

LLM-Guided Common-Sense Planning for Long-Horizon Underspecified Tasks

Roshan Dhakal

Abstract—Robots are increasingly expected to perform long-horizon tasks from high-level instructions such as “set a dining table” or “make dinner.” Such instructions are often *underspecified*: they name a desired activity without stating the goal state required to complete it, leaving the robot to infer intermediate objectives that humans supply effortlessly through common sense. This common-sense gap is a core obstacle to humanlike behavior in humanoid robots operating in everyday human environments. Classical planners require an explicitly defined goal and cannot operate directly from such instructions, whereas large language models (LLM)s possess broad common-sense knowledge but offer no guarantees on plan correctness or executability. We propose a hybrid framework that assigns each component the role it is suited for: given the current environment state (represented as a scene graph) and a high-level task description, an LLM infers a set of grounded symbolic subgoals, which are then verified against the planning domain and solved by an off-the-shelf symbolic planner. We hypothesize that decoupling *goal inference* from *plan generation* lets a robot complete a broader class of long-horizon tasks while retaining the reliability of planning-based execution for the steps it produces. We will evaluate the approach in a simulated kitchen–dining domain using task-completion rate, planning cost, planning time, and agreement between inferred and ground-truth goals, and we plan a real-world demonstration that mirrors the simulated domain.

I. INTRODUCTION

Household and service robots, including humanoid robots intended to act in spaces built for people, are increasingly expected to perform long-horizon tasks from natural language instructions. Unlike traditional planning problems [1], [2], [3], [4], [5], [6], [7], [8], human instructions are often *underspecified*, describing a desired activity without explicitly specifying the goal state required to complete it. For example, a user may instruct a robot to *set a dining table* or *make dinner*. Humans readily infer the intermediate objectives such tasks require, but robots typically need goals to be specified explicitly before planning can begin.

Consider *set a dining table* in a household domain. A person naturally infers that this involves placing clean plates and bowls on the table, arranging the chairs, and clearing dirty cups to the sink—yet none of these objectives appear in the instruction itself. A robot must bridge this gap between human intent and executable planning goals before it can produce a valid plan. Closing that gap is precisely what makes a robot’s behavior feel humanlike: the system supplies the unstated objectives a person would have assumed.

This problem is challenging because interpreting underspecified instructions relies largely on common sense. Humans draw on extensive background knowledge about object usage, social conventions, and task structure when reading

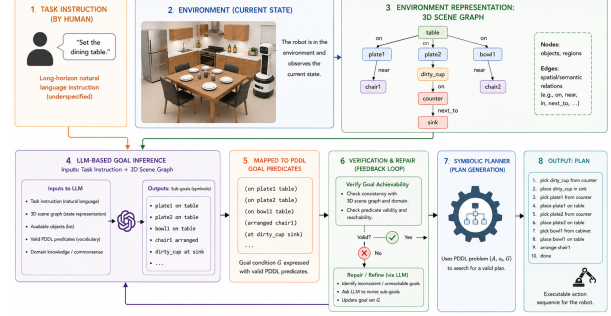


Fig. 1: Schematic of our proposed approach. An LLM infers grounded symbolic goals from a scene graph and an underspecified instruction; the goals are verified against the planning domain and solved by a symbolic planner. Figure drawn with assistance from ChatGPT [9].

high-level requests. Encoding such knowledge explicitly within symbolic planning models is difficult and scales poorly to realistic household environments.

Traditional planning approaches assume the desired goal state is already known and therefore cannot reason about what a user intended when issuing an underspecified instruction. Recent commonsense reasoning frameworks and knowledge bases provide semantic information about objects and activities [10], [11], but they typically depend on manually curated knowledge and predefined task structures.

Recent advances in LLMs and vision language models (VLM)s [12], [13], [14], [15] have created new opportunities for connecting human instructions to robot planning. While LLMs possess broad semantic and common-sense knowledge that lets them infer plausible interpretations of underspecified instructions, they are not designed to reason explicitly about action preconditions, effects, and environmental constraints, and so are better suited to proposing candidate goals than to producing reliable plans [16]. This motivates a hybrid design in which an LLM infers grounded planning goals while a symbolic planner remains responsible for generating valid action sequences.

In this work, we propose an approach in which an LLM infers grounded planning goals from underspecified human instructions. To isolate the goal-inference problem from perception, we represent the environment as a scene graph—a sparse semantic description of objects and their relationships [17], [18], [19] that integrates directly with symbolic planning. Given the scene graph and a high-level task description, the LLM produces a set of grounded symbolic goals capturing the intended outcome; these are

validated against the planning domain, incorporated into a planning problem, and solved by a symbolic planner to yield an executable action sequence. By pairing the common-sense reasoning of LLMs with the reliability of classical planning, the framework aims to bridge human intent and robot execution.

We will evaluate the approach in a simulated kitchen–dining domain built in PyBullet [20] and inspired by benchmark domains [21], [22], involving long-horizon manipulation tasks. We compare against both classical planners requiring manually specified goals and existing LLM-based task-decomposition approaches, measuring task-completion rate, planning cost, planning time, and agreement between inferred goals and human expectations. Finally, we demonstrate the approach on a real robot in a setting that mirrors the simulated domain.

II. RELATED WORK

Task Planning for Long-Horizon Tasks Classical planning approaches have demonstrated success in generating action sequences that achieve specified goals using symbolic reasoning, hierarchical decomposition, and integrated task and motion planning [1], [23], [24], [4], [25]. Benchmark environments such as ALFRED [15], BEHAVIOR [26], and VirtualHome [11] have further motivated research on long-horizon household tasks. These approaches, however, generally assume the desired goal state is specified before planning begins and therefore cannot operate directly from underspecified human instructions.

LLM-Based Planning and Commonsense Reasoning LLMs have enabled new ways of injecting common-sense knowledge into robot planning, including skill selection [27], task decomposition [14], policy generation [28], execution-time reasoning [29], and symbolic planning augmentation [13]; LLM-GROP [12] and AHAT [30] ground abstract household instructions into task specifications. Closest to our work, LLM+P [31] translates a problem description into a Planning Domain Definition Language (PDDL) problem for an external planner, and Guan et al. [32] use LLMs to construct PDDL world models. These methods, however, assume the goal is stated in the instruction. We instead target the *underspecified* setting, where the goal must be *inferred* from the scene and a high-level description, restricting the LLM to goal inference and leaving plan generation to a symbolic planner.

III. PROPOSED APPROACH

A. Preliminaries

To achieve high-level goals—e.g., clearing a table—robots must carry out high-level actions such as picking up a bowl and moving through the environment. We represent both the tasks and the actions available to the robot using the PDDL [33]. PDDL defines a fully observable, deterministic planning problem as a tuple (A, s_0, G) , where A is a set of parameterized actions, s_0 is the initial state, and G is the goal condition.

Consider a simple kitchen domain where a robot moves plates from a table to a dishwasher. The symbolic model encodes predicates such as `on-table` and `holding`, and actions such as `pickup` and `place`. The domain in Fig. 2 indicates whether a plate is on the table or in the dishwasher and whether the robot is holding an object or has an empty arm; the accompanying problem instance declares the objects, initial facts, and goal condition.

```
(define (domain kitchen)
  (:requirements :strips :equality)
  (:predicates
    (on-table ?x)
    (on-dishwasher ?x)
    (arm-empty)
    (holding ?x))
  (:action pickup
    :parameters (?ob)
    :precondition (and
      (on-table ?ob)
      (arm-empty))
    :effect (and
      (holding ?ob)
      (not (on-table ?ob))
      (not (arm-empty))))
  (:action place
    :parameters (?ob)
    :precondition (holding ?ob)
    :effect (and
      (arm-empty)
      (on-dishwasher ?ob)
      (not (holding ?ob))))
)

(define (problem pick_place)
  (:domain kitchen)
  (:objects plate1 plate2)
  (:init
    (on-table plate1)
    (on-table plate2)
    (arm-empty))
  (:goal (and
    (on-dishwasher plate1)
    (on-dishwasher plate2))))
)
```

Plan:

1. move robot to table
2. pickup plate1
3. place plate1
4. pickup plate2
5. place plate2

Fig. 2: PDDL domain, problem instance, and corresponding plan for a simple kitchen pick-and-place task.

The planner searches for an action sequence $\pi = \langle a_1, a_2, \dots, a_n \rangle$ that transforms s_0 into a state satisfying G , as shown in Fig. 2. Our approach uses the Fast Downward [34] planning system.

Unlike traditional planning problems, we consider an *underspecified* task description τ provided by a human user. The description τ does not explicitly specify the goal G , so the robot must infer planning objectives before planning can begin.

B. Grounding Underspecified Tasks into Planning Goals

We consider executing an underspecified task description τ in an environment with state s_0 . Given a planning domain in PDDL, we assume the available actions A and the initial state s_0 are known. Our objective is to infer a symbolic goal specification G from the pair (s_0, τ) such that the resulting planning problem $P = (A, s_0, G)$ captures the user’s intended outcome.

To do so, an LLM receives the current environment state (as a scene graph) and the task description, and generates a set of grounded symbolic predicates defining the goal condition G . The inferred predicates are restricted to the vocabulary of the planning domain so that they can be incorporated directly into the PDDL problem. An example prompt is shown in Listing 1.

Restricting the LLM to goal inference rather than plan generation lets the framework use common-sense reasoning to interpret underspecified instructions while keeping plan synthesis under the planner’s control. The reliability guarantee, however, applies only to the plan *relative to the inferred goal*: a planner returns a sound plan for G , but

it cannot detect whether G itself misrepresents the user’s intent. We therefore introduce a verification step between inference and planning. Each inferred predicate is checked against the domain’s predicate and object vocabulary and discarded if malformed; the candidate goal G is then tested for satisfiability and reachability from s_0 . If the planner reports the problem unsolvable or a constraint is violated, the failure signal is returned to the LLM for a bounded number of repair attempts. This loop keeps an erroneous inference from silently producing a valid-but-wrong plan and makes the separation of inference from execution robust in practice. Figure 1 illustrates the overall architecture.

Listing 1: Example prompt for goal inference.

You are a robot planning assistant.

Input:

- * Current environment state represented as a scene graph.
- * A list of available objects.
- * A list of valid PDDL predicates.
- * A task description.

Task:

Infer the intended task outcome and return ONLY symbolic goal predicates using the provided predicate vocabulary.

Task Instruction:

”Set the dining table”

Output:

```
(on plate1 table)
(on plate2 table)
(on bowl1 table)
(arranged chair1)
(at dirty_cup sink)
```

Once verified, the predicates defining G are incorporated into the PDDL problem, and the symbolic planner returns a plan $\pi = \langle a_1, a_2, \dots, a_n \rangle$ transforming the environment from s_0 into a state satisfying G .

C. Scene Graph Representation

To isolate goal inference from perception, we represent the environment as a 3D scene graph [17], [18], [19]. The scene graph $S = (V, E)$ comprises object and region nodes V connected by semantic and spatial relationships E , such as *on* or *inside*. This gives a compact symbolic description that is naturally compatible with PDDL predicates and can be consumed directly by the LLM for goal inference.

IV. PROPOSED EXPERIMENTS

We will evaluate the approach in a simulated kitchen–dining environment using tasks derived from the BEHAVIOR-1K benchmark [35], which contains a diverse collection of everyday household activities. The robot is given the current environment state and a high-level natural language instruction. We will report results over N tasks ($N \geq 50$) with multiple random seeds per task.

We compare the proposed approach against two baselines.

TABLE I: Evaluation metrics.

Metric	Description
Success Rate	Completed tasks / total tasks
Goal Precision/Recall	Inferred vs. ground-truth goal predicates
Invalid-Goal Rate	Malformed or unsatisfiable inferred goals
Planning Cost	Cost of the generated plan
Planning Time	Planning runtime
Goal Agreement	Human rating

Oracle Planner A symbolic planner given manually specified goals representing the intended outcome. It serves as an upper bound on planning performance when the correct goal is known.

ProgPrompt An LLM-based approach [14] that generates task decompositions directly from natural language, evaluating language-model task specification *without* explicit symbolic goal inference.

LLM+P An LLM-based approach [31] that translates a natural language problem description into a complete PDDL problem for an external planner. Unlike our method, it assumes the goal is stated in the instruction.

Ours The proposed approach, which uses an LLM to infer grounded symbolic goals from the *underspecified* instruction, verifies them, and solves the resulting planning problem with a symbolic planner.

We evaluate at two levels. At the *goal* level we measure how well the inferred goal matches intent, since this is the component our method introduces: goal precision and recall against ground-truth goals, the rate of malformed (out-of-vocabulary) predicates, and the fraction of inferred goals that are unsatisfiable or unreachable from s_0 . At the *plan/execution* level we measure task-completion rate, planning cost, and planning time. We additionally collect a human judgment of goal agreement. Human evaluators will be presented with a task description and the corresponding inferred goals and asked whether the generated goals accurately capture the intended task outcome. Metrics are summarized in Table I.

We will also measure how the underlying model affects goal inference by substituting GPT-4o [36], Claude Sonnet [37], and an open-source model such as Qwen-VL [38], holding the rest of the pipeline fixed.

Finally, we demonstrate the approach on a real robot in a kitchen–dining environment. The robot receives a natural language instruction, infers and verifies planner-ready goals with an LLM, and executes the plan produced by the symbolic planner, evaluating practicality in a realistic household setting.

V. CONCLUSION

We propose a framework for executing underspecified long-horizon tasks by combining LLM-based goal inference with symbolic planning. Given a natural language instruction and the current environment state, an LLM infers grounded symbolic goals that are incorporated into a PDDL planning problem and solved using a classical planner. By separating goal inference from plan generation, the proposed approach leverages the common-sense reasoning capabilities of LLMs

while preserving the reliability and correctness guarantees of symbolic planning. We will evaluate the approach on household manipulation tasks in both simulation and will demonstrate in real-world scenario reflecting our simulated domain.

REFERENCES

- [1] C. R. Garrett, R. Chitnis, R. Holladay, B. Kim, T. Silver, L. P. Kaelbling, and T. Lozano-Pérez, “Integrated task and motion planning,” *Annual review of control, robotics, and autonomous systems*, vol. 4, pp. 265–293, 2021.
- [2] E. Plaku and G. D. Hager, “Sampling-based motion and symbolic action planning with geometric and differential constraints,” in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 5002–5008.
- [3] M. Toussaint, “Logic-geometric programming: An optimization-based approach to combined task and motion planning,” in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.
- [4] L. P. Kaelbling and T. Lozano-Pérez, “Hierarchical task and motion planning in the now,” *IEEE International Conference on Robotics and Automation*, pp. 1470–1477, 2011.
- [5] S. Srivastava, E. Fang, L. Riano, R. Chitnis, S. Russell, and P. Abbeel, “Combined task and motion planning through an extensible planner-independent interface layer,” in *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2014, pp. 639–646.
- [6] B. Kim, Z. Wang, L. P. Kaelbling, and T. Lozano-Pérez, “Learning to guide task and motion planning using score-space representation,” *The International Journal of Robotics Research*, vol. 38, no. 7, pp. 793–812, 2019.
- [7] R. Chitnis, D. Hadfield-Menell, A. Gupta, S. Srivastava, E. Groshev, C. Lin, and P. Abbeel, “Guided search for task and motion plans using learned heuristics,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 447–454.
- [8] N. T. Dantam, Z. K. Kingston, S. Chaudhuri, and L. E. Kavraki, “Incremental task and motion planning: A constraint-based approach,” in *Robotics: Science and systems*, vol. 12. Ann Arbor, MI, USA, 2016, p. 00052.
- [9] OpenAI, “Chatgpt,” <https://chatgpt.com>, 2025, large language model.
- [10] M. Tenorth and M. Beetz, “Knowrob: A knowledge processing infrastructure for cognition-enabled robots,” *International Journal of Robotics Research*, vol. 32, no. 5, pp. 566–590, 2013.
- [11] X. Puig, T. Shu, S. Li, Z. Wang, S. F. Liao, and A. Torralba, “Virtuallhome: Simulating household activities via programs,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.
- [12] X. Zhang, Y. Ding, Y. Hayamizu, Z. Altaweel, Y. Zhu, Y. Zhu, P. Stone, C. Paxton, and S. Zhang, “Llm-grop: Visually grounded robot task and motion planning with large language models,” *The International Journal of Robotics Research*, 2025.
- [13] W. Liu *et al.*, “Language-augmented symbolic planning for open-world robot manipulation,” in *Robotics: Science and Systems*, 2024.
- [14] I. Singh, V. Blukis, A. Mousavian, A. Goyal, D. Xu, J. Tremblay, and D. Fox, “Progprompt: Generating situated robot task plans using large language models,” *IEEE International Conference on Robotics and Automation*, 2023.
- [15] M. Shridhar, J. Thomason, D. Gordon, Y. Bisk, W. Han, R. Mottaghi, D. Fox, and A. Farhadi, “Alfred: A benchmark for interpreting grounded instructions for everyday tasks,” in *CVPR*, 2020.
- [16] K. Valmeekam, M. Marquez, A. Olmo, S. Sreedharan, and S. Kambhampati, “Planbench: An extensible benchmark for evaluating large language models on planning and reasoning about change,” *Advances in Neural Information Processing Systems*, 2023.
- [17] I. Armeni, Z.-Y. He, J. Gwak, A. R. Zamir, M. Fischer, J. Malik, and S. Savarese, “3d scene graph: A structure for unified semantics, 3d space, and camera,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.
- [18] A. Rosinol, A. Gupta, M. Abate, J. Shi, and L. Carlone, “3d dynamic scene graphs: Actionable spatial perception with places, objects, and humans,” *arXiv preprint arXiv:2002.06289*, 2020.
- [19] A. Nguyen, D. Kanoulas, D. G. Caldwell, and N. G. Tsagarakis, “3-d scene graph: A sparse and semantic representation of physical environments for intelligent agents,” *IEEE Transactions on Cybernetics*, vol. 50, no. 12, pp. 4921–4933, 2020.
- [20] E. Coumans and Y. Bai, “PyBullet, a Python module for physics simulation for games, robotics and machine learning,” 2016–2021.
- [21] F. Lagriffoul, N. T. Dantam, C. Garrett, A. Akbari, S. Srivastava, and L. E. Kavraki, “Platform-independent benchmarks for task and motion planning,” *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3765–3772, 2018.
- [22] Z. Yang, C. R. Garrett, T. Lozano-Pérez, L. Kaelbling, and D. Fox, “Sequence-based plan feasibility prediction for efficient task and motion planning,” in *Proceedings of Robotics: Science and Systems (RSS)*, Daegu, Republic of Korea, July 2023.
- [23] M. Ghallab, D. Nau, and P. Traverso, *Automated Planning: Theory and Practice*. Morgan Kaufmann, 2004.
- [24] D. Nau, T.-C. Au, O. Ilghami, U. Kuter, J. W. Murdock, D. Wu, and F. Yaman, “Shop2: An htn planning system,” in *Journal of Artificial Intelligence Research*, vol. 20, 2003, pp. 379–404.
- [25] C. R. Garrett, T. Lozano-Pérez, and L. P. Kaelbling, “Pddlstream: Integrating symbolic planners and blackbox samplers via optimistic adaptive planning,” *International Conference on Automated Planning and Scheduling*, 2020.
- [26] S. Srivastava, C. Li, M. Lingelbach, J. Choi *et al.*, “Behavior: Benchmark for everyday household activities in virtual, interactive, and ecological environments,” *CoRL*, 2021.
- [27] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, C. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman *et al.*, “Do as i can, not as i say: Grounding language in robotic affordances,” *arXiv preprint arXiv:2204.01691*, 2022.
- [28] J. Liang, W. Huang *et al.*, “Code as policies: Language model programs for embodied control,” *IEEE International Conference on Robotics and Automation*, 2023.
- [29] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch, “Inner monologue: Embodied reasoning through planning with language models,” *Conference on Robot Learning*, 2023.
- [30] Z. Liu, Y. Li, R. Huang, C. Lu, and P. Cai, “Any house any task: Scalable long-horizon planning for abstract human tasks,” *arXiv preprint arXiv:2602.12244*, 2026.
- [31] B. Liu, Y. Jiang, X. Zhang, Q. Liu, S. Zhang, J. Biswas, and P. Stone, “Llm+π: Empowering large language models with optimal planning proficiency,” *arXiv preprint arXiv:2304.11477*, 2023.
- [32] L. Guan, K. Valmeekam, S. Sreedharan, and S. Kambhampati, “Leveraging pre-trained large language models to construct and utilize world models for model-based task planning,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 79 081–79 094, 2023.
- [33] M. Fox and D. Long, “PDDL2.1: An extension to PDDL for expressing temporal planning domains,” *Journal of Artificial Intelligence Research*, vol. 20, pp. 61–124, 2003.
- [34] M. Helmert, “The Fast Downward planning system,” *Journal of Artificial Intelligence Research*, vol. 26, pp. 191–246, 2006.
- [35] C. Li, R. Zhang, J. Wong, C. Gokmen, S. Srivastava, R. Martín-Martín, C. Wang, G. Levine, W. Ai, B. Martinez, H. Yin, M. Lingelbach, M. Hwang, A. Hiranaka, S. Garlanka, A. Aydin, S. Lee, J. Sun, M. Anvari, M. Sharma, D. Bansal, S. Hunter, K.-Y. Kim, A. Lou, C. R. Matthews, I. Villa-Renteria, J. H. Tang, C. Tang, F. Xia, Y. Li, S. Savarese, H. Gweon, C. K. Liu, J. Wu, and L. Fei-Fei, “Behavior-1k: A human-centered, embodied ai benchmark with 1,000 everyday activities and realistic simulation,” *arXiv preprint arXiv:2403.09227*, 2024.
- [36] OpenAI, “Hello gpt-4o,” OpenAI Technical Report, 2024. [Online]. Available: <https://openai.com/index/hello-gpt-4o/>
- [37] Anthropic, “Introducing claude 4,” Anthropic Technical Report, 2025. [Online]. Available: <https://www.anthropic.com/news/claude-4>
- [38] P. Wang, S. Bai *et al.*, “Qwen2-vl: Enhancing vision-language model perception of the world at any resolution,” 2024.