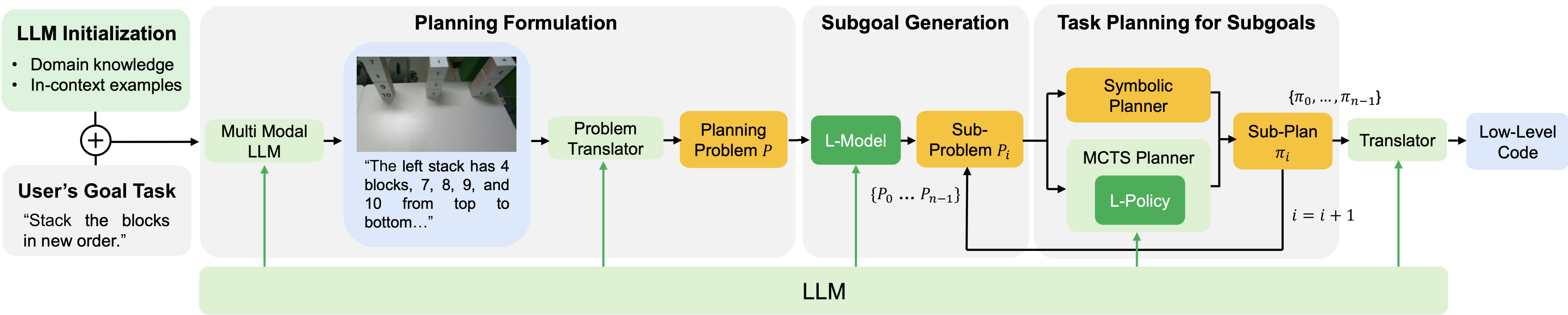
Neuro-Symbolic Task Planning and Replanning using Large Language Models

Minseo Kwon and Young J. Kim
Dept of Computer Sci. and Eng., Ewha Womans University
minseo.kwon@ewha.ac.kr, kimy@ewha.ac.kr

Abstract

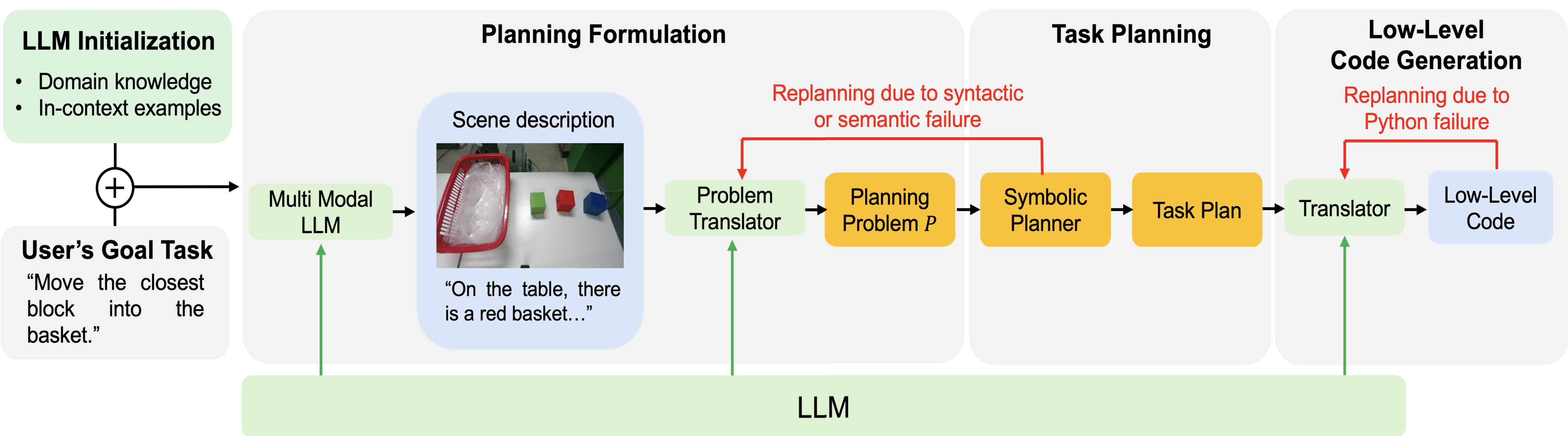
In robotic task planning, symbolic planners are robust but struggle with long, complex tasks due to their exponential growth of the search space. By contrast, LLM-based planners reason faster and incorporate commonsense knowledge, but show lower success rates and lack failure recovery. We present a **1) neuro-symbolic task planning framework with subgoal decomposition** to overcome the drawbacks of symbolic planners (slow speed) and LLM-based methods (low accuracy). It breaks down complex tasks into subgoals using a multimodal LLM, then selects either a symbolic planner or an MCTS-based LLM planner to handle each subgoal according to its complexity. Furthermore, we propose a **2) neuro-symbolic task replanning algorithm** for task planning failure recovery. During task planning and low-level code generation, the LLM acts as a multimodal error detector, ensuring the validity of the planning process and triggering replanning when necessary. We demonstrate that both task planning and replanning improve high success rates across diverse PDDL domains, as well as in real and simulated robotics environments.

1) Task Planning with Subgoal Decomposition



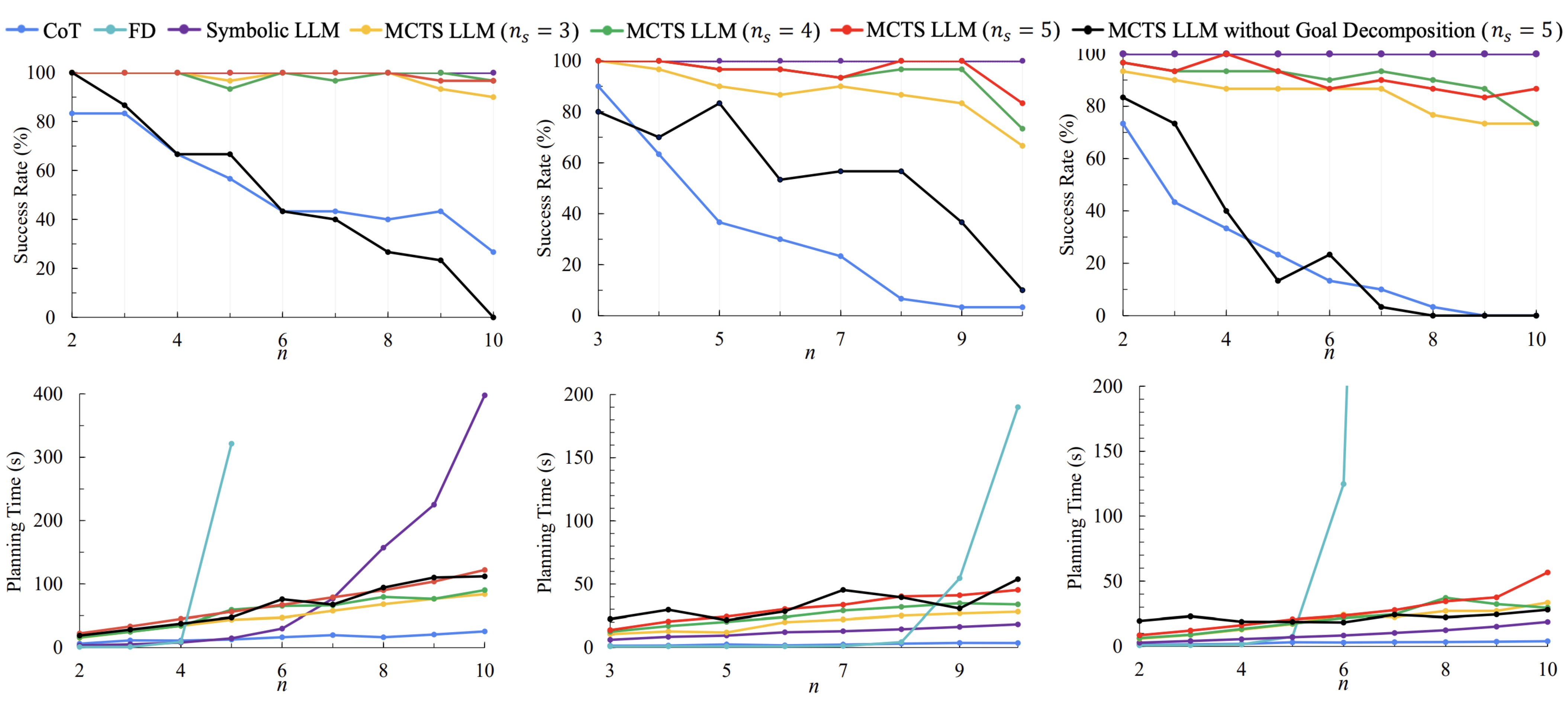
- **Planning Formulation**: Based on the multimodal LLM-generated scene description, user-provided goal task, domain knowledge, and an in-context example, the LLM formulates the planning problem P in PDDL format.
- **Subgoal Generation**: We then prompt the LLM with the domain knowledge and examples to generate a sequence of subgoals, and decompose P into n sub-problems P_i.
- **Task Planning for Subgoals**: If the given subgoal is complex, we use the MCTS algorithm while using LLMs as policy models to solve each subgoal, reducing the correction inefficiency in LLM-based planners. If the given subgoal complexity is moderate, we use a symbolic planner instead.

2) Task Replanning with Syntax and Semantic Checking



- Based on the planner's output, the LLM detects syntax and semantic errors in the **problem PDDL** and replans the task by reprompting error messages.
- If a run-time error occurs when executing the **Python code**, we also replan the task by reprompting the exception messages to LLM.

Experiment Results: 1) Subgoal Decomposition



Barman-new **Blocksworld-new** **Grippers-new**

- **Subgoal Decomposition**: Symbolic LLM and MCTS LLM planners achieve an 88.2% - 100% average success rate, outperforming the baseline LLM planner and reducing planning time compared to the baseline symbolic planner.

Experiment Results: 2) Task Replanning

Domain	Problem failure		Python failure	Success rates
	syntax	semantic		
Stack	6.6	16.7	3.3	73.3
Rearrange	3.3	3.3	0	93.3

Success and failure rates (%) without replanning

Domain	Problem failure		Python failure	Success rates
	syntax	semantic		
Stack	0	3.3	0	96.7
Rearrange	0	0	0	100

Success and failure rates (%) with replanning

- **Task Replanning** up to four times increased success rates to nearly 100%, outperforming runs without replanning.

This paper is an extended abstract version of the original papers [1], [2].

[1] M. Kwon, Y. Kim, and Y. J. Kim, "Fast and accurate task planning using neuro-symbolic language models and multi-level goal decomposition," in *2025 IEEE International conference on robotics and automation (ICRA)*. IEEE, 2025, accepted.

[2] M. Kwon and Y. J. Kim, "Neuro-symbolic task replanning using large language models," *Journal of Korea Robotics Society*, vol. 20, no. 1, p. 52–60, Feb. 2025.