_{iW} \times O_{iC}$: Observation set for agent i, including:
  - O_{iW}: World observations the agent receives through its sensors.
  - $O_{iC} = \Sigma_1 \times \cdots \times \Sigma_n$: Set of possible messages the agent can receive from any of its teammates.
- $T(s, a, s') = p(s'|s, a)$: Joint transition model defining the probability that, after taking joint action $a \in A_1 \times \cdots \times A_n$ in $s \in S$, the new state $s' \in S$ is achieved.

- $G = \{g_1, \ldots, g_k\}$: Task defined with several sub-goals for the agents to complete.
- $R(s, a, s') = -c(a) + \sum_{i=1}^{k} 1(s' = g_i) - 1(s = g_i)$: Reward function for the team, where:
  - $c(a)$: Cost for action a.
  - $1(\cdot)$: Indicator function that checks if the sub-goal g_i is satisfied in the world state s.
- γ: Discount rate.
- h: Planning horizon.

The agents are capable of executing one of the actions from the action space $A = A_{\text{NAV}} \cup A_{\text{INT}} \cup A_{\text{COM}}$, where A_{NAV} includes navigation actions, A_{INT} includes interaction actions, and A_{COM} includes a communication action with which the agent can send a message. **This is a ridiculously big action space**.

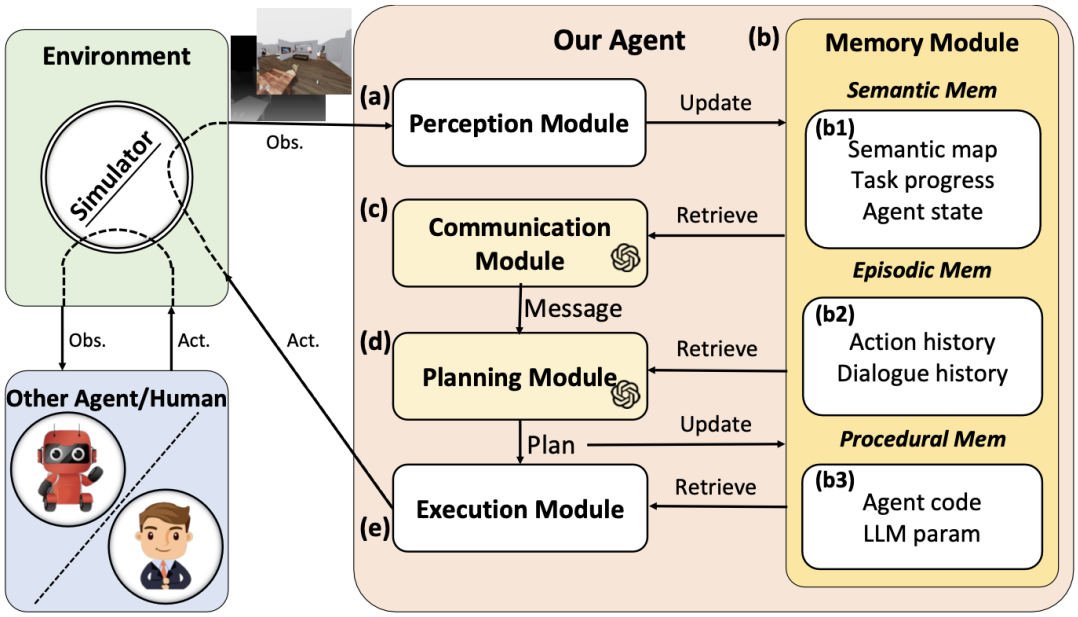
CoELA

Cognitive Inspired Structure

CoELA is a Cooperative Embodied Language Agent with a **cognitive architecture** with a novel modular framework that utilizes the **rich world knowledge**, **strong reasoning ability** and **mastery natural language understanding and generation capability** of LLMs, who plan and communicate with others to cooperatively solve complex embodied tasks. Modules include:

1. **Perception Module**: to perceive the observation and extract useful information.
2. **Memory Module**: to mimicking human's long-term memory to maintain the agent's understanding of both the physical environment and other agents.
   (a) **Semantic Memory** stores CoELA's knowledge about the world including a semantic map, the task progress, the state of self, and the state of others.
   (b) **Episodic Memory** stores CoELA's experience about the past including the action history and dialogue history.
   (c) **Procedural Memory** contains knowledge including how to carry out specific high-level plans in a specific environment implemented in code and the neural models' parameters.
3. **Communication Module**: to decide what to communicate utilizing the strong dialogue generation and understanding capability of LLMs.
   (a) Retrieves **the related information from the Memory Module** including the semantic map, task progress, agent state, others state, and the action and dialogue history.
   (b) Convert these into text descriptions using templates, finally prompt the LLMs with the concatenation of **Instruction Head, Goal Description, State Description, Action History, and Dialogue History** to generate the message to send.

4. **Planning Module**: to decide high-level plans including when to communicate considering all the information available.
   (a) Retrieve the related information from the Memory Module and converting them into text descriptions as in the Communication Module.
   (b) Compile an Action List of all **available high-level plans** proposed according to the current state and the procedural knowledge stored for the LLMs to make the choice, which **formalization makes it easier for the LLMs to concentrate on the reasoning** and make an executable plan without any few-shot demonstrations easily.
   (c) Prompt the LLMs with current information and the proposed Action List to generate a high-level plan.
5. **Execution Module**: to execute the plan by generating primitive actions using procedures stored in the memory module.
   (a) LLMs are poor at making low-level controls, so design an execution module for primitive actions. Practically, this design can also reduce the LLM inference time and is time-saving and economical.



Inference Time Iteration

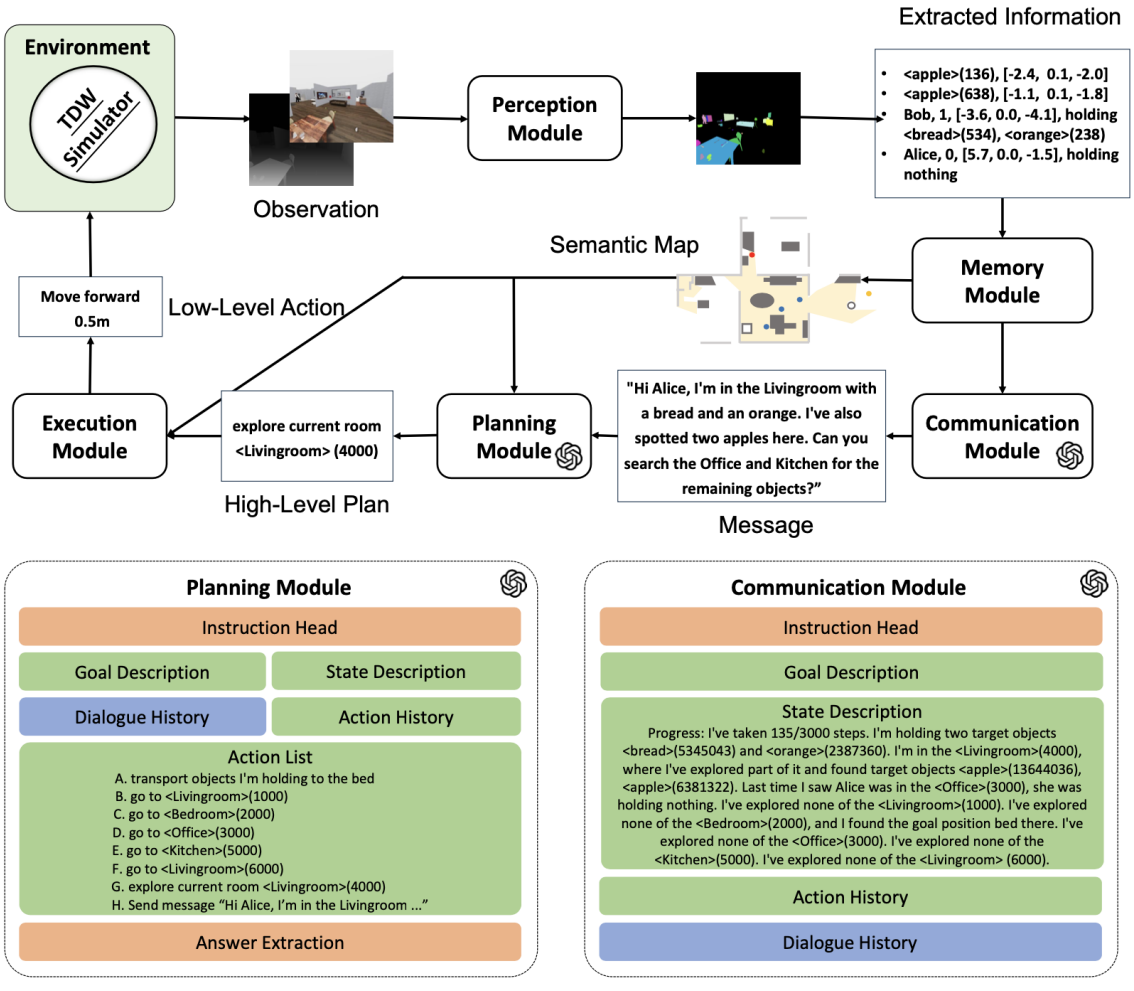
At each interaction step, CoELA operates through the following modules:

1. **Perception Module (a)**: Perceives the raw sensory observation received from the environment and extracts new information.
2. **Memory Module (b)**: Updates with the new information, storing knowledge and experiences of the world and others.
3. **Communication Module (c)**: Tackles the challenge of efficient communication with a two-step method:

- Decides on what information to send.
- Determines whether to send the message or choose an alternative plan by deliberately retrieving related information from the Memory Module (b) and utilizing an LLM to generate the best message to "send in mind" beforehand.

4. **Planning Module (d)**: Driven by an LLM with strong reasoning ability, it decides on which plan to execute based on the information retrieved from (b) and the available actions proposed regarding the current state. The generated plan is used to update the Episodic Memory (b2).
5. **Execution Module (e)**: Retrieves procedural knowledge stored in (b3) to convert the high-level plan into primitive actions executable in the environment.

Working Example on TDW-MAT



1. The environment provides an **observation** of 512 * 512 first-person view RGB image and Depth image.
2. The **Perception Module** takes these in, builds 3D point clouds, then extracts the states (positions, names, IDs, objects holding if agents) of the key objects including target objects, containers, and the agents, and builds a local occupancy map.

3. The **Memory Module** uses the extracted states of the key objects and the local occupancy map to **construct and update the semantic map**, which is stored in Semantic Memory.
   (a) Stores **the task progress, the states of the agents in the Semantic memory**.
   (b) Stores **agent's action and dialogue history in the Episodic Memory**, which are also updated when a message is received.
4. The **Communication Module** converts the semantic map, task progress, and agents' states into **textual State Description** and concatenates it with the Instruction Head, Goal Description, Action History, and Dialogue History as the prompt to condition the LLM on current states and **generate the message to be sent beforehand**.
5. The Planning Module similarly takes these inputs and converts them into a **prompt with the addition of an Action List compiled with all available high-level plans including sending the message just generated**, then taking advantage of the chain-of-thought prompting to decide on the high-level plan.
6. The **Execution Module first uses an A-Star-based planner to find the shortest path** from the current location to the target location with the help of the semantic map if needed, then carry out the interaction required to finish the high-level plan.