Safe Planner: Empowering Safety Awareness in Large Pre-Trained Models for Robot Task Planning

Siyuan Li[1], Feifan Liu[1], Lingfei Cui[2], Jiani Lu[1], Qinqin Xiao[1], Xirui Yang[3], Peng Liu[1], Kewu Sun[3*], Zhe Ma[3], Xun Wang[3]

[1]Harbin Institute of Technology
[2]Institute of Computer Application Technology, Norinco Group
[3]Intelligent Science & Technology Academy Limited of CASIC
siyuanli@hit.edu.cn

Abstract

Robot task planning is an important problem for autonomous robots in long-horizon challenging tasks. As large pre-trained models have demonstrated superior planning ability, recent research investigates utilizing large models to achieve autonomous planning for robots in diverse tasks. However, since the large models are pre-trained with Internet data and lack the knowledge of real task scenes, large models as planners may make unsafe decisions that hurt the robots and the surrounding environments. To solve this challenge, we propose a novel Safe Planner framework, which empowers safety awareness in large pre-trained models to accomplish safe and executable planning. In this framework, we develop a safety prediction module to guide the high-level large model planner, and this safety module trained in a simulator can be effectively transferred to real-world tasks. The proposed Safe Planner framework is evaluated on both simulated environments and real robots. The experiment results demonstrate that Safe Planner not only achieves state-of-the-art task success rates, but also substantially improves safety during task execution.

Extended version —
https://sites.google.com/view/safeplanner

Introduction

Robot task planning is a challenging temporal planning problem (Kaelbling and Lozano-Pérez 2013), aiming at obtaining a subtask sequence to accomplish a long-horizon target task, which is of great importance in the autonomous robot domain (Guo et al. 2023). Classical robot task planning methods are mostly based on search or STRIPS (Fikes and Nilsson 1993). However, these classical methods require much domain-specific knowledge and can hardly be applied to dynamic real-world environments. Following task-planning researchers pay attention to learning-based methods, e.g., reinforcement learning and imitation learning (Ceola et al. 2019; McDonald and Hadfield-Menell 2022; Jiang et al. 2019). Although these learning-based methods have shown promising results, they usually overfit the training scenes and cannot support natural languages as target task descriptions. Recently as large pre-trained models have demonstrated superior planning abilities in general settings, more and more researchers (Guan et al. 2023; Zhao, Lee, and Hsu 2024; Song et al. 2023) try to employ large pre-trained models to achieve autonomous robot task planning with languages as instructions.

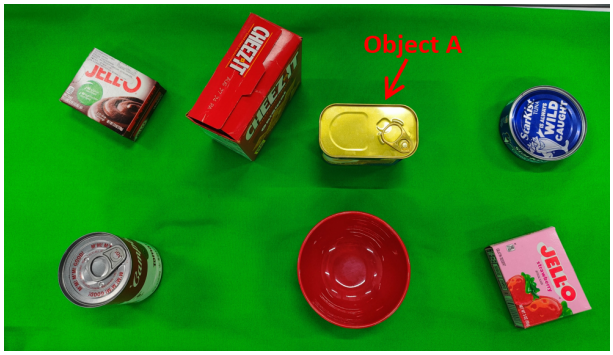

Figure 1: An illustrative example of safe planning.

Recent robot task planning methods based on large pre-trained models mainly utilize vision-language models (VLMs) and large language models (LLMs) to understand the task scenes and decompose the challenging long-horizon tasks (Driess et al. 2023; Hu et al. 2023; Huang et al. 2023b). These models are pre-trained with large-scale data, and thus are endowed with great generalization abilities. However, as lacking real-world training data, it is quite difficult for these models to accurately understand the task scenes, which may lead to unsafe and unexecutable planning results, causing damage to the robots and the real-world environments. For example, in a moving-object task shown in Figure 1, a bunch of objects has been put on one table, and the task is to move object A to a target position. The pre-trained models without real scene knowledge may output a direct planning solution, e.g., picking object A, and then placing it at the target place. However, as object A is surrounded by other objects, directly moving A may damage the nearby objects or cause collisions. Therefore, only using the pre-trained models is not enough, and a smarter planning model with safety awareness is needed, which is expected to move the objects nearby object A to other places before moving the target object A.

*Corresponding author.

To achieve safe planning in general robotics tasks, we propose a novel Safe Planner framework, which empowers safety awareness in large model based robot task planning. This framework is comprised of two levels: the high-level planner, and the low-level executors (skills). The high-level planner takes natural language task instruction and visual observations as inputs, and outputs the next skill to execute. For a structured interface between the two levels, we use the planning domain definition language (PDDL) (Fox and Long 2003) to define skills. As only using the large models pre-trained with Internet data as planners suffer from the unsafe problem, we develop a safety prediction module, which can predict the safety of executing the low-level skills. Then, the safety prediction results are leveraged to guide the planning process of the large pre-trained models, incentivize the reasoning ability of the large models, and thus promote safe planning in general robot tasks. Note that in real-world robot tasks, most unsafety is caused by collisions. Therefore in this work, the safety measure is defined with collision numbers, and the proposed framework can be easily adapted with a broader safety definition.

In the experiments, we compare the proposed framework with state-of-the-art large model planning methods in both the simulated environments and the real robot manipulation tasks. The experiment results demonstrate that Safe Planner not only achieves better task success rates, but also substantially improves safety during task execution. Besides, The safety prediction module trained with the simulated data can be effectively transferred to the real robot settings. To further investigate the safety module, we conduct thorough ablation studies on its design.

Preliminaries and Problem Statement

This section first briefly describes the preliminary knowledge for this work, including VLMs and PDDL, and then presents the problem statement.

Vision-Language Models

First, we introduce language models, which try to model the probability $p(W)$ of a text $W = \{w_0, w_1, w_2, ..., w_n\}$, a sequence of strings w. The probability $p(W)$ is generally modeled by the chain rule, $p(W) = \Pi_{j=0}^{n} p(w_j|w_{<j})$, so that each successive string can be predicted from the previous. Recent breakthroughs induced by the Attention mechanism (Vaswani et al. 2017) and Transformer architecture (Devlin et al. 2019) have enabled the efficient scaling of LLMs (Achiam et al. 2023; Touvron et al. 2023; Chowdhery et al. 2022), which have demonstrated increasingly large capacity and generalization ability. For real tasks grounded in the physical world, only the language modality is not enough, and VLMs augment LLMs with visual inputs to achieve the understanding of both images and languages. The visual inputs are processed with a vision encoder, such as ResNet (Alayrac et al. 2022) and vision transformers (ViT) (Driess et al. 2023; Chen et al. 2023b). In this work, we employ an LLM, GPT-4, and a VLM, GPT-4v to accomplish the proposed safe robot planning framework for physically grounded tasks.

Planning Domain Definition Language

The Planning Domain Definition Language (PDDL) (Fox and Long 2003) serves as a standardized encoding of planning problems. A PDDL planning problem is represented by two parts: a domain and a problem. Next, we describe these two terms informally and refer interested readers to comprehensive guides (Geffner and Bonet 2022).

- A PDDL *domain* defines the "universal" aspects of a problem, i.e., the elements in all specific problems such as object types, predicates, and operators. Types and predicates are used to describe the world state. For example, in a *House-Keeping* domain, the predicates may include "*in(X, Y): Is object X in container Y?*" and "*holding(X): Is the robot holding object X?*". An operator specifies a change to the state and is typically structured into three parts: preconditions, postconditions, and parameters. The preconditions determine whether the operator is applicable, and the postconditions define the effect of executing the operator. In the House-Keeping domain, the operators may include *pick(X)*, *place(X, Y)*, and *Open(X)*, and these operators can be regarded as robot skills.
- A PDDL problem specifies a list of objects, an initial state s_0, and a goal state s_g. Both s_0 and s_g are composed of a set of predicates. The types in the domain are used to describe the objects in the problem.

As PDDL predicates and operators have a clear structure, PDDL has been shown an effective interface in the LLM-based planning paradigm (Silver et al. 2024; Liu et al. 2023), which has alleviated the LLM hallucination problem and improved the generalization ability. Beyond that, since PDDL types, predicates, objects, and operators often include human-readable names like the ones above, PDDL boosts the ability of LLM to solve embodied planning problems.

Problem Statement

Assume that a set of planning tasks are in the same PDDL domain, the goals of these planning tasks are different, which are given in the form of natural language. This setting is common in the real world, e.g., a home service robot needs to do multiple household tasks, where these tasks are in the same house scenario, but have different goals, and these goals are described with natural languages. The planning framework aims to obtain a plan for each task, which can lead to both task completion and high safety. Here safety means that when the robot executes the plan, it will not damage the environment or itself.

Approach

Recently rich literature has tried to employ pre-trained LLMs and VLMs to accomplish comprehensive planning in complex long-horizon robotics tasks. In robotics tasks, the robot needs to interact with the physical world, but the models pre-trained with Internet data can hardly understand the real-world environments. Therefore, utilizing the large pre-trained models as a robotic task planner may hurt the robot or the environment. To address this problem, we propose a novel Safe Planner framework shown in Figure 2. In this

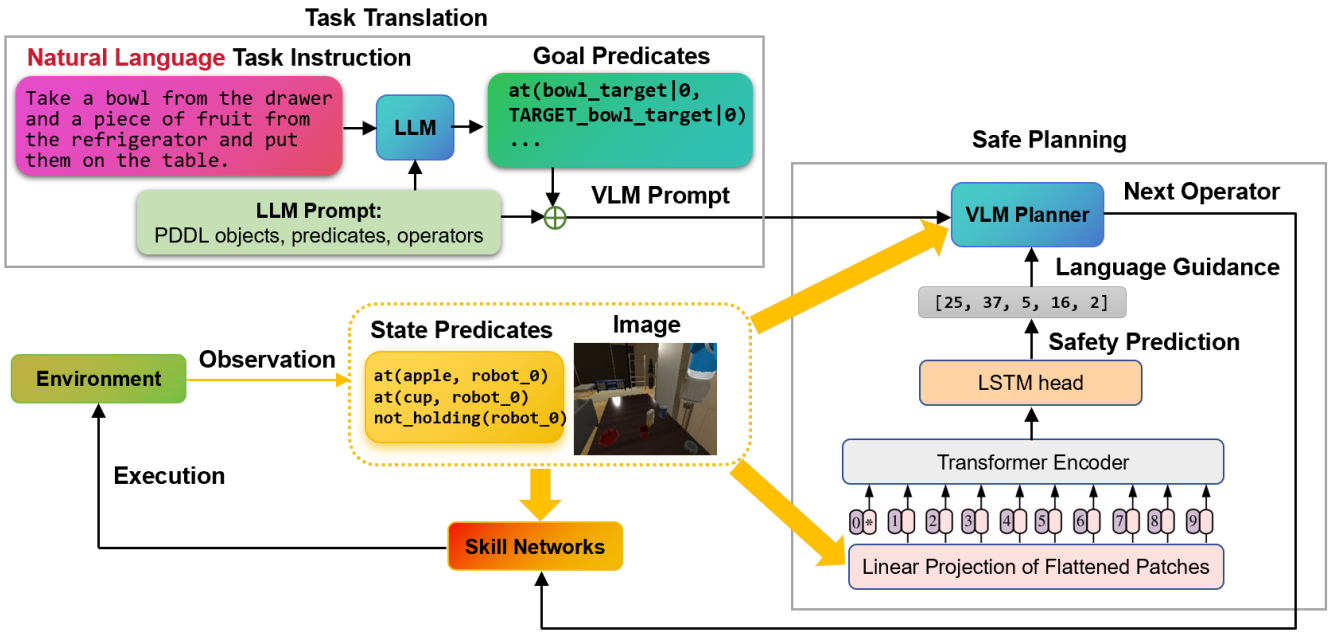

Figure 2: The Safe Planner framework. First, our framework translates the natural language task instruction into the PDDL goal predicates. Then, the goal predicates are combined with the PDDL domain as the prompt for the VLM planner. Note that in the Safe Planner framework, the VLM planner not only takes the observations as inputs, but also considers the safety of the operators, and outputs the plans. The next operator in the plan is executed in the environment, and the planner makes the next decision in a closed-loop way.

framework, the natural language task description is translated into the PDDL goals with an LLM. Taking the PDDL instruction, the current observation, and the safety prediction from the safety module as inputs, a VLM task planner outputs the plan in the form of PDDL operators, and the next operator is executed with low-level skills in the environment. After the skill execution completes or exceeds a preset timestep, the VLM planner replans in a closed-loop way. The following subsections elaborate on task translation and safe planning shown in the gray boxes in Figure 2.

Task Translation

Considering a robotic household scenario, human users usually describe the target tasks with natural languages, but for the robot, natural languages are not executable. Therefore, first of all, we need to translate the natural language task instruction to the language executable by robots. As PDDL is a well-structured planning language with high executability, we develop a PDDL-based task translator, as shown in the left-up box in Figure 2.

The inputs and outputs of the task translator are both languages. As the pre-trained LLMs have shown substantial language processing and reasoning abilities, we employ a pre-trained LLM as the main part of the task translator. To inspire the LLM to translate the task description into PDDL goals, we have designed a prompt template, which includes the PDDL domain information, the PDDL grammar, the translation requirement, and a successful translation example. Since the prompt template is quite long, we present it in the Appendix. The natural language task description, prompted by the designed template, is processed by an LLM to generate the corresponding PDDL goals. We utilize GPT-4 as the LLM in the task translator, and other pre-trained LLMs are applicable in the Safe Planner framework as well.

Safe Planning

Robot safety has long been a central issue for autonomous robots (Tzafestas 2013). As a broad concept, robot safety involves a lot of factors: collision with environments, collision with humans, velocity limits, force/torque limits, etc. Since unsafe actions with limit violations can be prohibited by a properly set action range, in this work we focus on robot safety concerning collisions, which is of great importance especially in dynamic environments. To achieve safe planning in long-horizon complex tasks, we propose a novel safety prediction module, which can predict the safety of executing skills based on the current observation. Then, the safety prediction can guide the VLM planner to accomplish safe and executable planning, as shown in the right part of Figure 2. Next, we elaborate on dataset collection, safety prediction module, and guidance to the VLM planner.

Dataset Collection: To diversify the training data, we randomize the initial scene of the tasks during data collection, i.e., the objects and the robot positions vary among episodes. In these diverse task scenes, the pre-trained low-level skills (corresponding to the PDDL operators) are executed to generate a trajectory set D. Then, we calculate the

Figure 3: The robot and target task scenes in the simulated experiments.

risk level of these trajectories as follows[1].

$$Risk(i, o) = C(robot, o^-) + \sum_{j,k} C(o_j, o_k)|_{j,k \in 1...N, j \neq k}, \tag{1}$$

where $Risk(i, o)$ denotes the risk of executing the i-th skill to manipulate the object o, and o^- denotes the objects except o in the current scene. C represents the collision number, which is calculated empirically with the collected trajectories in the simulators. Considering the influence of the robot on the scene $C(robot, o^-)$ and the induced collisions among the N objects $\sum_{j,k} C(o_j, o_k)|_{j,k \in 1...N, j \neq k}$, the risk level $Risk(i, o)$ is defined as Equation (1), which provides the labels for training the safety prediction module.

Safety Prediction Module: The safety prediction module is a multi-head neural network $f_\theta(obs, o; i)|i \in 1...I$, where the i-th head predicts the safety of executing the i-th skill in the skill set of size I, o is the object to manipulate and obs is the initial image observation in the trajectory of skill execution. To enable a comprehensive description of the object o, we represent o with images of the object from five different views[2], so the input of the safety prediction module is a set of images. In the safety prediction module, a large-sized ViT encoder (Wu et al. 2020) pre-trained with the ImageNet dataset (Deng et al. 2009) is employed to extract effective representations from the input images, and then the representations are processed with a learnable LSTM network (Hochreiter and Schmidhuber 1997) to predict the risk level $Risk(i, o)$ in Equation (1). Here, we use the LSTM network since the inputs are treated as a sequence of images. LSTM is lightweight and fulfills real-time and training data requirements of embodied agents. Regarding the safety prediction problem as a regression problem, the LSTM is optimized with the mean squared error (MSE) loss, and the pre-trained ViT encoder is frozen. Note that the target output of the safety prediction module is the expectation of Equation (1), the proposed framework can handle stochastic skills.

Guidance to VLM: To achieve safe planning, the VLM planner requires guidance from the safety prediction module. As a thorough safety prediction, we formulate the guidance with a matrix of size $I \times N$, where the rows correspond to the pre-trained skills, and the columns correspond to the objects in the available area of the robot. The safety matrix can be obtained by inferring the safety module with the N objects. The next question is how to guide the VLM planner with this safety matrix. As the exact values in this matrix are difficult to understand by VLMs, we transform the numeric values in the matrix into rankings in the form of natural languages. This transformation from the safety matrix to the ranking sentences relies on the PDDL operator names and object names. An example feedback of the safety module to the VLM planner is shown as follows:

The safest operator is to pick the bowl. The second safest operator is to pick the apple. $\cdots$

Besides the safety guidance, the inputs of the VLM planner include the task information from the task translation module as the prompt for VLM, and the current observation of state predicates and images. The details about the VLM prompt are provided in the Appendix. With these inputs, the VLM planner outputs the plan in the form of PDDL operators. Then, the pre-trained low-level skill corresponding to the next operator is executed. If the skill execution satisfies the success criteria or exceeds a certain timestep, the VLM replans with the renewed observation and safety matrix. In this work, we use GPT-4v as the VLM planner and utilize the PPO algorithm (Schulman et al. 2017) to pre-train low-level skills. Note that other VLMs and skill pretaining algorithms are also applicable.

Related Work

This work is mostly related to large model planning and robot safety. Next, we give a brief literature review about these two directions, and discuss the relationship between our work and previous literature.

Planning with Large Models. Recently, robot task planning with large pre-trained models has drawn much attention from embodied artificial intelligence researchers (Driess et al. 2023; Song et al. 2023; Guan et al. 2023; Wang et al. 2024). In most related works, the large pre-trained models serve as a high-level planner, which instructs low-level executors (Brohan et al. 2023; Hu et al. 2023; Singh et al. 2023; Huang et al. 2023b). To alleviate the hallucination issue in LLM planning, existing works propose to use structured interfaces to regularize the outputs of LLM, such as PDDL (Liu et al. 2023; Silver et al. 2024; Pallagani et al.

[1] N denotes the number of objects in the robot available area.

[2] The five views include the top-down view, the front-rear one and its opposite, the left-right one and its opposite. The images from these views can be automatically generated in the simulator with the object 3D file formatted in GLB.

Scene	*Mode*	Safe Planner	Safe Planner w/o SM	SayCan	PROGPROMPT
Table	Easy	**2.44**	3.80	3.40	3.05
	Hard	**4.33**	5.42	6.03	6.23
Counter	Easy	**2.07**	2.63	3.08	3.13
	Hard	**1.73**	3.20	3.53	3.71
Chair	Easy	**1.26**	2.30	2.28	2.26
	Hard	**6.21**	10.59	10.50	10.51

Table 1: The comparison results of the average collision numbers.

Scene	*Mode*	Safe Planner	Safe Planner w/o SM	SayCan	PROGPROMPT
Table	Easy	**0.70**	0.67	0.43	0.63
	Hard	0.73	**0.80**	0.67	0.73
Counter	Easy	**0.40**	**0.40**	0.29	0.38
	Hard	**0.40**	**0.40**	0.37	0.33
Chair	Easy	0.90	**0.95**	0.90	0.91
	Hard	0.82	**0.83**	0.81	0.80

Table 2: The comparison results of the average task success rates.

2022). However, these methods lack the real scene grounding, and thus may output plans that hurt the robot or environment. To address this problem, Huang et al. (2023a) propose Grounded Decoding, which guided the LLM planning with a hand-crafted safety score. The following works propose to employ VLMs to enhance the scene understanding abilities and improve planning safety (Hu et al. 2023; Chen et al. 2023a). In contrast to the previous works, the proposed Safe Planner framework explicitly learns a safety prediction module, which can guide the VLM planner to achieve safe planning in complex long-horizon tasks.

Safety in Robotics. Safety has long been a critical issue for autonomous robots, especially in the human-robot interaction setting (Tzafestas 2013; Vicentini 2021). Robot safety involves multiple perspectives, e.g., mechanical limitations, collisions, etc. This work considers robot safety from the collision avoidance view. An early related work builds high-level symbolic representations that capture physical interactions to avoid collisions (Mojtahedzadeh et al. 2015). The following work develops an object-level scene understanding approach, SafePicking, to improve manipulation safety (Wada, James, and Davison 2022). A recent work proposes to build a relation graph of the whole scene to promote robot safety (Li et al. 2024). These related works mainly address the safety problem with a small model, which limits their applicability to general settings. To the best of our knowledge, the proposed framework is the first to induce a learnable safety prediction module into large model planning, which injects safety awareness into general robotics planning with large models.

Experiments

In this section, we conduct experiments to answer the following questions: (1) Can the Safe Planner framework achieve state-of-the-art performance in challenging tasks? (2) Is the proposed framework applicable to real robots? (3) How is the planning result with safety awareness compared to that without safety awareness? (4) How can the safety module design influence the planning performance? Next, we briefly review the baseline methods and then demonstrate the experiment results.

Baselines

We compare Safe Planner with state-of-the-art methods in the large model planning domain, and all the comparison methods are listed as follows. For a fair comparison, in all the methods, the LLMs are implemented with GPT-4, and the VLMs are implemented with GPT-4v.

- Safe Planner: Empowering safety awareness in large models with a learnable safety prediction module.
- Safe Planner w/o SM: Removing the safety module (SM) in the proposed framework, which can be regarded as an ablation study.
- SayCan (Brohan et al. 2023): LLM planner with value functions as affordances and languages as interfaces.
- PROGPROMPT (Singh et al. 2023): Prompting LLMs to output Pythonic task plans.

Results in Simulated Environments

Task Settings: The simulated experiments are conducted in the Habitat 2.0 environment (Szot et al. 2021), where a mobile manipulator (Fetch robot) equipped with two RGBD cameras mounted on its head and arm is instructed to do housework. In the simulated experiments, the state predicates are obtained through reading data from the simulation engine, e.g., by reading the object position, we can get the "at(X,Y)" predicate, and by reading the gripper state, we can get the "not_holding(robot)" predicate. The experiments in this subsection involve three target task scenes, as shown in Figure 3(b)-(d). Due to the tight spaces, the *Chair* and *Counter* scenes are more difficult than the *Table* scene. The

long-horizon challenging tasks in the experiments are composed of both navigation subtasks and manipulation subtasks, i.e., the robot needs to first navigate to the target task scene, and then manipulate the target objects as instructed by the natural languages, e.g., "*Move the apple on the table to the chair*". The detailed natural language task instructions are listed in the Appendix. As a thorough evaluation of Safe Planner, we design two task modes with different difficulty levels in each target task scene: in the *"easy"* mode, there are totally 3 objects in the manipulation areas denoted with the red rectangles in Figure 3; in the *"hard"* mode, there are no less than 5 objects in the manipulation areas. More objects lead to complex obstacle geometry and collisions easily happening, and thus are reckoned as *hard*.

Metrics: We use two metrics to compare the performance of these planning methods: task success rates to evaluate executability, and the total collision numbers in one trajectory as calculated by Equation (1) to evaluate safety. The average results of 100 runs are shown in Tables 1 and 2.

Analysis: The experiment results demonstrate that the proposed Safe Planner framework can effectively reduce the collision numbers during task execution and output planning results with higher safety. The improvement of safety (reducing collisions) is at a little sacrifice of success rates, since with safety awareness, the skill sequences output by large models are longer than those without safety awareness, as the Safe Planner first moves the obstacles away to ensure the safe execution. An example planning result is shown in the next subsection, and the longer skill sequence leads to a lower success rate in the whole task due to the chaining rule. The success rate in the *Counter* task is lower due to the obstruction of the cabinets above the counter, and the countertop is not entirely in the operational range of the robot. For the same reason, the low-level skill success rates in the *Counter* setting, such as Pick and Place, are lower. Therefore, even if the high-level planning by our approach is correct, the success rate can not be as high as in other settings.

Comparing the last three columns in Tables 1 and 2, we find that without the safety prediction module, the collision number of the proposed framework is nearly the same as the baseline methods. However, the success rates of Safe Planner w/o SM are larger, which indicates the PDDL interface of Safe Planner is more effective than the natural language interface of SayCan and the Pythonic interface of PROG-PROMPT.

Results on Real Robots

Task Settings: The real-world experiments are conducted on a fixed-based 7-dof Franka Panda robot arm, as shown in Figure 4. As the collision data on the real robots is difficult to obtain, the safety prediction module trained with the simulated data is directly transferred to the real-world setting in a zero-shot manner without fine-tuning. The tasks in this subsection are about robot arm manipulation. Specifically, according to a natural language task instruction, the robot arm needs to pick the target object from a cluttered table, and then place this object at the target place. To achieve a thorough evaluation, the table is set with different levels of clutter: in the *easy* mode, there are 3 objects on the table, so accomplishing the Pick-Place task without hurting the surrounding objects is relatively easy. In the setting with *medium* difficulty, there are 5 objects on the table, and in the *hard* one, there are 7 objects[3]. Note that in the real robot experiments and the simulated experiments, we use different types of robots, so transferring the manipulation skills from sim to real is quite challenging. Therefore, for the real robot arm, the low-level manipulation skills are implemented by hand-crafted scripts with object pose estimation.

Figure 4: The task scene for real robot experiments.

Metrics: In the real robot experiments, we also evaluate the performance from two perspectives: task success rates and safety. As the collision number in the real-world setting is not easy to calculate, we measure the safety by examining the final state after task execution. If the robot collides with one object unrelated to the task, which causes the object position changed in the final state, we add 1 to the *collision* metric. If the collision is more severe, which causes the turning over of the object, we add 2 to the *collision* metric. The collisions with all the task-unrelated objects are summed together as the *"collisions"* metric. The results averaged over 10 runs are shown in Tables 3 and 4.

	Easy	Medium	Hard
Safe Planner	**0.2**	**1.0**	**1.0**
Safe Planner w/o SM	2.0	4.2	2.1

Table 3: The average *collisions* in the real robot tasks.

	Easy	Medium	Hard
Safe Planner	**1.0**	**1.0**	**0.7**
Safe Planner w/o SM	**1.0**	**1.0**	0

Table 4: The average success rates in the real robot tasks.

Analysis: Table 3 indicates that the safety module (SM)

[3]More detailed real robot task settings with photos are listed in the Appendix.

can significantly reduce the collision numbers and improve safety during manipulation in all the task settings. As the task success is measured by the final state of the target objects (no matter with the surrounding objects), in the easy and medium settings, with and without SM both can achieve a 100% success rate. However, the success of without SM is at the cost of collisions. The success rates in the easy and medium settings are a little higher than those in the simulated environment, since the low-level skills on the real robot with hand-crafted scripts are more accurate than the skills trained with the PPO method in the simulator. In the hard task, the objects are closely piled together. Only if the objects surrounding the target have been cleared up, the target object can be successfully picked. w/o SM, the robot directly executes the pick-target skill and place-target skill, so it fails. In contrast, the SM model predicts the collisions with surroundings and guides the VLM to move other objects first and then pick up the target. The success rates on the real robot are higher than in simulation, since the low-level skills in these two settings are implemented in different ways. On the real robot, the low-level skills are implemented with predefined scripts, which have higher success rates. In simulation, the low-level skills are trained with the reinforcement learning method, PPO, and the success rates are lower than those predefined scripts.

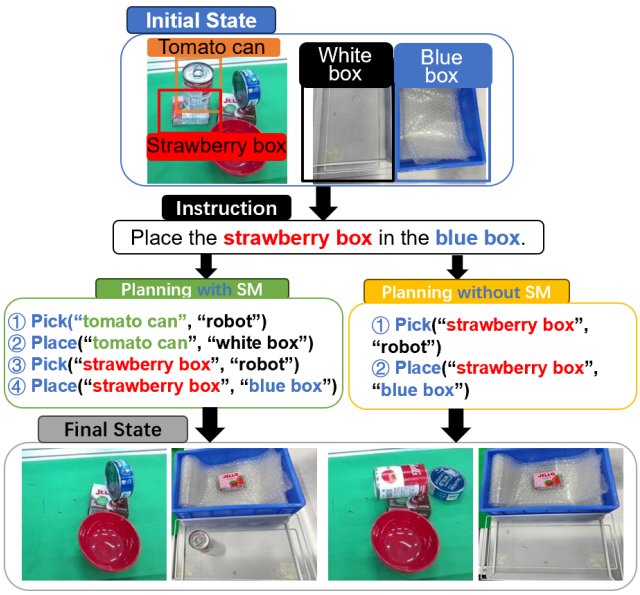

Figure 5: Comparison of the skill sequences planned by the model with SM (left) and without SM (right).

Skill Sequence Comparison: Diving into why Safe Planner has such great performance, we showcase the planning results on an example task (the task with *medium* difficulty). As shown in Figure 5, the objects are cluttered on the table at the initial state, and the instruction is to put the strawberry box under the tomato can into the blue box. Without safety awareness, the planner directly picks and places the target object (the strawberry box), which causes collisions with the tomato can and the blue can, as shown in the right part. In contrast, guided by the SM, the planner first moves the tomato can away, and then manipulates the strawberry box, which keeps the objects surrounding the strawberry box safe, as shown in the left part of Figure 5. The videos for more tasks are shown in https://sites.google.com/view/safeplanner.

Ablation Studies on Safety Module Design

In this subsection, we conduct ablation studies on the safety prediction module to investigate which vision encoder can achieve better prediction performance. In Figure 6 and Table 5, we compare the ResNet encoder with the ViT encoder, and their pre-trained and from-scratch versions. The y axis of Figure 6 is the MSE loss, and the x axis is the training epoch. With pre-training on the ImageNet dataset, both the ViT encoder and the ResNet encoder can converge to a small regression loss, and the ViT encoder has a faster convergent speed. Therefore, we employ the pre-trained ViT encoder as the vision backbone in the safety prediction module of the Safe Planner framework.

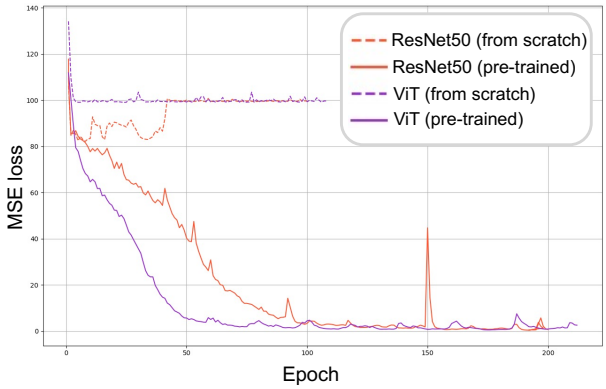

Figure 6: The regression losses of different safety prediction modules during training.

We further investigate the relationship between the convergent regression losses with the large model planning performance in the Chair (Hard) task. From Table 5, we can see that a smaller regression loss of the safety module reduces the collision numbers in the robot task. As for the vision encoders from scratch, the regression loss can hardly be optimized, so the collision numbers for these models are consistent with that in the last row of Table 1 without the safety module guidance.

model	convergent loss	collisions
Resnet50 (from scratch)	100.1	10.59
ResNet50 (pre-trained)	1.9	6.48
ViT (from scratch)	99.2	10.52
ViT (pre-trained)	**1.2**	**6.21**

Table 5: The convergent regression losses of safety modules and the collisions in the Chair (Hard) task.

Conclusion

To address the problem of large model planners lacking physics-grounded safety awareness, we propose a novel Safe Planner framework, which guides the large pre-trained models with a learnable safety prediction module to accomplish safe and executable robot task planning. This safety prediction module trained in simulated environments can be effectively transferred to real-world tasks. Experiment results on both simulators and real robots demonstrate that Safe Planner can significantly improve safety and avoid collisions during task execution. Beyond that, we conduct thorough ablation studies on the safety module design and find that the pretraining of the vision encoder is quite important.

For future work, the scene understanding ability of large pre-trained models needs to be further enhanced, so that the planner's reliance on the state predicates can be alleviated. Another interesting future direction is large model quantization. With a light model, the inference speed can be improved, which can promote real-time planning for the embodied agents. Furthermore, as the safety in this work mainly focuses on the collision aspect, extending the safety scope is an important future direction, e.g., force control, velocity limits, and safe human-robot interaction aspects.

Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable comments and helpful suggestions. The work is supported by the National Natural Science Foundation of China (62306088) and Natural Science Foundation of Heilongjiang Province (YQ2024007).

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Alayrac, J.-B.; Donahue, J.; Luc, P.; Miech, A.; Barr, I.; Hasson, Y.; Lenc, K.; Mensch, A.; Millican, K.; Reynolds, M.; et al. 2022. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35: 23716–23736.

Brohan, A.; Chebotar, Y.; Finn, C.; Hausman, K.; Herzog, A.; Ho, D.; Ibarz, J.; Irpan, A.; Jang, E.; Julian, R.; et al. 2023. Do as i can, not as i say: Grounding language in robotic affordances. In *Conference on robot learning*, 287–318. PMLR.

Ceola, F.; Tosello, E.; Tagliapietra, L.; Nicola, G.; and Ghidoni, S. 2019. Robot task planning via deep reinforcement learning: a tabletop object sorting application. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 486–492. IEEE.

Chen, B.; Xia, F.; Ichter, B.; Rao, K.; Gopalakrishnan, K.; Ryoo, M. S.; Stone, A.; and Kappler, D. 2023a. Open-vocabulary queryable scene representations for real world planning. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 11509–11522. IEEE.

Chen, X.; Djolonga, J.; Padlewski, P.; Mustafa, B.; Changpinyo, S.; Wu, J.; Ruiz, C. R.; Goodman, S.; Wang, X.; Tay, Y.; et al. 2023b. Pali-x: On scaling up a multilingual vision and language model. *arXiv preprint arXiv:2305.18565*.

Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; Chung, H. W.; Sutton, C.; Gehrmann, S.; et al. 2022. Palm: Scaling language modeling with pathways. arXiv 2022. *arXiv preprint arXiv:2204.02311*, 10.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186.

Driess, D.; Xia, F.; Sajjadi, M. S.; Lynch, C.; Chowdhery, A.; Ichter, B.; Wahid, A.; Tompson, J.; Vuong, Q.; Yu, T.; et al. 2023. PaLM-E: An Embodied Multimodal Language Model. In *International Conference on Machine Learning*, 8469–8488. PMLR.

Fikes, R.; and Nilsson, N. J. 1993. STRIPS, a retrospective. *Artificial intelligence*, 59(1-2): 227–232.

Fox, M.; and Long, D. 2003. PDDL2. 1: An extension to PDDL for expressing temporal planning domains. *Journal of artificial intelligence research*, 20: 61–124.

Geffner, H.; and Bonet, B. 2022. *A concise introduction to models and methods for automated planning*. Springer Nature.

Guan, L.; Valmeekam, K.; Sreedharan, S.; and Kambhampati, S. 2023. Leveraging pre-trained large language models to construct and utilize world models for model-based task planning. *Advances in Neural Information Processing Systems*, 36: 79081–79094.

Guo, H.; Wu, F.; Qin, Y.; Li, R.; Li, K.; and Li, K. 2023. Recent trends in task and motion planning for robotics: A survey. *ACM Computing Surveys*, 55(13s): 1–36.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation*, 9(8): 1735–1780.

Hu, Y.; Lin, F.; Zhang, T.; Yi, L.; and Gao, Y. 2023. Look before you leap: Unveiling the power of gpt-4v in robotic vision-language planning. *arXiv preprint arXiv:2311.17842*.

Huang, W.; Xia, F.; Shah, D.; Driess, D.; Zeng, A.; Lu, Y.; Florence, P.; Mordatch, I.; Levine, S.; Hausman, K.; et al. 2023a. Grounded decoding: Guiding text generation with grounded models for robot control. *arXiv preprint arXiv:2303.00855*.

Huang, W.; Xia, F.; Xiao, T.; Chan, H.; Liang, J.; Florence, P.; Zeng, A.; Tompson, J.; Mordatch, I.; Chebotar, Y.; et al. 2023b. Inner Monologue: Embodied Reasoning through Planning with Language Models. In *Conference on Robot Learning*, 1769–1782. PMLR.

Jiang, Y.; Yang, F.; Zhang, S.; and Stone, P. 2019. Task-motion planning with reinforcement learning for adaptable mobile service robots. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 7529–7534. IEEE.

Kaelbling, L. P.; and Lozano-Pérez, T. 2013. Integrated task and motion planning in belief space. *The International Journal of Robotics Research*, 32(9-10): 1194–1227.

Li, Y.; Wu, R.; Lu, H.; Ning, C.; Shen, Y.; Zhan, G.; and Dong, H. 2024. Broadcasting Support Relations Recursively from Local Dynamics for Object Retrieval in Clutters. *arXiv preprint arXiv:2406.02283*.

Liu, B.; Jiang, Y.; Zhang, X.; Liu, Q.; Zhang, S.; Biswas, J.; and Stone, P. 2023. Llm+ p: Empowering large language models with optimal planning proficiency. *arXiv preprint arXiv:2304.11477*.

McDonald, M. J.; and Hadfield-Menell, D. 2022. Guided imitation of task and motion planning. In *Conference on Robot Learning*, 630–640. PMLR.

Mojtahedzadeh, R.; Bouguerra, A.; Schaffernicht, E.; and Lilienthal, A. J. 2015. Support relation analysis and decision making for safe robotic manipulation tasks. *Robotics and Autonomous Systems*, 71: 99–117.

Pallagani, V.; Muppasani, B.; Murugesan, K.; Rossi, F.; Horesh, L.; Srivastava, B.; Fabiano, F.; and Loreggia, A. 2022. Plansformer: Generating symbolic plans using transformers. *arXiv preprint arXiv:2212.08681*.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Silver, T.; Dan, S.; Srinivas, K.; Tenenbaum, J. B.; Kaelbling, L.; and Katz, M. 2024. Generalized planning in pddl domains with pretrained large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 18, 20256–20264.

Singh, I.; Blukis, V.; Mousavian, A.; Goyal, A.; Xu, D.; Tremblay, J.; Fox, D.; Thomason, J.; and Garg, A. 2023. Progprompt: Generating situated robot task plans using large language models. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 11523–11530. IEEE.

Song, C. H.; Wu, J.; Washington, C.; Sadler, B. M.; Chao, W.-L.; and Su, Y. 2023. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2998–3009.

Szot, A.; Clegg, A.; Undersander, E.; Wijmans, E.; Zhao, Y.; Turner, J.; Maestre, N.; Mukadam, M.; Chaplot, D. S.; Maksymets, O.; et al. 2021. Habitat 2.0: Training home assistants to rearrange their habitat. *Advances in neural information processing systems*, 34: 251–266.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Tzafestas, S. G. 2013. *Introduction to mobile robot control*. Elsevier.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Vicentini, F. 2021. Collaborative robotics: a survey. *Journal of Mechanical Design*, 143(4): 040802.

Wada, K.; James, S.; and Davison, A. J. 2022. Safepicking: Learning safe object extraction via object-level mapping. In *2022 International Conference on Robotics and Automation (ICRA)*, 10202–10208. IEEE.

Wang, S.; Han, M.; Jiao, Z.; Zhang, Z.; Wu, Y. N.; Zhu, S.-C.; and Liu, H. 2024. LLMˆ 3: Large Language Model-based Task and Motion Planning with Motion Failure Reasoning. *arXiv preprint arXiv:2403.11552*.

Wu, B.; Xu, C.; Dai, X.; Wan, A.; Zhang, P.; Yan, Z.; Tomizuka, M.; Gonzalez, J.; Keutzer, K.; and Vajda, P. 2020. Visual Transformers: Token-based Image Representation and Processing for Computer Vision. arXiv:2006.03677.

Zhao, Z.; Lee, W. S.; and Hsu, D. 2024. Large language models as commonsense knowledge for large-scale task planning. *Advances in Neural Information Processing Systems*, 36.