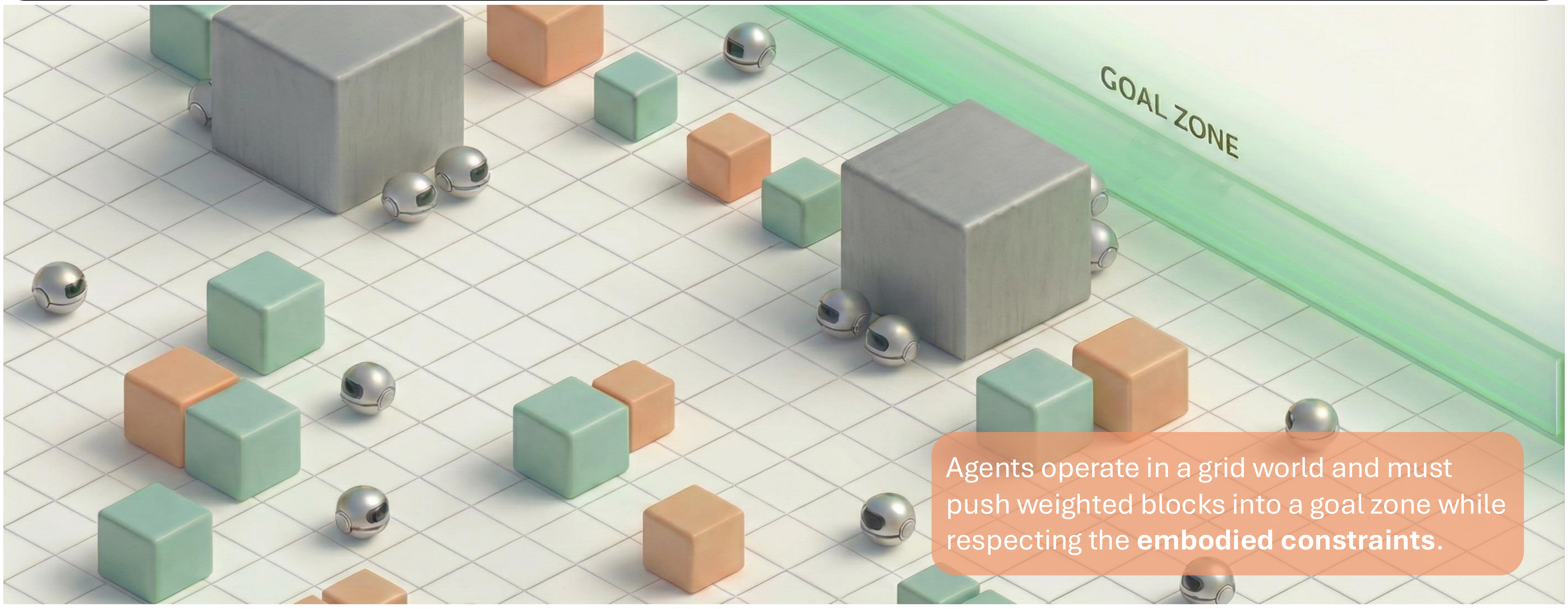
CUBE: Collaborative Multi-Agent Block-Pushing Environment for Collective Planning with LLM Agents

Hanqing Yang *[1], Narjes Nourzad *[2], Shiyu Chen [1], Carlee Joe-Wong [1]
[1] Carnegie Mellon University [2] University of Southern California
* Equal contribution † Work done during an internship at Carnegie Mellon University



Agents operate in a grid world and must push weighted blocks into a goal zone while respecting the **embodied constraints.**

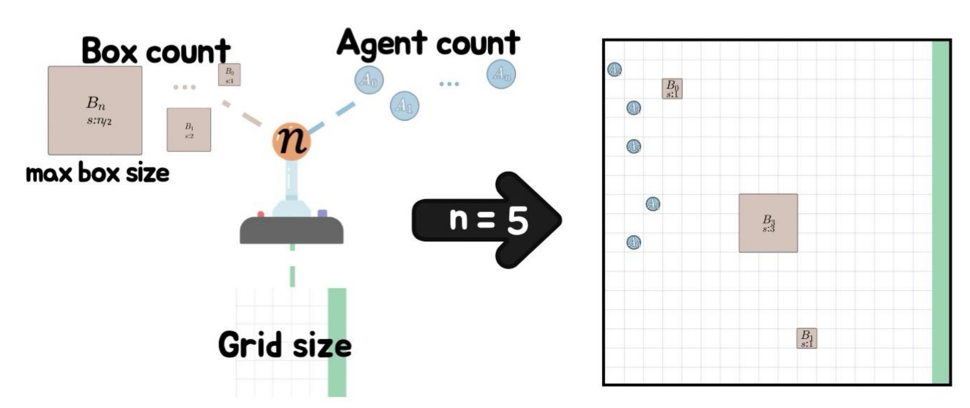
Why and Why CUBE

Why: Study **cooperative intelligence** in **LLM agents** at **scale.**

Why Embodied Tasks: Embodied constraints make cooperation necessary and keep the task **meaningful** even as the number of agents **scales**. The scene's dynamic reconfiguration from agent and block movement demands rich **collective planning**.

Why CUBE: 1) **Embodied scenario** where cooperation is necessary and scalable; 2) **Dual layer** design fits **LLM agents** by letting them plan symbolically while acting physically, with physical feedback guiding symbolic reasoning; 3) The env-dynamics create **emergent cooperation challenges** that require **extensive coordination.**

Controllable Difficulty Curriculum

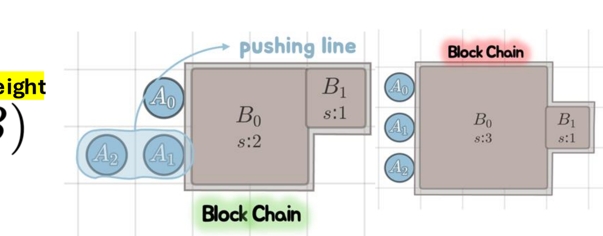


- Setting n fixes grid size, agent count, and block distribution.
- Larger n increases cooperation efforts.
- reproducible and controllable difficulty curriculum.

Embodied Constraints

- Agents and blocks occupy discrete cells.
- Agents exert 1 unit of **force** in their movement direction.
- Each **block's weight** equals the force required to move it and is proportional to its side length; agents may push from any side.
- A push succeeds when enough agents are **aligned** on the block face.
- Force transmits through **aligned** agents, forming an **agent chain**.
- Adjacent blocks form a **block chain**, which moves in direction d only if the applied force in d exceeds the total weight of all blocks in the chain.

$$\underbrace{\sum_i \overbrace{f_i}^{\text{force}}}_{\text{Agent Chain}} \geq \underbrace{\sum_{B\in\text{chain}}}_{\text{Block Chain}} \overbrace{w(B)}^{\text{Block Weight}}$$



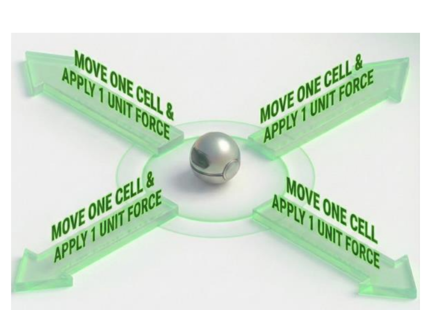
Dual Layer Environment for LLM Agents

Human reasoning blends symbolic planning with embodied feedback: **we form abstract plans, act in the world, observe discrepancies between expectation and outcome, and adjust**. Embodied LLM agents need this same loop: symbolic structure for reasoning about goals and relations, and acting in embodied scenarios to learn the effects of their actions from feedback as the environment evolves. CUBE supports this by allowing agents to plan symbolically while grounding those plans in an uncertain, continuously changing environment.

Primitive Layer / Symbolic Layer

Action

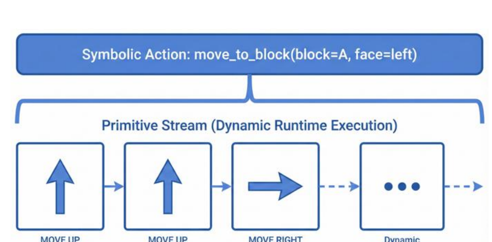
Primitive Layer:



{STAY, UP, DOWN, LEFT, RIGHT}

Symbolic Layer:

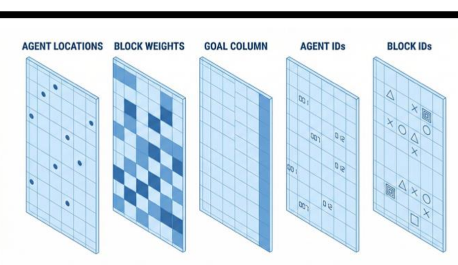
Action	Arguments
`move`	(*direction*, *steps*)
`move_to_block`	(*block*, *face*)
`rendezvous`	(*block*, *face*, *count*, *timeout*)
`push_block`	(*block*, *steps*)
`yield_block`	(*block*, *steps*)
`idle`	(*steps*)
`wait_agents`	(*count*, *timeout*)



Each symbolic action unfolds into a sequence of primitive actions during runtime

Obs

Primitive Layer:



Multi-Channel

Symbolic Layer:

```
{
  "grid_size": 20,
  "agent_positions": {0: (5, 5), 1: (5, 6)},
  "blocks": {
    "B1": {"weight": 2, "goal_dist": 4,
      "location": (10, 10)},
    "B2": {"weight": 1, "goal_dist": 9,
      "location": (15, 15)}
  },
  "history": ["a1", "a2", "a3", "..."]
}
```

A structured representation: grid size, agent positions, block properties (size, location, distances to the goal), and the history of symbolic actions taken.

Feedback

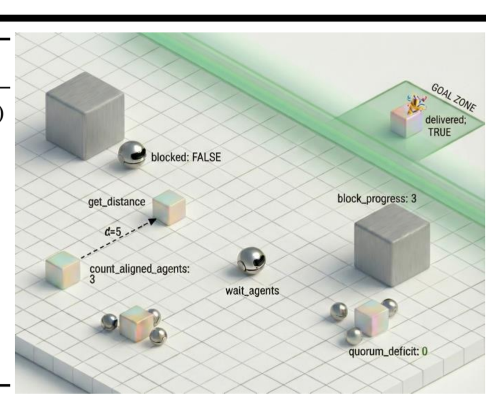
Primitive Layer:

$$r_i^{(t)} = -r_s + \frac{D^{(t)}}{|\mathcal{A}|}$$

where $D^{(t)} = \sum_{B\in\text{delivered}} r_d \cdot w(B)$

Symbolic Layer:

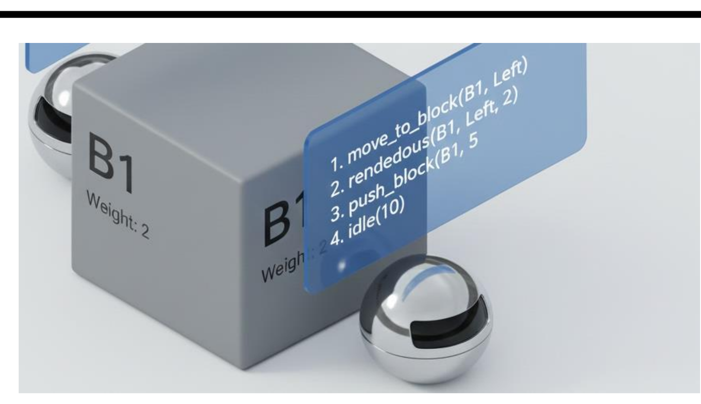
Concept	Arguments
`get_distance`	($entity_1$, $entity_2$)
`is_aligned_with_block`	(*block*, *side*)
`count_aligned_agents`	(*block*, *side*)
`all_aligned_positions`	(*block*, *side*)
`block_progress`	(*block*)
`delivered`	(*block*)
`quorum_status`	(*block*, *side*)
`quorum_deficit`	(*block*, *side*)
`blocked`	(*block*, *side*)



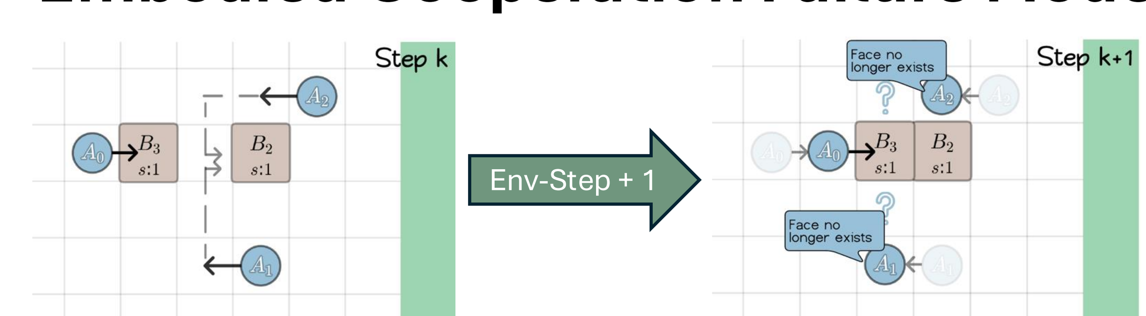
Plan

$$\pi_i = [a_i^1(\theta_i^1), a_i^2(\theta_i^2), \ldots, a_i^T(\theta_i^T)]$$

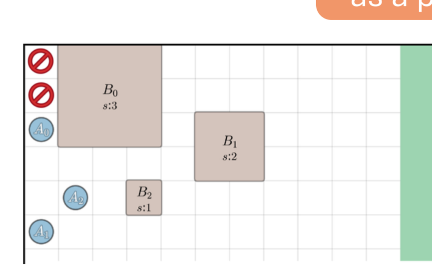
Symbolic actions seems compact but are highly expressive - their parameters unlock many instantiations, creating a **rich planning space.**



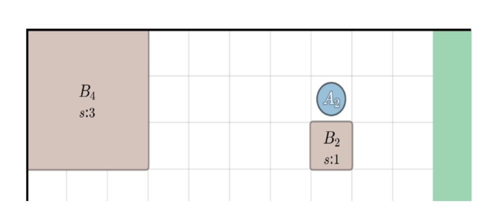
Embodied Cooperation Failure Modes



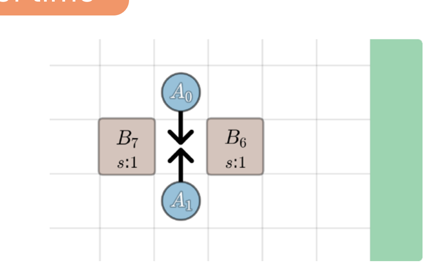
Embodied movement introduces uncertainty during execution, as a previously feasible plan could become infeasible over time



Agent movements can temporarily occlude the remaining positions

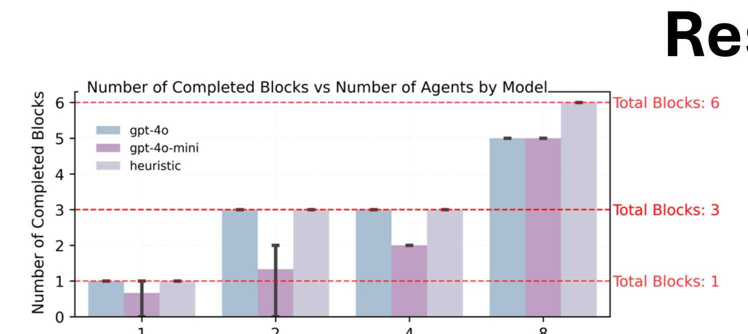


A corner block is not actionable (can't push)



Cell access race during concurrent movement

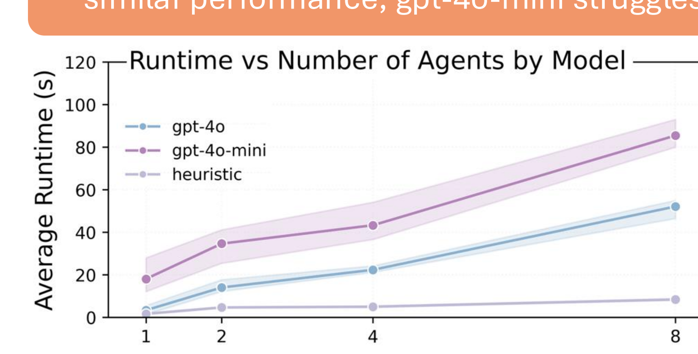
Results

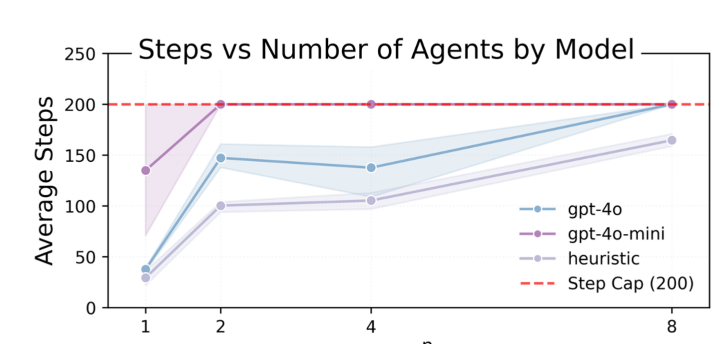


When n is small, the heuristic baseline performs well. The naive gpt-4o agent achieves similar performance; gpt-4o-mini struggles

Heuristic Baseline: A hand-crafted greedy strategy where agents repeatedly select the block closest to the goal and push it, yielding consistent cooperative behavior.

Naive LLM Baseline: Each agent generates short symbolic plans from its observations, executes them in the environment, and replans once the previous plan finishes.

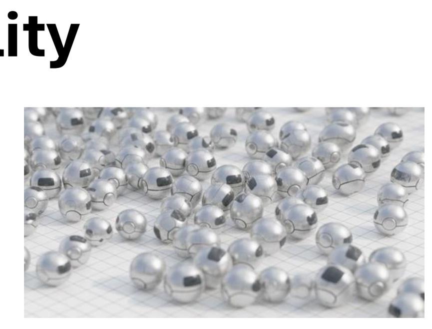




The heuristic baseline produces effective cooperative behavior when n is small, while naive LLM agents (gpt-4o, gpt-4o-mini) can generate symbolic plans but are less reliable and less efficient, highlighting the coordination gap that CUBE is designed to expose

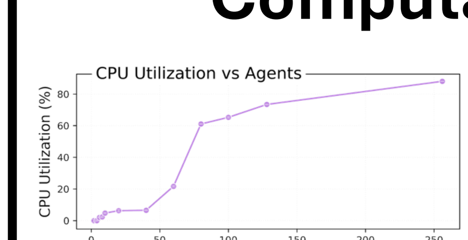
Scalability

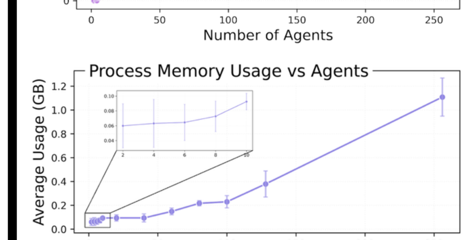
Number of agents

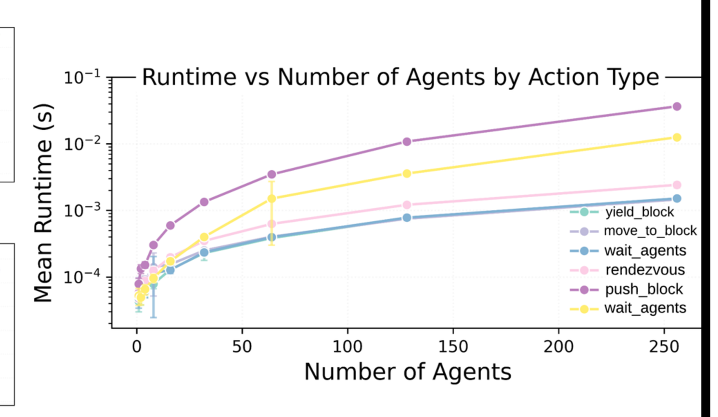


Task Difficulty
Number of blocks
Size of blocks

Computational Overhead







Model	Time (s)
gpt-4o	0.7 ± 0.09
gpt-4o-mini	1.6 ± 0.07

LLM Inference Latency

- LLM inference time >> Environment step time
- CUBE adds **negligible** computational overhead even when scaled to **hundreds** of agents

Challenges & Opportunities

Dynamic Scene and Task
- Moving any block or agent reshapes the task structure (temporarily or long-lasting)
- Existing plans may become invalid

Spatial Reasoning
- Navigate to avoid blocking or being blocked
- Understand the structure of block chains and agent chains

Asynchronization
- Agents may plan and finish their plans at different times

Synchronization
- Achieve spatio-temporal alignment at a block
- Coordinate through joint communication and planning

Collective Intelligence under Uncertainty
- Predict teammates' behavior, including intent, actions, and effects
- Decide when, who, and what to communicate
- Deploy adaptive strategies to adjust plans or policies as the environment changes