# Learning Task Planning from Multi-Modal Demonstration for Multi-Stage Contact-Rich Manipulation

Kejia Chen[*1], Zheng Shen[*1], Yue Zhang[1], Lingyun Chen[1],
Fan Wu[1], Zhenshan Bing[1], Sami Haddadin[1], Alois Knoll[1]

*Abstract*—Large Language Models (LLMs) have gained popularity in task planning for long-horizon manipulation tasks. To enhance the validity of LLM-generated plans, visual demonstrations and online videos have been widely employed to guide the planning process. However, for manipulation tasks involving subtle movements but rich contact interactions, visual perception alone may be insufficient for the LLM to fully interpret the demonstration. Additionally, visual data provides limited information on force-related parameters and conditions, which are crucial for effective execution on real robots.

In this paper, we introduce an in-context learning framework that incorporates tactile and force-torque information from human demonstrations to enhance LLMs' ability to generate plans for new task scenarios. We propose a bootstrapped reasoning pipeline that sequentially integrates each modality into a comprehensive task plan. This task plan is then used as a reference for planning in new task configurations. Real-world experiments on two different sequential manipulation tasks demonstrate the effectiveness of our framework in improving LLMs' understanding of multi-modal demonstrations and enhancing the overall planning performance.

## I. INTRODUCTION

Recent advances in Large Language Models (LLMs) [1], [2] and Visual-Language Models (VLMs) [3] have triggered a paradigm shift in the domain of long-horizon task planning. By leveraging the semantic understanding and reasoning capabilities of LLMs, immense progress has been made towards developing task-agnostic high-level task planner [4], [5]. The majority of prior works in LLM-based task planning use in-context learning techniques where carefully designed examples are employed to generate high-level task plans [6]–[9], and leverage visual information for skill grounding.

On the other hand, learning from demonstrations (LfD) offers an alternative that mitigates the need for prompting examples. LfD can automatically extract example task plans from demonstrations, eliminating the need for human experts to explicitly construct these plans. The key questions in the LfD approach for sequential manipulation planning are: i) how to segment demonstrations into reusable task plans consisting of skill sequences, and ii) how to ground the associated skills, which involves determining the necessary task-agnostic information to make skills executable. Since foundation models are pre-trained on large-scale internet data, it is natural to utilize visual demonstrations [10] or human play videos [11] to address these challenges.

However, visual information alone may be insufficient for perceiving and expressing contact-rich manipulations where the movement of objects is barely observable, yet changes

[1] School of Computation, Information and Technology, Technical University of Munich, Germany.

[*] Equal contribution. Correspondence to: Fan Wu (f.wu@tum.de)
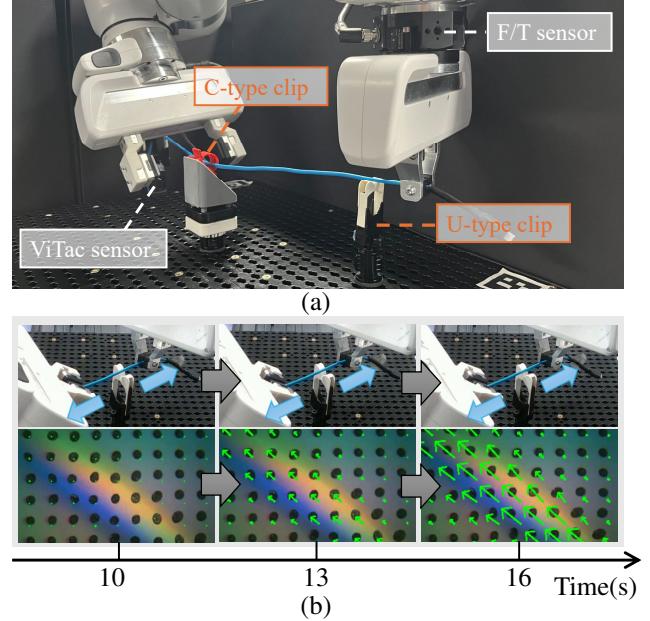Project site: sz-123.github.io/ContactRichPlanningLLM/

Fig. 1. Cable Manipulation with Two Robots. (a) Setup for cable mounting demonstration. The human operator controls robots to mount the cable onto a C-type clip and a U-type clip. Because of structural differences, the C-type clip expects a larger pushing force during insertion. (b) Multi-modal perception during cable stretching. Upper: camera observations. Lower: visual-tactile images on robot fingertip. The green arrows in ViTac images indicate force vectors, demonstrating a linear force applied along the grasped cable during the stretching process.

in force contact are significant. For example, in the cable mounting process shown in Fig. 1, humans intuitively stretch the cable to create tension, thereby easing the subsequent insertion. As illustrated in Fig. 1 (b), while camera-based observations (upper) struggle to capture any movement, images from visual-tactile sensors (lower) clearly reveal a consistent pattern of pulling force. Even when the stretching step is identified, visual observation alone struggles to define an appropriate success condition for this step. In such scenarios, force and tactile information play pivotal roles in detecting and understanding such "invisible" events. This example leads to a key insight that motivates our work: tactile information is important for both task segmentation, which requires multi-modal semantic reasoning to successfully convert observed long-horizon tasks into a sequence of actions associated with executable robot skills, and skill grounding, which extracts necessary information for chaining and executing the skills, such as task specifications and pre/post-conditions.

In this paper, we introduce **a novel in-context learning framework** that enables pre-trained LLMs to learn task

planning from multi-modal demonstration data, with an emphasis on utilizing both visual and tactile information for reasoning about demonstrations and grounding skill conditions. One single demonstration for each task is provided through teleoperation using a haptic device, where tactile sensing from ViTac sensors and force/torque (F/T) signals are collected alongside camera video recordings. Each modality is then integrated in a bootstrapped manner into the reasoning process of an LLM analyzer to segment the demonstration into skill sequences, and establish reliable transition conditions between skills. Subsequently, the resulting sequence and grounded skills serve as an example task plan for an LLM planner, enabling it to generate new plans for varying task configurations.

We evaluate our approach on two challenging, multi-step, contact-rich manipulation tasks: cable assembly and bottle cap tightening. The effectiveness of the proposed framework is validated by experimental results, demonstrating a high success rate in generalizing to novel task configurations. Ablation studies further highlight (1) the significant performance improvements in both reasoning and planning when compared to cases without demonstrations or using only visual demonstrations, (2) the critical role of grounding tactile-based skill conditions for successful reproduction of learned task plan in varied task configurations. These results underscore the importance of incorporating tactile and F/T information, enabling LLMs to effectively understand and learn complex contact-rich manipulation tasks.

## II. RELATED WORK

**LLMs as Task Planner**. LLMs have increasingly been used for symbolic task planning due to their strong semantic understanding and reasoning abilities [4], though their generated actions aren't always executable. To address this, early studies guided LLM plans using predefined action sets and incorporated pre- and post-conditions for better grounding [12]. Structured approaches like PDDL [9], [13], behavior trees [14], [15], and task trees [16] are now commonly applied to improve plan reliability. Building on this, we also use PDDL for refining a skill library and reasoning about skill sequences. Another approach to enhancing LLMs/VLMs for task planning is using demonstration datasets. For instance, PaLM-E employs task-and-motion planners to generate extensive planning examples [5]. Other methods use human demonstrations for symbolic task plans but are limited to trajectory-based actions [10]. Overall, these efforts rely on large datasets and are generally restricted to tasks with limited contact interactions.

**Learning Task Planning from Few Demonstrations**. Classic methods like Hidden Markov Models and Dynamic Motion Primitives learn task planning from limited demonstrations by segmenting tasks into subtasks and learning structured plans chained by subtasks or primitive actions [17]–[21]. However, these approaches often require extensive feature engineering and struggle to generalize to new tasks. While imitation learning with deep neural networks has made vast progress in learning and generalizing multi-stage tasks [22]–[25], it suffers from the high cost of collecting large demonstration datasets. Leveraging the multi-modal

reasoning capability of pre-trained foundation models, LLM-enabled task planning has improved generalization with fewer demonstrations. For instance, GPT-4-based planners achieve one-shot planning from human demonstrations [10], and hierarchical structures have been proposed to maximize knowledge distillation from these demonstrations [26]. Other methods use task conditions generated by LLMs to guide generalization during execution [12]. Our work extends this line of research by focusing on how LLMs can effectively utilize demonstrations for contact-rich manipulation task planning.

**Multi-Modal Sensory Data and LLMs**. Recent advances in multi-modal LLMs, such as GPT-4V [1], have exhibited significant capabilities in scene understanding. However, the current paradigm predominantly focuses on large-scale pretraining for language-conditioned visual representations, while comparatively few studies have investigated other sensory modalities and their potential applications in robotics.

The GenCHiP framework allows LLMs to reason about motion and force by exposing constraints on contact stiffness and forces in control interface [27], aiming to automate parameter tuning for contact-rich manipulation tasks. Other efforts enhance LLMs' reasoning with tactile data by fine-tuning models on specialized datasets that combine paired tactile-visual observations with tactile-semantic labels [28]. Another approach uses contact microphones as tactile sensors to leverage large-scale audio-visual pretraining, addressing the scarcity of non-visual data in low-data robotic applications [29]. In contrast, our approach minimizes the need for large-scale multi-modal datasets by bootstrapping LLMs with a single human demonstration, integrating ViTac images, F/T signals, and standard camera videos.

## III. METHODOLOGY

As shown in Fig. 2, our framework first derives a task plan from the demonstration (highlighted in the orange box), and then uses this plan as a reference for planning new, generalized tasks (highlighted in the green box). Prior to this, we collect ViTac data from the robot's fingers, third-person camera images, and F/T measurements at the robot's end-effector during kinesthetic teaching. In the reasoning stage, in addition to the skill library, we provide the LLM analyzer with each sensory modality in a bootstrapped manner, enabling it to derive from the demonstration a task plan that describes the demonstration as well as a skill library with transition conditions updated. When a new task is requested, the LLM planner uses the demonstration task plan as an example to generate new task plans, adjusting them interactively based on the robots' execution feedback.

In the following section, we first introduce the formulation of our object-centric skill library. Next, we explain each stage of the bootstrapped reasoning pipeline, presenting how each modality is integrated. Finally, we describe how the generated task plan is utilized for planning on new tasks.

### A. Object-Centric Skills

In contact-rich manipulation tasks, such as assembly, robot actions are typically highly object-centric. That is to say, each robot action is designed to change a certain status of an object, whether by changing its position or applying an
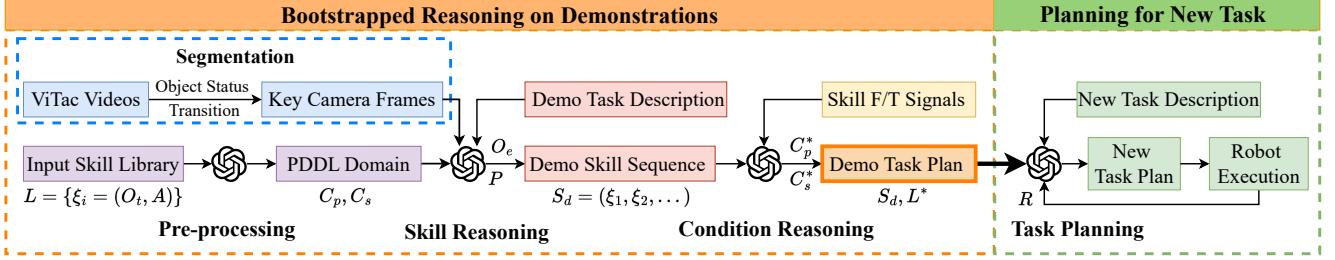
Fig. 2. Framework Overview. In bootstrapped reasoning, an LLM analyzer pre-processes the skill library, reasons about skill sequences and success conditions from multi-modal demonstration sequentially. The resulting demo task plan is used as an example for an LLM planner to plan for new tasks.
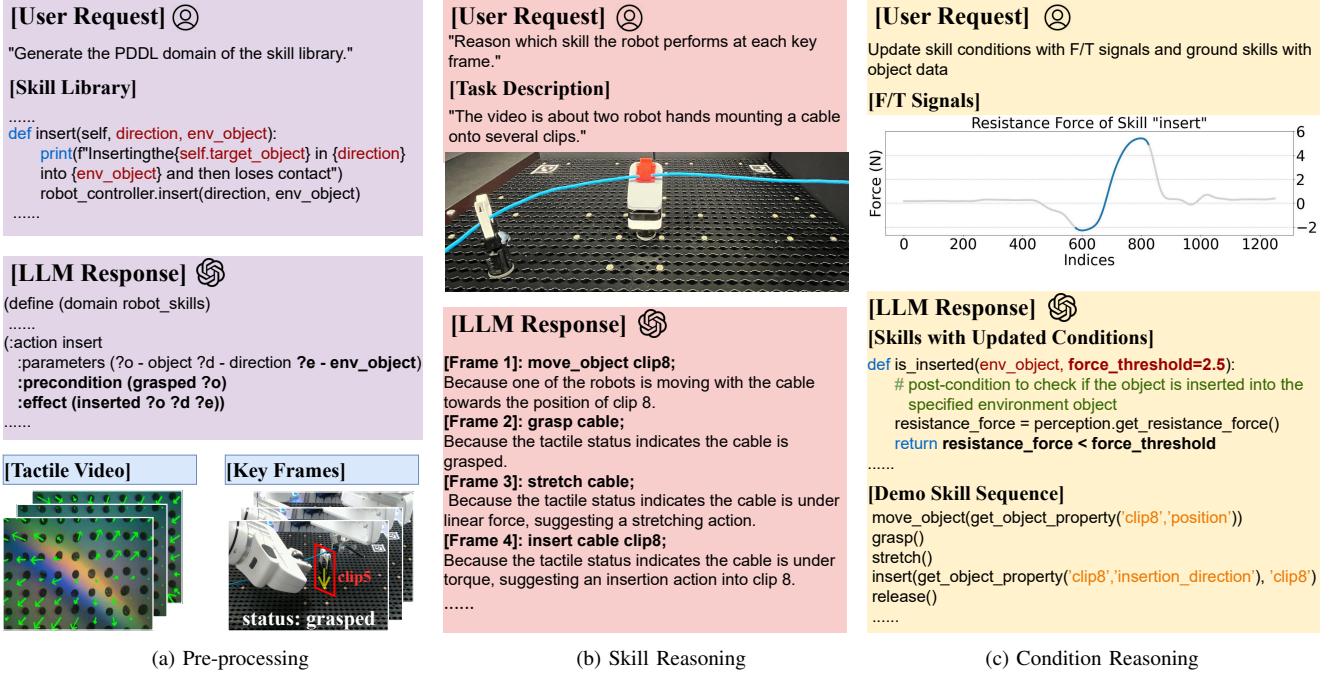


Fig. 3. Overview of Prompts and Responses in Bootstrapped Reasoning. More prompts can be found on our project website.

external force. In this work, we focus on two key aspects of object status, namely, positional status (which includes the object's position and orientation) and interaction status (which describes whether the object is grasped, released, or subjected to an external force or torque).

Inspired by this fact, we construct our skill library in an object-centric manner, where each skill is defined to describe a status change of the being-manipulated object (called target object). In this way, the formulation of skills remains consistent whether executed by a human or a robot, enabling LLMs to generalize human demonstrations into executable task plans in new task configurations. Additionally, we utilize the target object status as the bridge between raw sensory data from demonstrations and the grounding of LLM, which will be discussed in detail in Section III-B.

**Input Skill Library.** Each skill in our skill library is formulated as a tuple $\xi = (O_t, O_e, C_p, C_s, A, R)$. $O_t$ represents the target object which robot directly manipulates, such as a cable or a cap. $O_e$ represents the contextual object with which the $O_t$ interacts under the manipulation of robots, such as a cable clip or a bottle. The description and examples of all components in $\xi$ are summarized in Table I.

Our skill library $L = \{\xi_i\}$, serving as input to the LLM analyzer, is defined in the form of executable code scripts.

TABLE I Skill Formulation

|  | Description | Example | PDDL |
|---|---|---|---|
| $O_t$ | **Target object** that the robot directly manipulates. | cable | `parameters` |
| $O_e$ | **Contextual object** with which the $O_t$ interacts. | clip | `parameters` |
| $A$ | Executed **Action** by low-level robot controller. | insertion | `action` |
| $C_p$ | **Pre-condition** to be met for the skill to start execution. | cable positioned near a clip | `precondition` |
| $C_s$ | **Success condition** that defines when the skill is finished. | cable inserted into a clip | `effect` |
| $R$ | **Return** of a skill execution. | success (when $C_s$ is satisfied) or error | N/A |
| $P$ | **Skill parameters**, such as force thresholds or positions. | position of the clip | `parameters` |

Initially, we include object-agnostic skills, such as `move`, in a general `ObjectSkillLibrary`. These skills can be applied to various objects and serve as the foundation for object-specific libraries, which contain affordances of specific objects. For example, the `CableSkillLibrary` includes a "cable-centric" `insertion` skill (see Fig. 3(a)), which handles inserting a cable into other objects, such as clips. The definition of each skill provides explicitly $O_t$ and calls the low-level `robot_controller` interface to perform an action $A$. The `robot_controller` controls the robot
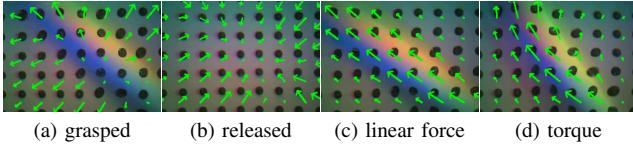
Fig. 4. Tactile Signal Patterns and Corresponding Object Statuses: (a) Sourcing pattern, referring to "grasped" status; (b) Sinking, referring to "released" status; (c) Uniform Flow, referring to "under a linear force" status; (d) Twisted Flow, referring to "under torque" status.

to move or apply force/torque via an adaptive impedance controller [30]. In the following procedures, we will guide the LLM to complete the remaining components in skill $\xi$ step by step.

**Pre-processing Translation.** While the original formatting of code scripts allows skills to be executed by robots, it lacks the logic to support effective reasoning, especially conditions for skill transitions. Inspired by prior works [9], [13], we ask the LLM analyzer to translate the input skill library into a PDDL domain as a preprocessing step. As a standard language to describe planning problems, PDDL provides a structured syntax to represent rules like actions, objects, and conditions in its domain knowledge, which closely aligns with the composition of our input skill library. A comparison between our skill library and the corresponding PDDL domain components is summarized in Table I. Through translation, we aim to leverage PDDL's standard format to enhance LLMs' understanding of skills and the ability to plan for long-horizon tasks. As is shown in Fig. 3(a), translation into PDDL encourages the LLM analyzer to automatically complete the transition conditions $C_p$ (precondition in PDDL) and $C_s$ (effect in PDDL).

### B. Bootstrapped Reasoning of Demonstration

Given the large volume and multi-modal nature of our demonstration data, it can be challenging for LLMs to interpret all modalities simultaneously. To address this problem, we adopt a bootstrapped approach for in-context learning of demonstration, where each modality is introduced sequentially to assist the LLM in different stages of reasoning. Tactile information is utilized to segment events from visual perception and identify object statuses, which then enables LLMs to comprehend the entire demonstration and infer the corresponding skill sequence $S_d = (\xi_1, \xi_2, ...)$. Afterwards, F/T signals are leveraged to ground and refine the transition conditions between skills ensuring that the learned skill sequence is executable and can generalize to new task scenarios by re-planning.

**Segmentation with Object Status.** To facilitate skill reasoning from demonstration data, we first segment the entire demonstration into events where object status changes. We rely on tactile status for this segmentation, as tactile information on robot fingers directly precepts the manipulation on target object $O_t$, thus capturing details that are often missed by third-person cameras.

Fig. 4 illustrates four typical patterns of Vi-Tac images, each reflecting different interaction status of the target object: a) *Sourcing*: Vectors point outward, indicating a source where force radiates outward. This pattern is usually observed when an object is "grasped". b) *Sinking*: Vectors point inward, showing a sink where force converges inward, usually when

a grasped object is "released". c) *Uniform Flow*: Vectors are parallel and evenly spaced, representing a consistent force in one direction. This pattern happens when an object is "under a linear force", such as being pushed or pulled. d) *Twisted Flow*: Vectors twist gently, suggesting rotational movement around a central point. This pattern happens when an object is "under torque", such as being rotated or screwed.

To identify the above tactile patterns in demonstrations, we fine-tune the video classifier *TimeSformer* [31] with a dataset of labeled tactile videos. Applying this classifier to the complete demonstration then segments it into events when new interaction happens. Object status alongside the timestamps of these events (referred to as key timestamps) are extracted for subsequent reasoning.

**Reasoning Skill Sequence** After segmentation, we provide the LLM analyzer with camera images taken at key timestamps (referred to as key frames) for skill reasoning. As is shown in Fig. 3 (a), each key frame is annotated with objects as well as the the corresponding status of $O_t$. Additionally, we include in the prompt a simple description of the task (e.g., "two robots are mounting a cable onto several clips") and of the objects (e.g., "the blue curve in the view is the cable"). The LLM is then asked to reason which skill from the PDDL domain the demonstrator has performed at each key timestamp.

Fig. 3(b) presents an example of the skill sequence reasoned by the LLM. We observe that the LLM adaptively utilizes camera images (for `move_object`) and object status (for `grasp`, `stretch`, and `insert`) to infer the corresponding skills. In addition, the LLM also fills in the contextual object $O_e$ (`env_object`) for each skill in its reasoning.

**Reasoning Skill Conditions** The final step in converting the skill sequence into an executable task plan is to reason about success conditions for each skill. We leverage the `precondition` and `effect` in the previously translated PDDL domain. The LLM is requested to implement these conditions and integrate them back into the skill library. Additionally, we provide the LLM with interfaces to access perception information about robot pose, grasping status, and F/T signals. Since most of the other information is either binary or straightforward when used to form conditions (e.g. whether an object is grasped or a position is reached), we focus especially on F/T conditions which are highly variable and crucial for contact-rich manipulations.

The raw six-dimensional F/T signals are complex for the LLM to interpret directly. To address this without sacrificing generality, we assume that the task is performed in a static environment where interactions with the object occur exclusively through the robot. In this context, the most relevant F/T information pertains to the force or torque opposing the robot's actions, as they provide direct feedback on the resistance encountered during manipulation. Based on this observation, we reduce our F/T perception interface to include only resistance force $f_r$ and torque $\tau_r$.

For each skill, we first ask the LLM to generate an initial success condition function, in which it determines which signal the condition should be based on (e.g. `resistance_torque` is used to form the `is_tightened` condition). We then provide a plot of the selected signal and prompt the LLM to update success condition func-
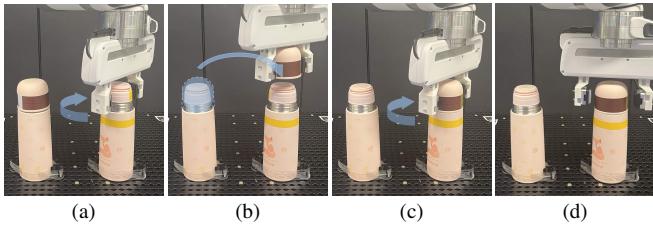
Fig. 5. Example Evaluation Task for Cap Tightening. (a) Tightening the inner cap. (b) Moving the outer cap to the target bottle. (c) Tightening the outer cap. (d) Releasing.



```
move_object('clip8','position')
insert('clip8','insertion_direction')
move_object('clip5','position')
insert('clip5','insertion_direction')
```
(a) Control Group A and B

```
move_object('bottle1', 'position')
grasp( )
move_object('bottle1', 'position')
tighten( )
release( )
move_object('bottle1', 'position')
```
(b) Control Group A

```
grasp( )
move_object('bottle1', 'position')
release( )
grasp( )
tighten( )
release( )
grasp( )
tighten( )
release )
......
# loops
```
(c) Our Pipeline

Fig. 6. Skill Sequences Reasoned from Demonstrations. (a) Sequence reasoned for cable mounting by control groups A and B. The demo skill sequence in Fig. 3(c) shows the result by our pipeline. (b) and (c) Sequence reasoned for cap tightening. The resulting sequence of group B is similar to group A but with more move_object steps.

tions accordingly. An example of the resulting function `is_inserted` for the task of mounting cable to clips is shown in Fig. 3(c). The LLM defines the success condition for insertion as the resistance force falling below a certain threshold, indicating that the cable has been securely inserted. We present later in Section IV-B that numerical thresholds are particularly refined with the F/T signals.

### C. Planning in New Scenarios

From the above bootstrapping reasoning, the LLM analyzer has extracted the skill sequences corresponding to human demonstrations, and has extended the original skill library with appropriate success conditions. These outputs are then combined as an demonstration task plan, which will be used as an example for task planning on a new task. As shown in Fig. 2, in the planning request to an LLM planner, we include the demonstration task plan as well as an image and description of the new task scene. To make the plan more dynamic and flexible, we also use the LLM planner to monitor the execution process. After execution of each skill, we feed its return $R$ back to the LLM planner, which then decides whether the plan should be continued or adjusted.

## IV. EXPERIMENTS

In this section, we present real-world experiments to evaluate the effectiveness of our demonstration reasoning pipeline and planning results for new tasks. This evaluation is conducted through ablation study: by disabling or replacing a certain parts in our framework, we design the following control groups.

A. Transition Frames Without Object Status: Key frames in our demonstration reasoning pipeline are replaced by frames at key timestamps but without status annotation.

B. Uniform Sampled Frames Without Object Status: Similar to (a), but frames are sampled uniformly from video, again without status annotation.

C. Conditions Without F/T Signals: Force/torque signals are excluded from the demonstration reasoning pipeline, so the success conditions remain as initially generated by the LLM without any updates.

D. Without Demonstrations: No demonstration data is provided and the LLM generates the plan solely based on its prior knowledge.

### A. Experimental Setup

**Setup** We use sigma.7 haptic devices [32] to control two Franka Emika robots for collecting demonstrations. The visual observation is captured by a third-person Intel RealSesne D435 camera. As shown in Fig 1 (a), tactile videos are collected by the GelSight ViTac sensor [33] on robot fingers. F/T signals are collected by Bota Systems SensOne 6D force-torque sensor at the wrist, which also provides the human operator with haptic feedback during demonstration. In the grounding stage, we use GPT-4 Omni as both analyzer and planner for its supreme compability of processing images and videos.

**Evaluation Tasks** We evaluate our framework on two sequential manipulation tasks, each presenting distinct challenges for task planning:

• Cable mounting task, where a cable needs to be moved and inserted sequentially onto several clips. This task is a common process in industries like car manufacturing. Inspired by how human handles this task, we use a bi-manual robotic setup. One robot (called robot leader) holds the cable and moves it to each clip. The second robot (called robot follower) joins the grasping when the cable reaches a clip, and both robots will perform the insertion together. During demonstration, the operator teaches the robots to mount the cable onto two clips of different types (See Fig. 1(a) and task description in Fig. 3(b)). For evaluation of planning compability, we randomize the number and position of clips as new task configurations.

• Cap tightening task, where one robot should attach cap(s) onto bottles. During demonstration, the operator teaches the robot to pick a cap from the desk and tighten it to a target bottle. For evaluation of new task planning, the target bottle has an inner cap and an outer cap with randomized positions (See Fig. 5). The robot is supposed to tighten both caps to the target bottle.

### B. Evaluation on Demonstration Reasoning

Firstly, we evaluate the bootstrap reasoning pipeline in terms of its ability to extract correct skill sequences from demonstrations. We draw a comparison with control group A and B which rely solely on visual information to reason skill sequences. The extracted skill sequence for cable mounting and cap tightening are presented in Fig. 6. While skills involving obvious movements (such as move_object and insert) are successfully recognized by all groups, the LLM struggles to reliably identify skills involving intensive physical interactions (such as grasp and stretch) when only visual data is provided (Fig. 6 (a)). Despite tighten occurring multiple times in the cap tightening demonstration, the control groups manage to identify it only once (Fig. 6

move_object (clip5) ⟶ grasp ⟶ stretch ⟶ insert (clip5) ⟶ open_hand ⟶

⟶ ··· ⟶ stretch ⟶ insert (clip6) ⟶ ··· ⟶ grasp ⟶ insert (clip8) ⟶ ···
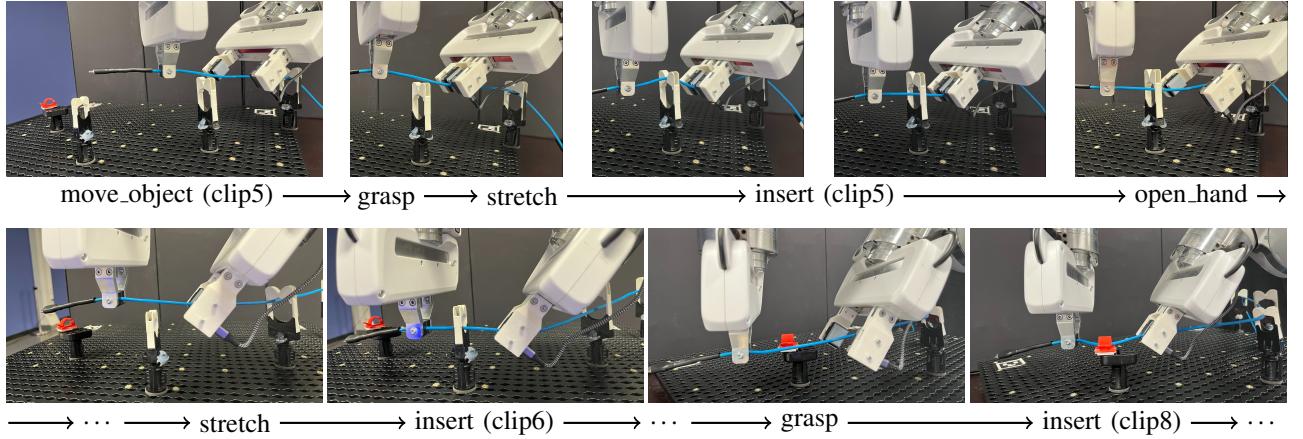
Fig. 7. Execution of Cable Mounting Plan Generated by the LLM Planner.

(b)). Furthermore, `tighten` is often misidentified as other skills, leading to an unreasonable sequence. This comparison highlights that incorporating object status data from tactile sensors in our pipeline significantly improves the accuracy and completeness of the skill sequences extraction.

Next, we evaluate the transition conditions reasoned by our pipeline using F/T signals. Table II shows `stretched` and `inserted` conditions for cable mounting as well as `tightened` condition for cap tightening generated by the LLM, each containing force-relevant thresholds. We observe that after F/T signals were introduced, the LLM retained its initial condition formulations but used the signals to update threshold estimations. We then executed these skills on real-world robots to compare the success rate of each skill before (control group (c)) and after the update. Since directly assessing the success of the `stretch` skill is challenging, we evaluated it in combination with the `insert` process. The `insert` skill was tested on both clip types, with 20 trials conducted at random positions for each, while the `tighten` skill was evaluated with the cap positioned at initial orientations ranging from 30 to 180 degrees. The success rates before and after the updates (shown in Table II) indicate that the conditions for cable insertion into the U-type clip and cap tightening improved significantly with the integration of the demonstrated F/T signals. However, for the C-type clip which involves more dramatic and abrupt force changes, despite a rise in success rate after updates, the simple threshold reasoned by the LLM is still insufficient for robustly detecting success. More complex indicators, such as the change rate of resistance force, are likely required for better accuracy.

### C. Evaluation on Task Planning

Finally, we evaluate the plans generated by the LLM for new task configurations. To compare the performance of our framework against all the control groups (a)-(d), we first assess the reasonableness of the generated skill sequences, assuming all skills in the plan are executed successfully. Next, we test the plans on real-world robots, evaluating whether each skill is executed correctly without errors in skill return and whether the entire task is completed successfully. For instance, if the `insert` success condition fails to detect when the cable has been inserted, the robots may continue

TABLE II EVALUATION ON SKILL CONDITION

| | Before Update | | After Update | |
|---|---|---|---|---|
| | Condition | Success | Threshold | Success |
| Stretch | $f_r > 10N$ | 0.00 | $f_r > 9.5N$ | 0.90 |
| Insert U | $f_r < 5N$ | | $f_r < 2.5N$ | |
| Stretch | $f_r > 10N$ | 0.00 | $f_r > 9.5N$ | 0.50 |
| Insert C | $f_r < 10N$ | | $f_r < 4.5N$ | |
| Tighten | $\tau_r > 0.02N \cdot m$ | 0.00 | $\tau_r > 2N \cdot m$ | 1.00 |

TABLE III EVALUATION ON TASK PLANNING

| | Reasonableness | | Executability | | Success | | Overall | |
|---|---|---|---|---|---|---|---|---|
| | Cable | Cap | Cable | Cap | Cable | Cap | Cable | Cap |
| **Ours** | **1.00** | **1.00** | **0.91** | **1.00** | **0.91** | **1.00** | **0.94** | **1.00** |
| Group A | 0.00 | 0.00 | 0.95 | 0.00 | 0.15 | 0.00 | 0.36 | 0.00 |
| Group B | 0.00 | 0.00 | 0.95 | 0.00 | 0.15 | 0.00 | 0.36 | 0.00 |
| Group C | 1.00 | 100% | 1.00 | 1.00 | 0.00 | 0.00 | 0.66 | 0.66 |
| Group D | 0.00 | 1.00 | 0.90 | 1.00 | 0.00 | 0.00 | 0.30 | 0.66 |

pushing blindly, eventually throwing an error due to joint torque limits. In such cases, while the task itself is technically completed (the cable is inserted into the clip), the execution is considered a failure due to the error.

The overall performance is calculated as the average of these three criteria. The evaluation results, presented in Table III, show that our framework outperforms all control groups, particularly in terms of task success rate. One example of LLM-generated cable mounting plan as well as the execution process is demonstrated in Fig. 7, which successfully accomplish the assembly task. For more detailed planning results, please refer to our accompanying video and project website.

### V. CONCLUSION

We introduce an in-context learning framework that enables task planning for sequential, contact-rich manipulation tasks using multi-modal demonstration data. This framework leverages tactile and force/torque information to segment and convert the entire demonstration into a task plan, which is then used as a reference to generate more reliable plans for new tasks. For future work, we plan to incorporate language instructions to further enhance LLMs' understanding of demonstrations. Additionally, fine-tuning a VLM to directly interpret tactile and force/torque data presents another promising approach to leverage multi-modal demonstrations.

## References

[1] OpenAI, J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman *et al.*, "Gpt-4 technical report," 2024. [Online]. Available: https://arxiv.org/abs/2303.08774

[2] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan *et al.*, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 1877–1901.

[3] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal *et al.*, "Learning transferable visual models from natural language supervision," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 8748–8763.

[4] X. Huang, W. Liu, X. Chen, X. Wang, H. Wang, D. Lian, Y. Wang, R. Tang, and E. Chen, "Understanding the planning of llm agents: A survey," *arXiv preprint arXiv:2402.02716*, 2024.

[5] D. Driess, F. Xia, M. S. M. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong *et al.*, "Palm-e: an embodied multimodal language model," in *Proceedings of the 40th International Conference on Machine Learning*, ser. ICML'23. JMLR.org, 2023.

[6] C. H. Song, J. Wu, C. Washington, B. M. Sadler, W.-L. Chao, and Y. Su, "Llm-planner: Few-shot grounded planning for embodied agents with large language models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023.

[7] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng, "Code as policies: Language model programs for embodied control," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 9493–9500.

[8] Y. Xie, C. Yu, T. Zhu, J. Bai, Z. Gong, and H. Soh, "Translating natural language to planning goals with large-language models," *arXiv preprint arXiv:2302.05128*, 2023.

[9] Z. Zhou, J. Song, K. Yao, Z. Shu, and L. Ma, "Isr-llm: Iterative self-refined large language model for long-horizon sequential task planning," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 2081–2088.

[10] N. Wake, A. Kanehira, K. Sasabuchi, J. Takamatsu, and K. Ikeuchi, "Gpt-4v (ision) for robotics: Multimodal task planning from human demonstration," *arXiv preprint arXiv:2311.12015*, 2023.

[11] C. Wang, L. Fan, J. Sun, R. Zhang, L. Fei-Fei, D. Xu, Y. Zhu, and A. Anandkumar, "MimicPlay: Long-Horizon Imitation Learning by Watching Human Play," in *Proceedings of The 7th Conference on Robot Learning*. PMLR, Dec. 2023, pp. 201–221.

[12] H. Zhou, M. Ding, W. Peng, M. Tomizuka, L. Shao, and C. Gan, "Generalizable long-horizon manipulations with large language models," *arXiv preprint arXiv:2310.02264*, 2023.

[13] T. Silver, S. Dan, K. Srinivas, J. B. Tenenbaum, L. Kaelbling, and M. Katz, "Generalized planning in pddl domains with pretrained large language models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 18, 2024, pp. 20 256–20 264.

[14] H. Zhou, Y. Lin, L. Yan, J. Zhu, and H. Min, "LLM-BT: Performing Robotic Adaptive Tasks based on Large Language Models and Behavior Trees," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, May 2024, pp. 16 655–16 661.

[15] J. Ao, Y. Wu, F. Wu, and S. Haddadin, "Behavior tree generation using large language models for sequential manipulation planning with human instructions and feedback," in *ICRA 2024 Workshop Exploring Role Allocation in Human-Robot Co-Manipulation*, 2024.

[16] M. S. Sakib and Y. Sun, "From cooking recipes to robot task trees–improving planning correctness and task efficiency by leveraging llms with a knowledge network," *arXiv preprint arXiv:2309.09181*, 2023.

[17] G. Konidaris, S. Kuindersma, R. Grupen, and A. Barto, "Robot learning from demonstration by constructing skill trees," *The International Journal of Robotics Research*, vol. 31, no. 3, pp. 360–375, 2012.

[18] S. Ekvall and D. Kragic, "Robot learning from demonstration: a task-level planning approach," *International Journal of Advanced Robotic Systems*, vol. 5, no. 3, p. 33, 2008.

[19] S. Manschitz, J. Kober, M. Gienger, and J. Peters, "Learning to sequence movement primitives from demonstrations," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 4414–4421.

[20] D. Kulić, C. Ott, D. Lee, J. Ishikawa, and Y. Nakamura, "Incremental learning of full body motion primitives and their sequencing through human motion observation," *The International Journal of Robotics Research*, vol. 31, no. 3, pp. 330–345, 2012.

[21] S. Niekum, S. Osentoski, G. Konidaris, and A. G. Barto, "Learning and generalization of complex tasks from unstructured demonstrations," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 5239–5246.

[22] Y. Zhu, P. Stone, and Y. Zhu, "Bottom-up skill discovery from unsegmented demonstrations for long-horizon robot manipulation," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4126–4133, 2022.

[23] A. Mandlekar, D. Xu, R. Martín-Martín, S. Savarese, and L. Fei-Fei, "Learning to generalize across long-horizon tasks from human demonstrations," *arXiv preprint arXiv:2003.06085*, 2020.

[24] A. Gupta, V. Kumar, C. Lynch, S. Levine, and K. Hausman, "Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning," *arXiv preprint arXiv:1910.11956*, 2019.

[25] L. Shao, T. Migimatsu, Q. Zhang, K. Yang, and J. Bohg, "Concept2robot: Learning manipulation concepts from instructions and human demonstrations," *The International Journal of Robotics Research*, vol. 40, no. 12-14, pp. 1419–1434, 2021.

[26] G. Chen, T. Cui, T. Zhou, Z. Peng, M. Hu, M. Wang, Y. Yang, and Y. Yue, "Human demonstrations are generalizable knowledge for robots," *arXiv preprint arXiv:2312.02419*, 2023.

[27] K. Burns, A. Jain, K. Go, F. Xia, M. Stark, S. Schaal, and K. Hausman, "Genchip: Generating robot policy code for high-precision and contact-rich manipulation tasks," 2024. [Online]. Available: https://arxiv.org/abs/2404.06645

[28] L. Fu, G. Datta, H. Huang, W. C.-H. Panitch, J. Drake, J. Ortiz, M. Mukadam, M. Lambeta, R. Calandra, and K. Goldberg, "A touch, vision, and language dataset for multimodal alignment," *arXiv preprint arXiv:2402.13232*, 2024.

[29] J. Mejia, V. Dean, T. L. Hellebrekers, and A. Gupta, "Hearing touch: Audio-visual pretraining for contact-rich manipulation," *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6912–6919, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:269761987

[30] L. Johannsmeier, M. Gerchow, and S. Haddadin, "A framework for robot manipulation: Skill formalism, meta learning and adaptive control," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 5844–5850.

[31] G. Bertasius, H. Wang, and L. Torresani, "Is space-time attention all you need for video understanding?" in *Proceedings of the International Conference on Machine Learning (ICML)*, July 2021.

[32] A. Tobergte, P. Helmer, U. Hagn, P. Rouiller, S. Thielmann, S. Grange, A. Albu-Schäffer, F. Conti, and G. Hirzinger, "The sigma.7 haptic interface for mirosurge: A new bi-manual surgical console," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 3023–3030.

[33] W. Yuan, S. Dong, and E. H. Adelson, "Gelsight: High-resolution robot tactile sensors for estimating geometry and force," *Sensors*, vol. 17, no. 12, p. 2762, 2017.