Integrating LMM Planners and 3D Skill Policies for Generalizable Manipulation

Yuelei Li*, Ge Yan* , Annabella Macaluso, Mazeyu Ji, Xueyan Zou, Xiaolong Wang
UC San Diego

Abstract

*The recent advancements in visual reasoning capabilities of large multimodal models (LMMs) and the semantic enrichment of 3D feature fields have expanded the horizons of robotic capabilities. These developments hold significant potential for bridging the gap between high-level reasoning from LMMs and low-level control policies utilizing 3D feature fields. In this work, we introduce LMM-3DP, a framework that can integrate **LMM** planners and **3D** skill **P**olicies. Our approach consists of three key perspectives: high-level planning, low-level control, and effective integration. For high-level planning, LMM-3DP supports dynamic scene understanding for environment disturbances, a critic agent with self-feedback, history policy memorization, and reattempts after failures. For low-level control, LMM-3DP utilizes a semantic-aware 3D feature field for accurate manipulation. In aligning high-level planning and low-level control for robot actions, language embeddings representing the high-level policy are jointly attended with the 3D feature field in the 3D transformer for seamless integration. We extensively evaluate our approach across multiple skills and long-horizon tasks in a real-world kitchen environment. Our results show a significant **1.45x** success rate increase in low-level control and an approximate **1.5x** improvement in high-level planning accuracy compared to LLM-based baselines. Demo videos and an overview of LMM-3DP are available at https://lmm-3dp-release.github.io.*

1. Introduction

Building generally capable robots that can perform a wide range of long-horizon tasks in the real world is a long-standing problem. Recent advancements in robotics have been driven by large language models (LLMs) that have shown remarkable capabilities in understanding the real world and common sense reasoning. Some studies leverage LLMs to decompose an abstract task into a sequence of high-level language instructions for planning [3, 12, 21, 22, 34, 38, 39, 41, 47, 51]. Despite the significant advancements they have facilitated in various real-world tasks, the current integration of LLMs into robotics presents several major drawbacks. First, LLMs can only process natural language with no visual understanding, making it difficult to comprehend and adapt to dynamic real-world scenarios requiring rich visual information. Additionally, LLM-based planners usually depended on human language feedback to perform long-horizon planning consistently [21, 39, 47], which significantly constrains autonomy. However, large multimodal models (LMMs), with multi-sensory inputs, have emerged as a powerful tool to equip robots with strong visual understanding and generalization capabilities across various environments. This allows the robot to adjust language plans according to the environment change. In this paper, we focus on leveraging LMMs to generate language plans based on environment feedback and keep self-improvement in a closed-loop manner.

Existing LLM-based planners typically rely on a predefined set of primitive skills for low-level control [2, 8, 22, 25, 35, 51], which is the main bottleneck of large-scale applications to open-world environments. Therefore, the ability to acquire robust low-level skills capable of adapting to the novel environment in a data-efficient manner presents a significant challenge for most LLM-based frameworks. Some recent studies use LLMs to directly output low-level control [24, 48]. However, they are only effective in relatively simple manipulation tasks that do not involve rapid high-dimensional control. Due to insufficient 3D understanding, LLMs often fail in complex environments that require comprehending the 3D structure of the scene efficiently. In addition, recent works leverage vision-language models (VLMs) for visual grounding by predicting bounding boxes or keypoints of target objects [3, 23]. Despite promising results, they rely on off-the-shelf VLMs which may not be fully optimized for specific, complex tasks in dynamic environments.

To address these challenges, we introduce LMM-3DP, an LMM-empowered framework that integrates LMM for self-improved high-level planning and an efficient 3D pol-

*Equal Contribution.

icy for low-level control (see Fig. 1). Our framework is designed to satisfy two key requirements: 1) it ensures our LMM agent achieves high autonomy during continuous deployment by decomposing a long-horizon task into high-level plans, calling low-level policy for execution, receiving the environment feedback, and updating language plans accordingly. 2) it allows the low-level policy to learn various skills efficiently with only a few human demonstrations and improve continually.

For high-level planning, we introduce three key modules to build an autonomous agent capable of planning a sequence of language instructions: 1) **Interactive planning with visual feedback.** Incorporating visual feedback within the loop is crucial for enabling an agent to rapidly adapt to dynamic scene changes. In this work, we adopt GPT-4V [1, 30] as an LMM planner to receive environmental feedback and monitor the ongoing events during execution. 2) **Self-improvement with memory and critic.** We introduce a critic agent to analyze the plan generated by the LMM planner. It outputs the critique of the planner's decisions and informs whether the plan needs to be updated. In addition, LMM-3DP stores history critique into a memory module and summarizes learned experience for the planner. This approach significantly improves planning accuracy and consistency, especially in challenging long-horizon tasks. 3) **Life-long learning with a skill library.** Open-ended real-world scenarios usually bring an infinite set of tasks with different skill compositions. The ability to acquire new skills in a data-efficient manner is critical for robots to be generally capable of performing various real-world tasks. Thus, LMM-3DP builds a skill library to retrieve different skills required by the LMM planner. When requiring new skills, we adopt an efficient imitation learning policy to grasp such skills with limited human demonstrations.

More specifically, for precise low-level control, we develop a language-conditioned multi-task 3D policy to learn generalizable skills. To tackle challenging tasks with various object categories and complex environments (*e.g.*, partial occlusion, various geometry shapes, and intricate spatial relationships), it is essential to have a comprehensive semantic and geometry understanding of the scene. Therefore, we first use a vision foundation model to extract 2D semantic features from RGB images, which are then back-projected into 3D space. We then fuse the semantic feature with the geometric point cloud features from a point-based network [32]. Based on this unified 3D and semantic representation, we train an end-to-end imitation learning policy with a 3D transformer architecture. Our approach is capable of learning various skills efficiently, only requiring a limited number of demonstrations. This facilitates the construction of our ever-growing skill library with robust low-level skills that are reusable and generalizable to novel tasks and environments.

For evaluation, we designed a series of experiments to demonstrate our framework's reliable high-level planning, generalizable low-level control, and exceptional performance in long-horizon tasks. For challenging long horizon tasks, LMM-3DP have an average accuracy of **56.5%**, while our baseline only has an overall average accuracy of **7%** and first step average accuracy of **50%** (in a multi-step execution). Additionally, we ablate the design of the critic agent and visual feedback in the loop to delve deeper into the contribution of each component in our framework.

2. Related Work

LLMs as Task Planners. Recent advancements in large language models (LLMs) have greatly influenced robotics in various applications. Notable methods typically include using LLMs to generate high-level plans [3, 12, 22, 45, 51]. For example, SayCan [3] underscores the extraordinary commonsense reasoning ability of LLMs by generating feasible language plans and adopting an affordance function to weigh the skill's likelihood for execution. Some approaches also leverage LLMs to produce programming code or symbolic API as plan [4, 20, 25, 26, 38, 40, 50]. However, these methods only take natural language instructions as input and lack the ability to perceive the world with multi-modal sensory observations. Therefore, they cannot adjust the language plans based on environmental feedback, which strongly limits their performance in dynamic real-world environments. Due to the emergence of LMMs, some studies [17, 19, 42] leverage GPT-4V [1] for planning with visual input. However, they only use GPT-4V as a fixed planner without critic and self-improvement while we allow the agent to continue exploring and improving in open-world environments.

Low-Level Robot Primitives. Despite the significant progress in high-level planning, previous LLM-based language planners [2, 8, 22, 25, 51] hold a strong assumption that there exist reliable low-level skills for high-level planners to retrieve, which are usually manually pre-defined set of skills. Some studies [10, 25, 43, 48] use LLMs to output direct low-level control in text, which is impractical to apply to complex real-world tasks requiring high-dimensional control. Some methods [16, 18, 23, 27, 36] also leverage vision language models (VLMs) to infer language-grounded affordances and perform motion planning. However, they still lack accurate 3D understanding for challenging environments with diverse geometry shapes and intricate 3D structures. However, LMM-3DP addresses this challenge by integrating the high-level planner with a language-conditioned 3D policy that can efficiently learn new skills with a comprehensive 3D understanding of the scene structure.

3D Representations for Low-Level Skills. To learn a visual imitation learning policy for various skills, most

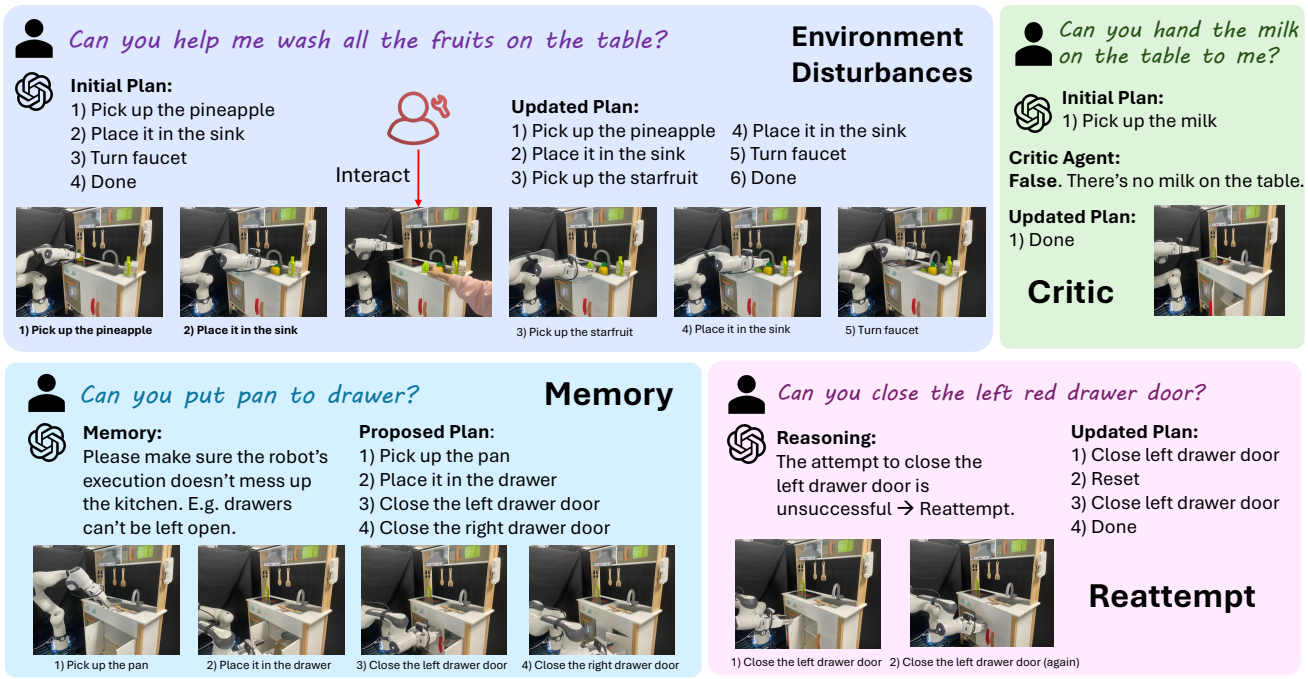

Figure 1. LMM-3DP effectively handles environment disturbances, retries if previous attempts fail, and performs accurate reasoning even when human instructions are not aligned with observations.

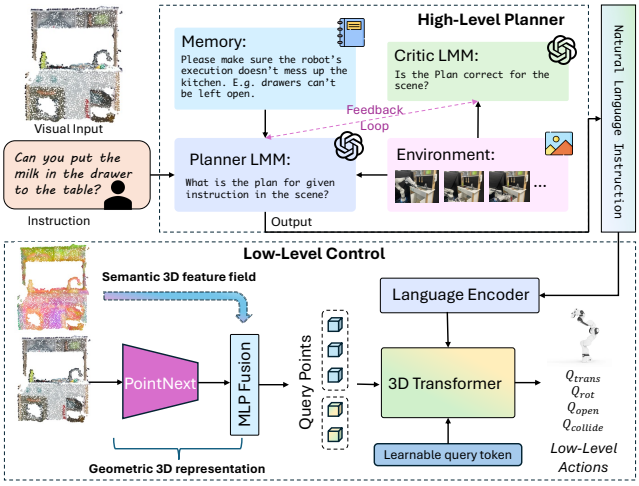

Figure 2. ***Full Framework Pipeline***.

previous works [5, 6, 11, 13, 15, 31] have been leveraging 2D image-based representation for policy training, while the advantage of 3D representation over 2D images has been increasingly recognized by recent studies [9, 37, 46, 49, 52, 53]. GNFactor [49] and DNAct [46] learn a 3D representation by distilling 2D features from vision foundation models. However, they still require laborious multi-view image collection to train a NeRF [28] model, which poses a challenge to large-scale deployment. In this work, we learn a unified 3D and semantic representation by adopting a two-branch architecture with PointNext [32] and DINO [7] to provide geometry and semantic understanding respectively. Our policy is capable of learning multiple skills with only a few demonstrations.

3. Method

In this work, we aim to develop a robust planning framework to generate high-level language plans, along with a generalizable skill-level control policy to follow language plans and execute actions. In this section, we first discuss the design of our self-improved high-level planner, then introduce our language-conditioned skill-level policy (see Fig. 2 for the whole pipeline).

3.1. LMM for High-Level Planning.

Planning with Visual Feedback. In the real world, the optimal plan to execute a task may not be the one initially devised. For instance, you might plan to put vegetables in your favorite blue bowl for dinner, but upon discovering that the blue bowl is unavailable, you use a red bowl instead. Similarly, in robotic planning, the robot must be able to update its plan according to the current situation, which necessitates visual feedback during task execution. We integrate GPT-4V as a planner within the robot's execution loop, enabling it to update the plan after each skill is executed. This design enhances the robot's ability to adapt

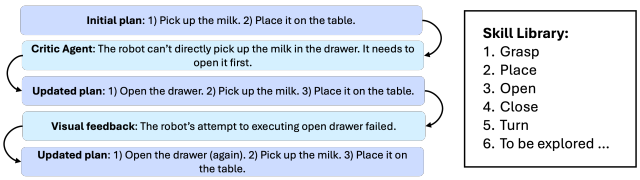

Figure 3. Example of how our planner updates the plan during the robot's execution.

to dynamic scenes (e.g., when there are environmental disturbances) and reattempt a previous skill if the low-level control fails to execute (see Fig. 3).

Critic Agent. To ensure that the plan generated by the planner is correct and reliable, we introduce an additional critic agent to proactively identify flaws in the generated plan with continuous self-improvement. The critic agent, which only takes visual observation and proposed plan as input (without human instruction), checks whether the next step is feasible in the current situation. If the critic finds that executing the next step will result in an undesirable outcome, its reasoning is input back to the planner, which then proposes a new plan. For instance, the planner's output can be easily skewed by human instructions. This issue persists even with popular prompting techniques [44]. If the human instruction is to close all the drawers, but some drawers are already closed in the scene, the planner might still generate a plan that involves closing all the drawers. However, the critic can accurately determine that the robot should not close a drawer that is already closed, thereby correcting the planner's mistake.

Lifelong Learning. We aim for the planner to improve over time and avoid repeating mistakes by learning from past experience, similar to human learning. However, fine-tuning the planner is computationally expensive. Instead, we employ human critiques of the GPT-4V's output plan and reasoning and then summarize these critiques for in-context learning. These summaries are stored as memory for the planner to reference in the future. Additionally, the planner can propose new skills to the skill library when necessary, then the low-level policy will be updated accordingly with these new skills. For example, in a cooking task, without the click skill, the robot cannot turn on the stove. The planner would identify the click skill as essential for future learning. This approach enables our framework to handle more complex tasks as the skill library expands.

3.2. Skill learning with 3D Semantic Representation

Given the language instructions generated by the planner, we train a language-conditioned 3D policy to learn the required low-level skills from human demonstration data. Instead of predicting every continuous action, we extract keyframe actions and convert the skill learning into a keyframe-based action prediction problem. This approach simplifies continuous control and is more sample-efficient for learning a generalizable policy capable of handling novel objects and environments.

Ours	Grasp	Place	Turn	Open	Close
w/o distractor	90%	65%	80%	40%	100%
w/ distractor	56%	50%	70%	40%	80%

Table 1. Skill Accuracy.

Ours		OWL-v2		Voxposer	
Grasp	Place	Grasp	Place	Grasp	Place
90%	65%	60%	45%	60%	50%

Table 2. Skill Comparison.

Vision and language encoder. To tackle complex real-world environments with various objects and scene structures, we learn a unified 3D and semantic representation by adopting a two-branch architecture: i) Pre-trained with internet-scale data, the vision foundation model has achieved great success in understanding complex scenes with strong zero-shot generalization ability. To leverage these powerful vision foundation models in robotics, we apply a foundation model (e.g., DINO [7]) to extract 2D image features with rich semantics. We then obtain a 3D point feature by back-projecting the 2D feature maps to 3D space. ii) Despite rich semantics from the vision foundation model, it still lacks an accurate geometric understanding. Therefore, we adopt a separate branch of a point-based model (e.g., PointNext [32]) to learn a geometry point feature for better capturing local 3D structures. Subsequently, both semantic and geometry point features are fused by an MLP layer. To incorporate language understanding into our policy, we use a pre-trained language encoder from CLIP [33] to obtain a language embedding.

Keyframe action prediction. Given the fused 3D point feature, language embedding, and robot proprioception, we adopt a 3D transformer architecture to predict the 6-DOF pose of the next best keyframe. Instead of predicting continuous action, we simplify the model prediction into translation $a_{\text{trans}} \in \mathbf{R}^3$, rotation $a_{\text{rot}} \in 0, 1^{(360/5)3}$, gripper openness $a_{\text{open}} \in [0, 1]$, and collision avoidance $a_{\text{collision}} \in [0, 1]$. Specifically, we approximate the continuous 3D field via sampling a fixed set of query points in the gripper's workspace. We do this because, unlike voxel-based methods that discretize the output space and are memory inefficient, the sampling-based approach provides a continuous output space and saves memory during training. We also define a learnable token to attend to the local structures more efficiently. Both the query points and the learnable token

are passed through multiple cross-attention layers with the visual and language features, to obtain the token feature $f_{\mathbf{t}}$ and query point feature $f_{\mathbf{q}}$. We then assign a score for each query point by computing the inner product of $f_{\mathbf{t}}$ and $f_{\mathbf{q}}$. The next best waypoint P_i is chosen by applying an argmax operation to the score. Inspired by [14], we subsequently resample a reduced set of query points around P_i and refine the selection of waypoints among these query points based on previous predictions.

$$\mathcal{L}_{\text{bc}} = \lambda_{\text{trans}} \cdot \text{CE}_{\alpha}(\mathcal{V}_{\text{trans}}, Y_{\text{trans}}) + \lambda_{\text{rot}} \cdot \text{CE}(\mathcal{V}_{\text{rot}}, Y_{\text{rot}}) + \lambda_{\text{open}} \cdot \text{CE}(\mathcal{V}_{\text{open}}, Y_{\text{open}}) + \lambda_{\text{collide}} \cdot \text{CE}(\mathcal{V}_{\text{collide}}, Y_{\text{collide}}),$$

where $\mathcal{V}_i = \text{softmax}(\mathcal{Q}_i)$ for $\mathcal{Q}_i \in \{\mathcal{Q}_{\text{trans}}, \mathcal{Q}_{\text{rot}}, \mathcal{Q}_{\text{open}}, \mathcal{Q}_{\text{collide}}\}$. $Y_i \in \{Y_{\text{trans}}, Y_{\text{rot}}, Y_{\text{open}}, Y_{\text{collide}}\}$ is the ground-truth one-hot encoding. $\text{CE}(p, y) = -\sum_j y_j \log p_j$ is the cross-entropy loss, and CE_{α} denotes cross-entropy with label smoothing parameter α, applied only to the translation term to prevent overfitting and mitigate label noise in real-world demonstrations.

4. Experiments

Experiment Setup & Implementation Details. We set up a real-world kitchen environment for our experiments, which is more complicated and has more visual features compared to a simple tabletop setting. Our robot is a 7-DoF Franka Emika Panda robot with a 1-DoF deformable gripper. For visual input, we use two Intel RealSense D435 cameras: one provides a front view, and the other is mounted on the gripper. To collect data for our imitation learning-based low-level policy, we use an HTC VIVE controller and base station to track the 6-DOF poses of human hand movement. Then we use SteamVR to map the controller movement to the end effector of the Franka robot. In low-level policy training, we use 100 human demonstrations for one kitchen setting and 200 demonstrations for two kitchen settings (10 demonstrations for each task). We employ the Adam optimizer with a learning rate of 3×10^{-4}. The training is conducted on one NVIDIA GeForce RTX 3090 with a batch size of 16. We apply color dropout and translation augmentation techniques to improve the model's performance.

4.1. Main Results

To perform well on long-horizon tasks, we need to ensure the following: 1) a generalizable low-level policy capable of performing various skills, 2) an adaptable low-level policy that can compose these skills together, and 3) a situation-aware high-level planner with strong reasoning abilities. We systematically evaluate our framework on each of the three criteria individually, then integrate all these components and test our framework's performance on long-horizon tasks

Location / Object	pineapple (s,d)	starfruit (s)	milk	duck (d)	pan
sink	90%	90%	80%	80%	50%
drawer	90%	80 %	70%	90%	40%

Table 3. Pick/Place Accuracy. s means the object has been trained to be placed in the sink. d means the object has been trained to be placed in the drawer.

Task	SayCan	Voxposer	Ours
Open both drawer doors.	20%	90%	90%
Place the gray pan in the drawer, which is closed initially.	0%	50 %	80%
Put the fruits (pineapple, starfruit) into the bowl.	40%	100%	90%
Stack all the bowls on the kitchen table.	90%	100 %	100 %
Place the pineapple in the sink.	100 %	100 %	100 %

Table 4. High-level Planning Comparison.

(See Fig. 1 for qualitative results). If not otherwise stated, each reported accuracy rate is obtained with 10 trials.

Low Level Skills. We train and evaluate our pipeline's performance across five distinct skills: grasping, placing, turning, opening, and closing. Each skill is tested with various objects and task scenarios (Pick is tested 5 times for each of 5 objects, place 5 times for each of 4 locations, and other skills 10 times total). To show the generalization ability of our low-level policy, we report the individual skill accuracy with and without distractors, where the distractors include 1 - 2 extra toys placed in the kitchen to make it more cluttered (see Table 1).

We use two baselines. First, we use OWL-v2 [29], an open-vocabulary object detector as an affordance model to output bounding boxes for different objects. This baseline is similar to the approach used in recent works, like [27]. We also include Voxposer [23] as a baseline, which is a recent SoTA method on LLM for long-horizon tasks and robot manipulation. Our results demonstrate that our method significantly outperforms the baseline (see Table 2). The detector performs poorly with asymmetrical objects, whereas our method learns to grasp these items from human demonstrations efficiently. Also, the detector is highly view-dependent for locating the center of the object of interest, whereas our method performs well as long as the front camera captures the entire scene.

Skill Composition. To successfully compose different skills in sequence, it is essential to demonstrate that these skills are disentangled so that the execution of one skill does not impact the subsequent skills. Our focus is on pick and place operations, as they are highly interrelated. For instance, after training the model on the tasks of putting object A to location B, and putting object C to location D, the model should also be capable of putting C to B. We randomly combine two locations and five objects. Our results in Table 3 show that the pick and place skills can be com-

	Grasp	Place	Turn	Open	Close
1st kitchen (2 kitchen checkpoint)	72%	75%	70%	50%	90%
2nd kitchen (2 kitchen checkpoint)	72%	60%	70%	30%	80%
Overall (2 kitchen checkpoint)	72%	67.5%	70%	40%	85%
1st kitchen (1 kitchen checkpoint)	90%	65%	80%	40%	100%

Table 5. Two kitchen setting experiments.

Human demonstrations per task	Grasp	Place	Turn	Open	Close
10	72%	67.5%	70%	40%	85%
5	56 %	65%	80%	50%	75%

Table 6. 5 vs 10 human demonstrations on two kitchens.

posed together arbitrarily without extra training. For example, though the milk hasn't been directly trained on being placed in the sink and drawer, the place skill still achieves a high accuracy rate on the milk object.

High level planning: GPT4V vs LLM. We compare our high-level planning method with SayCan [2], a widely used method that leverages LLM reasoning with affordance scores to generate robotic plans, and Voxposer [23], a recent SoTA method on LLM for long-horizon tasks and robot manipulation. We found that SayCan's method of selecting the next action based on maximum log-likelihood from an action list limits the language model's reasoning ability. This approach makes the model verb-insensitive, fails to understand the semantic meaning of nouns, and is prone to repeating previous actions. Our proposed framework, similar to Voxposer, directly prompts the planning agent in a conversational format rather than a language completion approach, which produces overall better results. Additionally, both Saycan and Voxposer cannot generate plans that consider the state of objects due to the absence of visual input, whereas our method benefits from visual feedback to generate the most reasonable plan. Our results show that our model performs comparably to Voxposer on common kitchen tasks that do not require visual information for reasoning, but demonstrates superior performance when visual information is necessary (see Table 4). In the "place gray pan in drawer" task, our method reliably identifies the closed drawer, opens it, and then places the pan inside. In contrast, SayCan and Voxposer frequently neglect to open the drawer.

Long-Horizon [Multi-Steps] Tasks. Here we define long-horizon tasks as those that require $\geq$ 3 action steps to complete (see the Actions column in Table 7). To evaluate performance, we design three such tasks that combine skill-level control, skill composition, and high-level planning, and test their accuracy rates. Our first baseline includes a high-level planning module from SayCan [2] and a 2D object detection control module using OWLv2 [29], similar to our previous experiments. We also use Voxposer [23] as a second baseline. The result (see Table 7 and Fig. 1) shows that our method achieves a much higher accuracy rate compared to the baseline methods. We found that Saycan usually fails with incorrect planning; while Voxposer has better planning ability, it mostly fails due to its suboptimal low-level policy, and its inability to re-plan upon failure attempts. Out planning part, with visual feedback and a critic agent, has nearly 100 % accuracy. Our mistakes mostly stem from low-level policy, which accumulates through each step.

Two Kitchen Results. To further investigate the generalization ability of our low-level policy, we also trained our model jointly on 2 kitchens and reported the accuracy rates of each skill in each kitchen. Because of the more diverse and complicated data in the two kitchen setting, we notice there is an accuracy decrease of about 10% to 20% accuracy in each of the skills (see Table 5).

Human Demonstrations. We conduct experiments to show that our model scales effectively with the number of human demonstrations provided. Increasing the number of demonstrations from 5 to 10 per task significantly improves the performance of the grasping and placing skills, while other skills can be learned fairly well with 5 demonstrations already (see Table 6).

4.2. Ablation Studies

We ablate two of our design choices: visual feedback and the critic agent. The key findings are: 1) visual feedback enables the robot to update its initial plan when there are environmental disturbances, and 2) it allows the robot to reattempt a skill if the previous attempt is unsuccessful. 3) The critic agent is essential when free-form language instructions are unclear or do not align with visual observations.

Ablation on Planning with Visual Feedback. We investigate the advantages of having GPT-4V's visual feedback in the execution loop through "random noise" and "environment disturbances" experiments (see Table 8 and Fig. 1). In the "random noise" experiments, uniform random noise is added to the predicted pose to simulate a flawed low-level policy (-0.05 to 0.05 in x/y, 0 to 0.08 in z for the "turn faucet" task, and -0.05 to 0.05 in x/y, -0.03 to 0.03 in z for the "close drawer" task), so the robot needs to retry tasks until successful completion. Our observations indicate that our method can replan effectively with visual feedback in the robot's execution loop while using GPT-4V only for one shot at the beginning and can't replan accordingly. Most errors in our methods stem from the robot arm colliding with the kitchen after the noise is introduced. Our baseline Voxposer, however, struggles to turn the faucet or open the kitchen drawer even without noise, thus failing in all the trials.

Task Description	Actions	Ours	Saycan + Owl-v2	Voxposer
Put all the fruits in the sink.	1) Pick up the pineapple. 2) Place it in the sink. 3) Pick up the starfruit. 4) Place it in the sink.	60% (80%)	20% (60%)	20% (70%)
Put the duck in the right drawer and close the drawer doors.	1) Pick up the duck. 2) Place it in the right drawer. 3) Close the left red drawer door. 4) Close the right orange drawer door.	40% (90%)	0% (20%)	0% (70%)
Wash the pineapple.	1) Pick up the pineapple. 2) Place it in the sink. 3) Turn faucet.	70% (90%)	0% (40 %)	10% (60%)

Table 7. Long horizon task accuracy. The notation $(\dots)$ refers to the accuracy of successfully finishing the first step.

	Voxposer		Ours w/o close-loop & critic		Ours	
random noise	turn faucet 0%	close left drawer 0%	turn faucet 40%	close left drawer 40%	turn faucet 80%	close left drawer 70%
environment disturbances	find fruits 0%	close both drawers 0%	find fruits 0%	close both drawers 0%	find fruits 60%	close both drawers 50%
unaligned instruction	pick pan 0%	open drawer 0%	pick pan 10%	open drawer 50%	pick pan 100%	open drawer 90%

Table 8. Ablation on GPT-4V close-loop planning and reflection. "Ours w/o close-loop & critic" means we only use GPT-4V planning once at the beginning without including the Critic Agent and updating the planning in the following steps.

In the "environment disturbances" experiments, we modify the scene after the robot's initial execution to determine if the in-the-loop update adjusts the plan according to the novel scene. In the "find fruits" task, the robot is required to pick up all the fruits in the scene and place them in the sink. Initially, only a pineapple is visible on the table, but a starfruit appears after the robot's first action. The experiment is successful if our framework updates the original plan and places the starfruit in the sink. In the "close both drawers" task, the robot is asked to close both drawers. After the robot closes one drawer, a human closes the other. Success is achieved if the framework updates the original plan to avoid closing the already closed drawer again. Our method can adapt to new scenes with a high accuracy rate (see Table 8). Our baseline Voxposer cannot update its plan once the environment changes, leading to failures in all tests again.

Ablation on Critic Agent. We find the critic agent in the execution loop useful when there is "unaligned instruction": human instruction is not fully aligned with visual observation. In the "pick pan" task, the robot is asked to pick up a pan not present in the scene. The experiment is successful if the framework correctly reasons and outputs "Done" directly. In the "open drawer" task, the robot is instructed to open a drawer that is already open. Success is achieved if the framework outputs "Done" directly without attempting to open it again. While the planner is easily skewed by human instruction, the critic agent can consistently base its reasoning on visual observation, since it does not take in human instruction as input. This corrects the planner, resulting in accurate plan outputs in subsequent iterations (see Table 8 and Fig. 1). Our baseline Voxposer has 0% in "unaligned instruction": it misinterprets the scene information by, for example, searching for a pan that is not present and attempting to open a drawer that is already open, which indicates a lack of understanding of the objects' states.

5. Conclusion

In this work, we propose LMM-3DP, a framework that includes LMMs as high-level planners and a language-conditioned 3D policy capable of learning various skills with only a few human demonstrations. Our experiments show that our high-level planning surpasses baselines by **1.5x** and our low-level control outperforms baselines by **1.45x**. These designs enable a significantly improved ability to handle environment disturbances and unaligned language instructions, execute various low-level skills in sequence, and recover from failed attempts. Our work's limitations include the need for careful prompt crafting, difficulty with tasks requiring continuous trajectories, and challenges in generalizing skills like picking objects to novel items with limited demonstrations.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 2

[2] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. Do as i can, not as i say: Grounding language in robotic affordances, 2022. 1, 2, 6

[3] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022. 1, 2

[4] Michael Ahn, Debidatta Dwibedi, Chelsea Finn, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Karol Hausman, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, et al. Autort: Embodied foundation models for large scale orchestration of robotic agents. *arXiv preprint arXiv:2401.12963*, 2024. 2

[5] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022. 3

[6] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023. 3

[7] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 3, 4

[8] Boyuan Chen, Fei Xia, Brian Ichter, Kanishka Rao, Keerthana Gopalakrishnan, Michael S Ryoo, Austin Stone, and Daniel Kappler. Open-vocabulary queryable scene representations for real world planning. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11509–11522. IEEE, 2023. 1, 2

[9] Shizhe Chen, Ricardo Garcia, Cordelia Schmid, and Ivan Laptev. Polarnet: 3d point clouds for language-guided robotic manipulation. *arXiv preprint arXiv:2309.15596*, 2023. 3

[10] Guangran Cheng, Chuheng Zhang, Wenzhe Cai, Li Zhao, Changyin Sun, and Jiang Bian. Empowering large language models on robotic manipulation with affordance prompting. *arXiv preprint arXiv:2404.11027*, 2024. 2

[11] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023. 3

[12] Ishita Dasgupta, Christine Kaeser-Chen, Kenneth Marino, Arun Ahuja, Sheila Babayan, Felix Hill, and Rob Fergus. Collaborating with language models for embodied reasoning. *arXiv preprint arXiv:2302.00763*, 2023. 1, 2

[13] Pete Florence, Corey Lynch, Andy Zeng, Oscar A Ramirez, Ayzaan Wahid, Laura Downs, Adrian Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning. In *Conference on Robot Learning*, pages 158–168. PMLR, 2022. 3

[14] Theophile Gervet, Zhou Xian, Nikolaos Gkanatsios, and Katerina Fragkiadaki. Act3d: 3d feature field transformers for multi-task robotic manipulation, 2023. 5

[15] Huy Ha, Pete Florence, and Shuran Song. Scaling up and distilling down: Language-guided robot skill acquisition. In *Conference on Robot Learning*, pages 3766–3777. PMLR, 2023. 3

[16] Daniel Honerkamp, Martin Büchner, Fabien Despinoy, Tim Welschehold, and Abhinav Valada. Language-grounded dynamic scene graphs for interactive object search with mobile manipulation. *IEEE Robotics and Automation Letters*, 2024. 2

[17] Yingdong Hu, Fanqi Lin, Tong Zhang, Li Yi, and Yang Gao. Look before you leap: Unveiling the power of gpt-4v in robotic vision-language planning. *arXiv preprint arXiv:2311.17842*, 2023. 2

[18] Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual language maps for robot navigation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10608–10615. IEEE, 2023. 2

[19] Haoxu Huang, Fanqi Lin, Yingdong Hu, Shengjie Wang, and Yang Gao. Copa: General robotic manipulation through spatial constraints of parts with foundation models. *arXiv preprint arXiv:2403.08248*, 2024. 2

[20] Siyuan Huang, Zhengkai Jiang, Hao Dong, Yu Qiao, Peng Gao, and Hongsheng Li. Instruct2act: Mapping multi-modality instructions to robotic actions with large language model. *arXiv preprint arXiv:2305.11176*, 2023. 2

[21] Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International conference on machine learning*, pages 9118–9147. PMLR, 2022. 1

[22] Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*, 2022. 1, 2

[23] Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*, 2023. 1, 2, 5, 6

[24] Xiaoqi Li, Mingxu Zhang, Yiran Geng, Haoran Geng, Yuxing Long, Yan Shen, Renrui Zhang, Jiaming Liu, and Hao Dong. Manipllm: Embodied multimodal large language model for object-centric robotic manipulation. *arXiv preprint arXiv:2312.16217*, 2023. 1

[25] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control, 2023. 1, 2

[26] Kevin Lin, Christopher Agia, Toki Migimatsu, Marco Pavone, and Jeannette Bohg. Text2motion: From natural language instructions to feasible plans. *Autonomous Robots*, 47 (8):1345–1365, 2023. 2

[27] Fangchen Liu, Kuan Fang, Pieter Abbeel, and Sergey Levine. Moka: Open-vocabulary robotic manipulation through mark-based visual prompting, 2024. 2, 5

[28] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis, 2020. 3

[29] Matthias Minderer, Alexey Gritsenko, and Neil Houlsby. Scaling open-vocabulary object detection. *Advances in Neural Information Processing Systems*, 36, 2024. 5, 6

[30] OpenAI. Gpt-4v(ision) system card, 2023. 2

[31] Abhishek Padalkar, Acorn Pooley, Ajinkya Jain, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anikait Singh, Anthony Brohan, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023. 3

[32] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. *Advances in Neural Information Processing Systems*, 35:23192–23204, 2022. 2, 3, 4

[33] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 4

[34] Shreyas Sundara Raman, Vanya Cohen, Eric Rosen, Ifrah Idrees, David Paulius, and Stefanie Tellex. Planning with large language models via corrective re-prompting. In *NeurIPS 2022 Foundation Models for Decision Making Workshop*, 2022. 1

[35] Krishan Rana, Jesse Haviland, Sourav Garg, Jad Abou-Chakra, Ian Reid, and Niko Suenderhauf. Sayplan: Grounding large language models using 3d scene graphs for scalable robot task planning, 2023. 1

[36] Krishan Rana, Jesse Haviland, Sourav Garg, Jad Abou-Chakra, Ian D Reid, and Niko Suenderhauf. Sayplan: Grounding large language models using 3d scene graphs for scalable task planning. *CoRR*, 2023. 2

[37] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, pages 785–799. PMLR, 2023. 3

[38] Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. Progprompt: Generating situated robot task plans using large language models. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11523–11530. IEEE, 2023. 1, 2

[39] Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M Sadler, Wei-Lun Chao, and Yu 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*, pages 2998–3009, 2023. 1

[40] Sai Vemprala, Rogerio Bonatti, Arthur Bucker, and Ashish Kapoor. Chatgpt for robotics: Design principles and model abilities, 2023. 2

[41] Naoki Wake, Atsushi Kanehira, Kazuhiro Sasabuchi, Jun Takamatsu, and Katsushi Ikeuchi. Chatgpt empowered long-step robot control in various environments: A case application. *IEEE Access*, 2023. 1

[42] Naoki Wake, Atsushi Kanehira, Kazuhiro Sasabuchi, Jun Takamatsu, and Katsushi Ikeuchi. Gpt-4v (ision) for robotics: Multimodal task planning from human demonstration. *arXiv preprint arXiv:2311.12015*, 2023. 2

[43] Peng Wang, Mattia Robbiani, and Zhihao Guo. Llm granularity for on-the-fly robot control. *arXiv preprint arXiv:2406.14653*, 2024. 2

[44] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023. 4

[45] Jimmy Wu, Rika Antonova, Adam Kan, Marion Lepert, Andy Zeng, Shuran Song, Jeannette Bohg, Szymon Rusinkiewicz, and Thomas Funkhouser. Tidybot: Personalized robot assistance with large language models. *Autonomous Robots*, 47(8):1087–1102, 2023. 2

[46] Ge Yan, Yueh-Hua Wu, and Xiaolong Wang. Dnact: Diffusion guided multi-task 3d policy learning. *arXiv preprint arXiv:2403.04115*, 2024. 3

[47] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022. 1

[48] Takahide Yoshida, Atsushi Masumori, and Takashi Ikegami. From text to motion: Grounding gpt-4 in a humanoid robot" alter3". *arXiv preprint arXiv:2312.06571*, 2023. 1, 2

[49] Yanjie Ze, Ge Yan, Yueh-Hua Wu, Annabella Macaluso, Yuying Ge, Jianglong Ye, Nicklas Hansen, Li Erran Li, and Xiaolong Wang. Gnfactor: Multi-task real robot learning with generalizable neural feature fields. In *Conference on Robot Learning*, pages 284–301. PMLR, 2023. 3

[50] Rowan Zellers, Ari Holtzman, Matthew Peters, Roozbeh Mottaghi, Aniruddha Kembhavi, Ali Farhadi, and Yejin Choi. Piglet: Language grounding through neuro-symbolic interaction in a 3d world. *arXiv preprint arXiv:2106.00188*, 2021. 2

[51] Andy Zeng, Maria Attarian, Brian Ichter, Krzysztof Choromanski, Adrian Wong, Stefan Welker, Federico Tombari, Aveek Purohit, Michael Ryoo, Vikas Sindhwani, et al. So-

cratic models: Composing zero-shot multimodal reasoning with language. *arXiv preprint arXiv:2204.00598*, 2022. 1, 2

[52] Tong Zhang, Yingdong Hu, Hanchen Cui, Hang Zhao, and Yang Gao. A universal semantic-geometric representation for robotic manipulation. *arXiv preprint arXiv:2306.10474*, 2023. 3

[53] Yifeng Zhu, Zhenyu Jiang, Peter Stone, and Yuke Zhu. Learning generalizable manipulation policies with object-centric 3d representations. *arXiv preprint arXiv:2310.14386*, 2023. 3