Robotics: Science and Systems 2024
Delft, Netherlands, July 15-July 19, 2024

VLMPC: Vision-Language Model Predictive Control for Robotic Manipulation

Wentao Zhao*, Jiaming Chen*, Ziyu Meng, Donghui Mao, Ran Song†, Wei Zhang
School of Control Science and Engineering, Shandong University, China
E-mail: *zwt7@mail.sdu.edu.cn; ppjmchen@gmail.com; mziyu@mail.sdu.edu.cn; donghui.mao@mail.sdu.edu.cn; ransong@sdu.edu.cn; davidzhang@sdu.edu.cn*

Abstract—**Although Model Predictive Control (MPC) can effectively predict the future states of a system and thus is widely used in robotic manipulation tasks, it does not have the capability of environmental perception, leading to the failure in some complex scenarios. To address this issue, we introduce Vision-Language Model Predictive Control (VLMPC), a robotic manipulation framework which takes advantage of the powerful perception capability of vision language model (VLM) and integrates it with MPC. Specifically, we propose a conditional action sampling module which takes as input a goal image or a language instruction and leverages VLM to sample a set of candidate action sequences. Then, a lightweight action-conditioned video prediction model is designed to generate a set of future frames conditioned on the candidate action sequences. VLMPC produces the optimal action sequence with the assistance of VLM through a hierarchical cost function that formulates both pixel-level and knowledge-level consistence between the current observation and the goal image. We demonstrate that VLMPC outperforms the state-of-the-art methods on public benchmarks. More importantly, our method showcases excellent performance in various real-world tasks of robotic manipulation. Code is available at https://github.com/PPjmchen/VLMPC.**

I. Introduction

Burgeoning foundation models [54, 11, 14, 9, 17] have demonstrated powerful capabilities of knowledge extraction and reasoning. Exploration based on foundation models has thus flourished in many fields such as computer vision [45, 13, 15, 5], AI for science [8], healthcare [49, 66, 80, 57], and robotics [10, 25, 61, 77, 48]. Recently, a wealth of work has made significant progress in incorporating foundation models into robotics. These works usually leveraged the strong understanding and reasoning capabilities of versatile foundation models on multimodal data including language [34, 10, 61, 77, 62, 48], image [34, 44] and video [10] for enhancing robotic perception and decision-making.

To achieve knowledge transfer from foundation models to robots, most early works concentrate on robotic planning [32, 33, 12, 72, 64, 59, 65, 43, 42, 16, 78, 74, 46, 56, 52], which directly utilize large language models (LLMs) to decompose high-level natural language command and abstract tasks into low-level and pre-defined primitives (*i.e.,* executable actions or skills). Although such schemes intuitively enable robots to perform complex and long-horizon tasks, they lack the capability of visual perception. Consequently, they heavily rely on pre-defined individual skills to interact with specific physical entities, which limits the flexibility of robotic planning. Recent works [34, 10, 71, 31] remedy this issue by integrating with large-scale vision-language models (VLMs) to improve scene perception and generate trajectories adaptively for robotic manipulation in intricate scenarios without using pre-defined primitives.

Although existing methods have shown promising results in incorporating foundation models into robotic manipulation, interaction with a wide variety of objects and humans in the real world remains a challenge. Specifically, since the future states of a robot are not fully considered in the decision-making loop of such methods, the reasoning of foundation models is primarily based on current observations, resulting in insufficient forward-looking planning. For example, when opening a drawer, the latest method based on VLM [34] cannot directly generate an accurate trajectory to pull the drawer handle due to the lack of prediction on the future state, and thus it still requires to design specific primitives on object-level interaction. Hence, it is desirable to develop a robotic framework that performs with a human-like *"look before you leap"* ability.

Model predictive control (MPC) is a control strategy widely used in robotics [63, 3, 29, 73, 38]. MPC possesses an appealing attribute of predicting the future states of a system through a predictive model. Such forward-looking attribute allows robots to plan their actions by considering potential future scenarios, thereby enhancing their ability to interact dynamically with various environments. Traditional MPC [63, 29, 73, 68, 24] usually builds a deterministic and sophisticated dynamic model corresponding to the task and environment, which does not adapt well to intricate scenes in the real world. Recent research [21, 76, 51, 75, 67, 20] has explored using vision-based predictive models to learn dynamic models from visual inputs and predict high-dimensional future states in 2D [21, 76, 67, 20] or 3D [21, 51, 75, 20] spaces. Such methods are based on current visual observation for proposing manipulation actions in the MPC loop, which enables robots to make more reasonable decisions based on visual clues. Nevertheless, the effectiveness of such methods is constrained by the limitations inherent in visual predictive models trained on finite datasets. Such models struggle to accurately predict scenarios involving scenes or objects they have not previously encountered. This issue becomes especially pronounced in the

*These authors contributed equally to this work.
†Corresponding author: Ran Song.

real-world environments often partially or even fully unseen to robots, where the models can only perform basic tasks that align closely with their training data.

Naturally, large-scale vision-language models have the potential to address this problem by providing extensive open-domain knowledge and offering a more comprehensive understanding of diverse and unseen scenarios, thereby enhancing the predictive accuracy and adaptability of the scheme for robotic manipulation. Thus, this work presents the **V**ision-**L**anguage **M**odel **P**redictive **C**ontrol (**VLMPC**), a framework that combines VLMs and model predictive control to guide the robotic manipulation with complicated path planning including rotation and interaction with scene objects. By leveraging the strong ability of visual reasoning and visual grounding for sampling actions provided by VLM, VLMPC avoids the manual design of individual primitives, and addresses the limitation that previous methods based on VLMs can only compose coarse trajectories without foresight.

As shown in Fig. 1, VLMPC takes as input either a goal image indicating the prospective state or a language instruction. We propose an action sampling module that uses VLM to initialize the task and handle the current observation, which generates a conditional action sampling distribution for further producing a set of action sequences. With the action sequences and the history image observation, VLMPC adopts a lightweight action-conditioned video prediction model to predict a set of future frames. To assess the quality of the candidate action sequences through the future frames, we also design a hierarchical cost function composed of two sub-costs: a pixel-level cost measuring the difference between the video predictions and the goal image and a knowledge-level cost making a comprehensive evaluation on the video predictions. VLMPC finally chooses the action sequence corresponding to the best video prediction, and then picks the first action from the sequence to execute while feeding the subsequent actions into the action sampling module combined with conditional action sampling.

The main contributions of this paper are as follows:

- We propose VLMPC for robotic manipulation, which incorporates a learning-based dynamic model to predict future video frames and seamlessly integrates VLM into the MPC loop for open-set knowledge reasoning.
- We design a conditional action sampling module to sample robot actions from a visual perspective and a hierarchical cost function to provide a comprehensive and coarse-to-fine assessment of video predictions.
- We conduct experiments in both simulated and real-world scenes to demonstrate that VLMPC provides good knowledge reasoning and effective foresight, achieving state-of-the-art performance without any primitives.

II. Related Work

Since the proposed VLMPC integrates MPC with foundation models, this section reviews them in the context of robotic manipulation.

A. Model Predictive Control for Robotic Manipulation

Model predictive control (MPC) is a multivariate control algorithm widely used in robotics [63, 3, 29, 73, 38, 28, 53, 68, 24, 34, 21]. It employs a predictive model to estimate future system states, subsequently formulating the control law through solving a constrained optimization problem [27, 28]. The foresight capability of MPC, combined with its constraint-handling features, enables the development of advanced robotic systems which operate safely and efficiently in variable environments [29].

In the context of robotic manipulation, the role of MPC is to make the robot manipulator move and act in an optimal way with respect to input and output constraints [7, 21, 23, 75, 76, 51, 67]. In particular, action-based predictive models are frequently used in MPC for robotic manipulation, referring to a prediction model designed to forecast the potential future outcomes of specific actions, connecting sequence data to decision-making processes. Bhardwaj *et al.* [7] proposed a sampling-based MPC integrated with low discrepancy sampling, smooth trajectory generation, and behavior-based cost functions to produce good robotic actions reaching the goal poses. Visual Foresight [21, 23] first generated robotic planning towards a specific goal by leveraging a video prediction model to simulate candidate action sequences and then scored them based on the similarity between their predicted futures and the goal. Xu *et al.* [75] proposed a 3D volumetric scene representation that simultaneously discovers, tracks, and reconstructs objects and predicts their motion under the interactions of a robot. Ye *et al.* [76] presented an approach to learn an object-centric forward model, which planned for action sequences to achieve distant desired goals. Recently, Tian *et al.* [67] conducted a simulated benchmark for action-conditioned video prediction in the form of MPC framework that evaluated a given model for simulated robotic manipulation through sampling-based planning.

Recently, some video prediction models independent of the MPC framework have also been proposed for robotic manipulation. For instance, VLP [19] and UniPi [18] combined text-to-video models with VLM to generate long-horizon videos used for extracting control actions. V-JEPA [6] developed a latent video prediction strategy to make predictions in a learned latent space. Similarly, Dreamer [26] learned long-horizon behaviors through predicting state values and actions in a compact latent space where the latent states have a small memory footprint. RIG [50] used a latent variable model to generate goals for the robot to learn diverse behaviors. Planning to Practice [22] proposed a sub-goal generator to decompose a goal-reaching task hierarchically in the latent space.

B. Foundation Models for Robotic Manipulation

Foundation models are large-scale neural networks trained on massive and diverse datasets [9]. Breakthroughs such as GPT-4, Llama and PaLM exemplify the scaling up of LLMs [54, 11, 69, 14], showcasing notable progress in knowledge extraction and reasoning. Simultaneously, there has been an increase in the development of large-scale VLMs [2, 58, 36, 60, 17, 4]. VLMs typically employ cross-modal connectors to merge visual and textual embeddings into a unified representation space, enabling them to process multimodal data effectively.

The application of foundation models in advanced robotic systems is an emerging research field. Many studies focus on employing LLMs for knowledge reasoning and robotic manipulation [33, 79, 35, 40, 31]. To allow LLMs to perceive the physical environments, auxiliary modules such as textual descriptions of the scene [33, 79], affordance models [35], and perception APIs [40] are essential. Additionally, using VLMs for robotic manipulation is being explored [34, 17, 10]. For example, PaLM-E enhanced the understanding of robots with regard to complex visual-textual tasks [17], while RT-2 specialized in real-time image processing and decision-making [10]. However, most existing methods are limited by their reliance on pre-defined executable skills or hand-designed motion primitives [40, 34], constraining the adaptability of robots in complex, real-world environments and their interaction with diverse, unforeseen objects.

Difference from closely related works. This work is closely related to some MPC-based methods [21, 23, 67, 75] designed for robotic manipulation. However, most of these methods were designed for manipulation tasks merely involving specific objects as regular MPC has limitations in two aspects: (1) The predictive models used in regular MPC are constrained with small-scale training datasets, and thus cannot precisely predict the process of interaction with objects unseen during training; (2) The cost functions of regular MPC are usually designed with a defined set of constraints such as physical limitations or operational safety margins. Although these constraints ensure that robot actions adhere to them while striving for optimal performance, accurately modeling such constraints is highly difficult in real-world scenarios. To address the above two problems, the proposed VLMPC leverages a video prediction model which is trained with a large-scale robot manipulation dataset [55] and can be directly transferred to the real world. Also, VLMPC incorporates powerful VLMs into cost functions with high-level knowledge reasoning, which provides constraints produced through interactions with open-set objects.

III. Method

Fig. 1 illustrates the overview of the VLMPC framework. It takes as input either a goal image indicating the prospective state or a language instruction that depicts the required manipulation, and performs a dynamic strategy that iteratively makes decisions based on the predictions of future frames. First, a conditional action sampling scheme is designed to prompt VLMs to take into account both the input and the current observation and reason out prospective future movements, from which a set of candidate action sequences are sampled. Then, an action-conditioned video prediction model is devised to predict a set of future frames corresponding to the sampled action sequences. Finally, a hierarchical cost function including two sub-costs and a VLM switcher are proposed to comprehensively compute the coarse-to-fine scores for the video predictions and select the best action sequence. The first action in the sequence is fed into the robot for execution, and the subsequent actions go through a weighted elementwise summation with the conditional action distribution. We elaborate each component of VLMPC in the following.

A. Conditional Action Sampling

In an MPC framework, N candidate action sequences $\mathcal{S}_t = \{S_t^1, S_t^2, ..., S_t^N\}$ are sampled from a custom sampling distribution at each step t, where $S_t^n = \{a_{t+1}^n, a_{t+2}^n, ..., a_{t+T}^n\}$ contains T actions and $n \in \{1, ..., N\}$. For every $\tau \in \{t+1, ..., t+T\}$ representing a future step after t, $a_\tau^n \in \mathbb{R}^7$ is a 7-dimensional vector composed of the movement $[d_\tau^x, d_\tau^y, d_\tau^z]$ of the end-effector in Cartesian space, the rotation $[r_\tau^x, r_\tau^y, r_\tau^z]$ of the gripper, and a binary grasping state g_t indicating the open or close state of the end-effector.

Given a goal image G or a language instruction L as the input of VLMPC along with the current observation O_t, we expect VLMs to generate appropriate future movements, from which a sampling distribution is derived for action sampling. As shown in Fig. 2, the current observation $O_t \in \mathbb{R}^{h \times w \times 3}$ is represented as an RGB image with the shape of $h \times w \times 3$ taken by an external monocular camera. We design a prompt ϕ_s that drives VLMs to analyze O_t alongside the input. ϕ_s forces VLMs to identify and localize the object with which the robot is to interact, reason about the manner of interaction, and generate appropriate future movements. The output of VLMs can be formulated as

$$\mathrm{VLM}(O_t, G \vee L | \phi_s) = \{\widehat{d_t^x}, \widehat{d_t^y}, \widehat{d_t^z}, \widehat{r_t^x}, \widehat{r_t^y}, \widehat{r_t^z}, g_t\} \quad (1)$$

where $\widehat{\cdot} \in \{+1, 0, -1\}$ denotes the predicted moving/rotation direction alongside the corresponding axis and $g_t \in \{0, 1\}$ represents the predicted binary state of the end-effector.

To obtain a set of candidate action sequences, we follow the scheme of Visual Foresight [21] and adopt Gaussian sampling that samples N action sequences with the expected movement in each action dimension as the mean. Hence we further map the output of VLMs into a sampling mean μ_t^{VLM}:

$$\mu_t^{\mathrm{VLM}} = w_m * \{\widehat{d_t^x}, \widehat{d_t^y}, \widehat{d_t^z}\} \cup w_r * \{\widehat{r_t^x}, \widehat{r_t^y}, \widehat{r_t^z}\} \cup \{g_t\} \quad (2)$$

where w_m and w_r are hyperparameters for mapping the output of VLMs into the action space of the robot.

Hallucination phenomenon is a common issue which hinders the stable use of large-scale VLMs in real-world deployment, as it may result in unexpected consequences caused by incorrect understandings of the external environment. To mitigate the hallucination phenomenon, we propose to make use of the historical information derived from the subsequent candidate action sequence of the last step. This leads to another sampling mean μ_t^{sub}. Please refer Sec. III-C for the detailed process of obtaining μ_t^{sub}. Then we perform a weighted elementwise summation of μ_t^{sub} and μ_t^{VLM} to produce the final

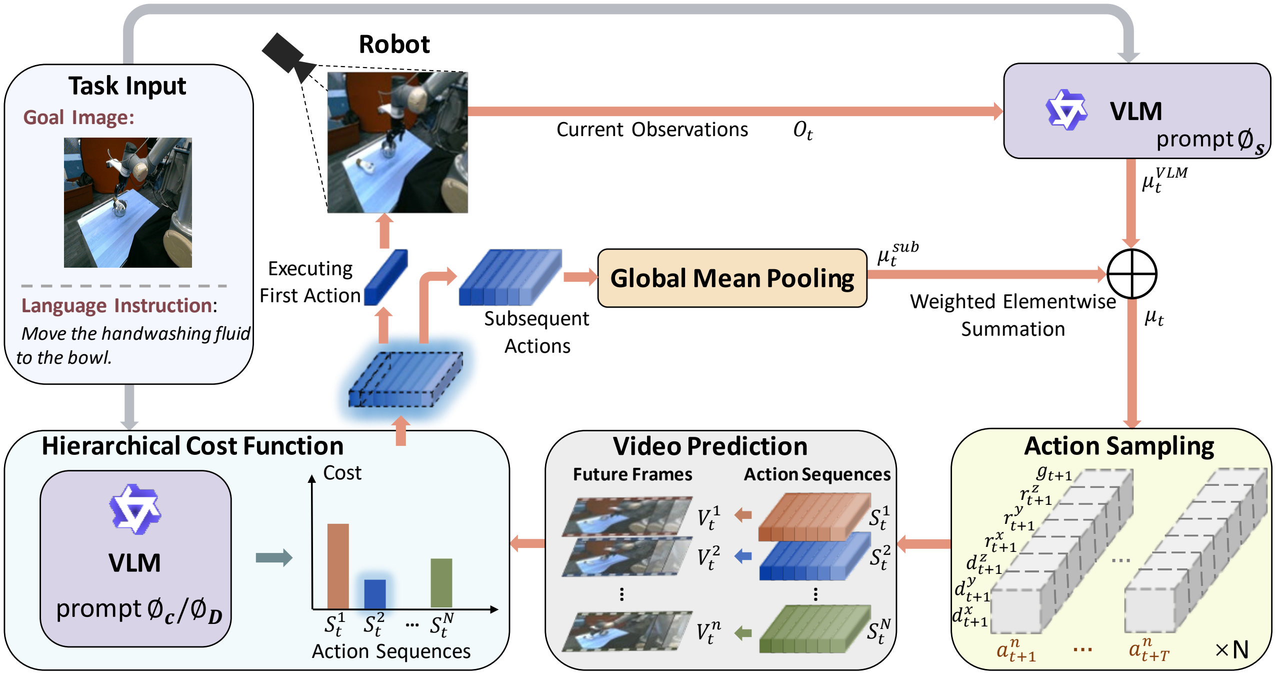

Fig. 1. VLMPC takes as input either a goal image or a language instruction. It first prompts VLMs to generate a conditional sampling distribution, from which action sequences are derived. Then, such action sequences are fed into a lightweight action-conditioned video prediction model to predict a set of future frames. The assessment of VLMPC is performed with a hierarchical cost function composed of two sub-costs: a pixel distance cost and a VLM-assisted cost for performing video assessments based on the future frames. VLMPC finally selects the best action sequence, in which the robot picks the first action to execute and the subsequent actions are fed into the action sampling module to further assist conditional action sampling.

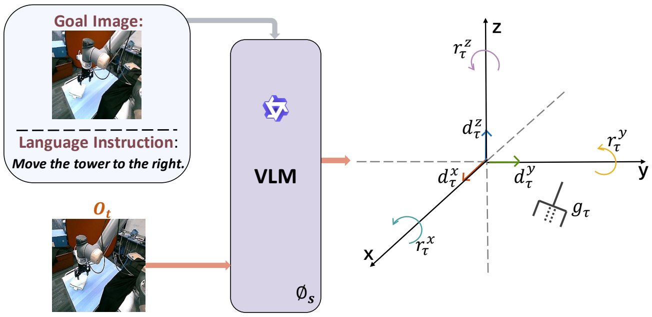

Fig. 2. The VLMs subject to a specifically designed prompt ϕ_s take as input the current observation O_t and a goal image or a language instruction to generate an end-effector moving direction at coarse level.

sampling mean μ_t of step t:

$$\mu_t = w_{\text{VLM}} * \mu_t^{\text{VLM}} + w_{\text{sub}} * \mu_t^{\text{sub}} \tag{3}$$

where w_{VLM} and w_{sub} are weighting parameters. Finally, we sample S_t from the Gaussian distribution $S_t^n \sim \mathcal{N}(\mu_t, I)$ repeatedly N times.

This conditional action sampling scheme allows VLMs to provide the guidance of robotic manipulation at a coarse level via knowledge reasoning from the image observation and the task goal. Next, with the candidate action sequences, we introduce the module for action-conditioned video prediction.

B. Action-Conditioned Video Prediction

Given the candidate action sequences, it is necessary to estimate the future state of the system when executing each sequence, which provides the forward-looking capability of VLMPC.

Traditional MPC methods often rely on hand-crafted deterministic dynamic models. Developing and refining such models typically require extensive domain knowledge, and they may not capture all relevant dynamics. On the contrary, video is rich in semantic information and thus enables the model to handle complex, dynamic environments more effectively and flexibly. Moreover, video can be directly fed into a VLM for knowledge reasoning. Thus, we use the action-conditioned video prediction model to predict the future frames corresponding to candidate action sequences.

We build a variant version of DMVFN [30], an efficient dynamic multi-scale voxel flow network for video prediction, to perform action-conditioned video prediction. We name it DMVFN-Act. Given the past two historical frames O_{t-1} and O_t, DMVFN predicts a future frame $\widehat{O}_{t+1}$, formulated as

$$\widehat{O}_{t+1} = \text{DMVFN}(O_{t-1}, O_t) \tag{4}$$

With the candidate action sequences S_t and the corresponding executed actions a_{t-1} and a_t, we expect DMVFN-Act to take the actions one by one and predict future states frame by frame as illustrated in Fig. 3. For simplicity, we explain this process by taking one sequence $S_t^n = \{a_{t+1}^n, a_{t+2}^n, ..., a_{t+T}^n\}$

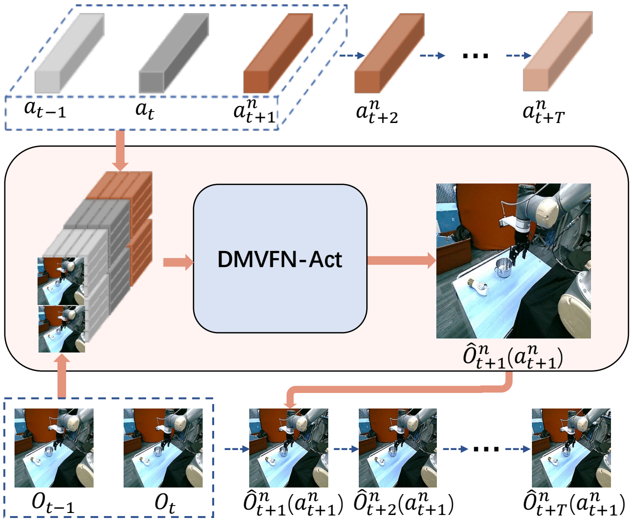
a_{t-1}
a_t
a^n_{t+1}
a^n_{t+2}
...
a^n_{t+T}
DMVFN-Act
$\hat{O}^n_{t+1}(a^n_{t+1})$
O_{t-1}
O_t
$\hat{O}^n_{t+1}(a^n_{t+1})$
$\hat{O}^n_{t+2}(a^n_{t+1})$
...
$\hat{O}^n_{t+T}(a^n_{t+1})$

Fig. 3. Given the past two frames O_t and O_{t-1} with the executed actions a_{t-1} and a_t corresponding to them and the action a^n_{t+1}, DMVFN-Act predicts the next frame $\widehat{O}^n_{t+1}(a^n_{t+1})$. The dashed boxes and arrows indicate the iterative process of taking the actions one by one and predicting the future states frame by frame.

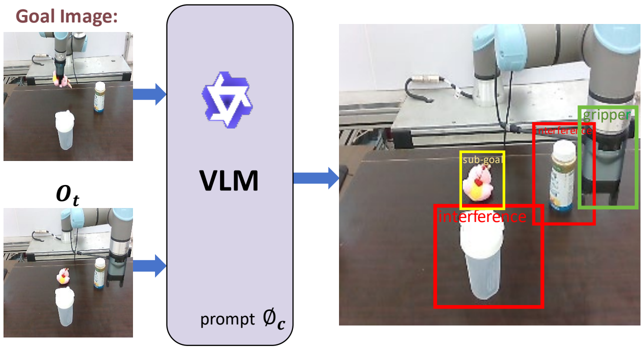

Fig. 4. Illustration of the end-effector, the next sub-goal and the interference objects in the current observation. Red, green, and yellow boxes denote the interference objects, the end-effector and the next sub-goal generated by VLMPC.

as example. We broadcast a_{t-1}, a_t, $a^n_{t+1} \in \mathbb{R}^7$ to the image size a'_{t-1}, a'_t, ${a^n_{t+1}}' \in \mathbb{R}^{h\times w\times 7}$, and then concatenate them with O_{t-1} and O_t respectively, formulated as

$$O'_{t-1} = [O_{t-1} \cdot a'_{t-1} \cdot a'_t \cdot {a^n_{t+1}}'],\\ O'_t = [O_t \cdot a'_{t-1} \cdot a'_t \cdot {a^n_{t+1}}'] \tag{5}$$

where $[\cdot]$ represents the channelwise concatenation, and O'_{t-1} and O'_t denote the action-conditioned historical observations. In DMVFN-Act, the input layer is modified to adapt O'_{t-1} and O'_t and predict one future frame $\widehat{O}^n_{t+1}(a^n_{t+1})$ conditioned by the candidate action a^n_{t+1}, expressed as

$$\widehat{O}^n_{t+1}(a^n_{t+1}) = \text{DMVFN-Act}(O'_{t-1}, O'_t) \tag{6}$$

DMVFN-Act iteratively predicts future frames via Eqs. (5) and (6) until all candidate actions are used . The action-conditioned video prediction can be represented as:

$$V^n_t = \{\widehat{O}^n_{t+1}(a^n_{t+1}), \widehat{O}^n_{t+2}(a^n_{t+2}), ..., \widehat{O}^n_{t+T}(a^n_{t+T})\} \tag{7}$$

To improve efficiency, the N candidate action sequences are organized into a batch and predict all the action-conditioned videos $V_t = \{V^1_t, V^2_t, ..., V^N_t\}$ at step t in one inference.

C. Hierarchical Cost Function

To comprehensively assess the video predictions, we design a cost function composed of two sub-costs that provide a hierarchical assessment at pixel and knowledge levels, respectively. We also propose a VLM Switcher which dynamically selects one or both sub-costs in a manner adaptive to the observation.

1) Pixel Distance Cost: While the task input is the goal image G, an intuitive way to assess video predictions is to sum the pixel distances between each future frame and the goal image. Following Visual Foresight [21], we calculate the l_2 distance between each future frame $\widehat{O}^n_\tau(a^n_\tau)$ in an action-conditioned video V^n_t and G, and then sum the distances as the pixel distance cost $C^n_P(t)$ for V^n_t over τ:

$$C^n_P(t) = \sum_{\tau=t+1}^{t+T} ||\widehat{O}^n_\tau(a^n_\tau) - G||_2 \tag{8}$$

Then, the pixel distance cost $C_P(t)$ at step t for V_t can be computed as

$$C_P(t) = \{C^n_P(t) | n \in \{1, 2, ..., N\}\} \tag{9}$$

The pixel distance cost encourages the robot to move directly towards the goal position in accordance with the goal image. This cost is simple yet effective when the task contains only one sub-goal, *e.g.*, *push a button*. However, for tasks that require manipulating objects with multiple sub-goals, where a common type is *taking an object from position A to B*, this cost usually guide the robot to move directly towards *position B* to reduce the pixel distance. To facilitate such situations, we further introduce the VLM-assisted cost.

2) VLM-Assisted Cost: Many robotic manipulation tasks contain multiple sub-goals and interference objects, which require knowledge-level task planning. For example, in the task of '*grasp the bottle and put it in the bowl, while watching out the cup*', the bottle should be identified as the sub-goal before the robot grasps it, and the bowl is the next sub-goal after the bottle is grasped, and the cup is an interference object. It is thus critical to dynamically identify the sub-goals and interference objects in each step, and make appropriate assessments on the action-conditioned video predictions so that we can select the best candidate action sequence to achieve the sub-goals as long as avoiding the interference object. Moreover, We design a VLM-assisted cost to realize it at the knowledge level.

Specifically, with the current observation O_t and the task input G or L, we design a prompt ϕ_C to drive VLMs to reason out and localize the next sub-goal and all the interference

```
Algorithm 1: VLMPC
Input: Goal image $G$ or language instruction $L$, and
       obvevation $O_t$ at every step
while task not done or $t \le T_{max}$ do
    Generates a sampling distribution by VLM
      $D(\mu^{\text{VLM}}) \leftarrow \text{VLM}(O_t, G \vee L|\phi_s)$;
    Refine it with historical information $\mu_t^{\text{sub}}$
      $D(\mu_t) = D(w_{\text{VLM}} * \mu_t^{\text{VLM}} + w_{\text{sub}} * \mu_t^{\text{sub}})$;
    $\mathcal{S}_t \leftarrow$ sample $N$ action sequences;
    foreach sequence $S_t^n \in \mathcal{S}_t$ do
        for future step $\tau = t+1, ..., t+T$ do
            $\widehat{O}_\tau^n(a_\tau^n) \leftarrow$ predict the future frame;
        end
        $V_t^n = \{\widehat{O}_\tau^n(a_\tau^n) | \tau \in \{t+1, ..., t+T\}$ ;
    end
    $C_P(t) \leftarrow$ calculate the pixel distance cost;
    $C_{\text{VLM}} \leftarrow$ calculate the VLM-assisted cost;
    $C_t \leftarrow$ arrange cost through VLM swicher;
    $S_t^{n^\star} \leftarrow$ select the optimal action sequence;
    Execute the first action $a_{t+1}^{n^\star}$ in the optimal
      sequence;
    Update $\mu_{t+1}^{sub}$ using $\{a_\tau^{n^\star} | \tau \in \{t+2, ..., t+T\}\}$;
end
```

objects, where the sub-goal is usually the next object to interact with the robot. As illustrated in Fig. 4, this process yields the bounding boxes of the robot's end-effector e_t, the next sub-goal s_t and all the interference objects I_t in the current observation:

$$\text{VLM}(O_t, G \vee L|\phi_C) = \{e_t, s_t, I_t\} \tag{10}$$

Since the predicted videos V_t share the historical frame O_t, a lightweight visual tracker VT can be used to localize both the end-effector e_τ^n and the sub-goal s_τ^n in each future frame in all the action-conditioned predicted videos initialized with e_t, s_t, and I_t, formulated as:

$$\begin{aligned} \text{VT}(V_t|e_t, s_t, I_t) = \{e_\tau^n, s_\tau^n, I_\tau^n | n \in \{1, 2, ..., N\}, \\ \tau \in \{t+1, t+2, ..., t+T\}\} \end{aligned} \tag{11}$$

where we employ an efficient real-time tracking network SiamRPN [39] as the visual tracker in this work.

To encourage the robot to move towards the next sub-goal and avoid colliding with all the interference objects, we calculate the VLM-assisted cost C_{VLM}^n as:

$$C_{\text{VLM}}^n(t) = \sum_{\tau=t+1}^{t+T} (||c(e_\tau^n) - c(s_\tau^n)||_2 - ||c(e_\tau^n) - c(I_\tau^n)||_2), \tag{12}$$

$$C_{\text{VLM}}(t) = \{C_{\text{VLM}}^n(t) | n \in \{1, 2, ..., N\}\} \tag{13}$$

where $c(\cdot)$ represents the center of the given bounding boxes.

3) VLM Switcher: The pixel distance cost can provide fine-grained guidance on the pixel level, and VLM-assisted cost fixes the gap in knowledge-level task planning. With these two sub-costs, we further design a VLM switcher with prompt ϕ_D, which dynamically selects one or both appropriate sub-costs in each step t adaptive to the current observation through knowledge reasoning to produce the final cost $C(t)$:

$$\text{VLM}(O_t, G \vee L|\phi_D) = w_D \in \{0, 0.5, 1\}, \tag{14}$$

$$C(t) = w_D * C_P(t) + (1 - w_D) * C_{\text{VLM}}(t) \tag{15}$$

With the cost $C(t) = \{C^n(t) | n \in \{1, 2, ..., N\}\}$ as the assessment of all the action-conditioned videos, we select the candidate action sequence with the lowest cost for the following process. When the first action in this sequence is executed, the subsequent actions are fed into a global mean pooling layer to generate the sampling mean μ_t^{sub} to provide historical information in the action sampling of the next step.

Algorithm 1 summarizes the whole process of the VLMPC framework. When the task is done or reaching the maximum time limit, the system will return an end signal.

IV. Experiments

In this section, we first provide the implementation details of the proposed VLMPC framework. Then, we conduct two comparative experiments in simulated environments. The first is to compare VLMPC with VP^2 [67] on 7 tasks in the RoboDesk environment [37]. The second is to compare VLMPC with 5 existing methods in 50 simulated environments provided by the Language Table environment [47]. Next, we evaluate VLMPC in real-world scenarios. Finally, we investigate the effectiveness of each core component of VLMPC through ablation study. We provide the details of all the hyperparameters and the VLM prompts in the supplementary material.

A. Implementation Details

VLMPC employs Qwen-VL [4] and GPT-4V [1] as the VLMs. In the conditional action sampling module, VLMPC first uses GPT-4V to identify the target object with which the robot needs to interact, and then localizes the object through Qwen-VL. In the VLM-assisted cost, VLMPC first extracts the sub-goals and interference objects with GPT-4V, and then localizes them through Qwen-VL. The VLM switcher uses GPT-4V to dynamically select one or both sub-costs in each time step.

The training policy of the DMVFN-Act video prediction model contains 2 stages. In the first stage, we select 3 sub-datasets from the Open X-Embodiment Dataset [55], a large-scale dataset containing more than 1 million robot trajectories collected from 22 robot embodiments. The 3 sub-datasets used for pre-training DMVFN-Act are Berkeley Autolab UR5, Columbia PushT Dataset, and ASU TableTop Manipulation. In the second stage, we collect 20 episodes of robot execution in the environment where the experiments are conducted and train DMVFN-Act to adapt to the specific scenario.

B. Simulation Experiments

1) Simulation Environments and Experiment Settings: The first evaluation is conducted on the popular simulation benchmark VP^2 [67] which offers two environments RoboDesk [37]

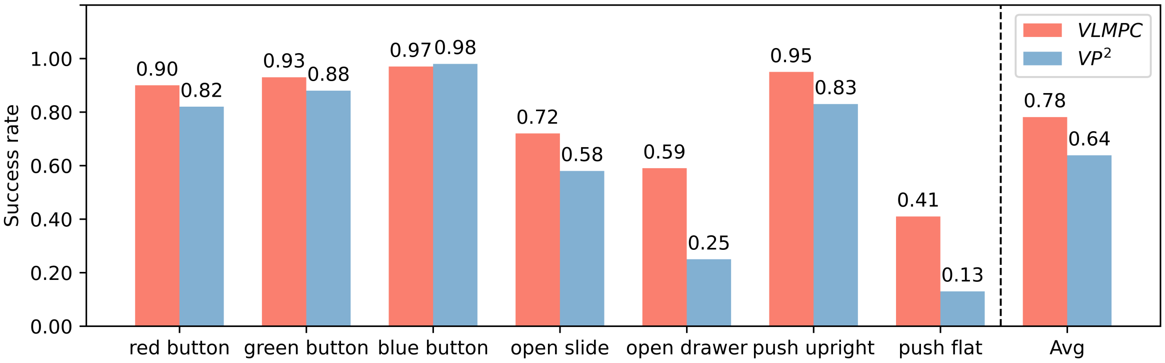

Fig. 5. Quantitative comparison with the VP^2 baseline in the RoboDesk environment.

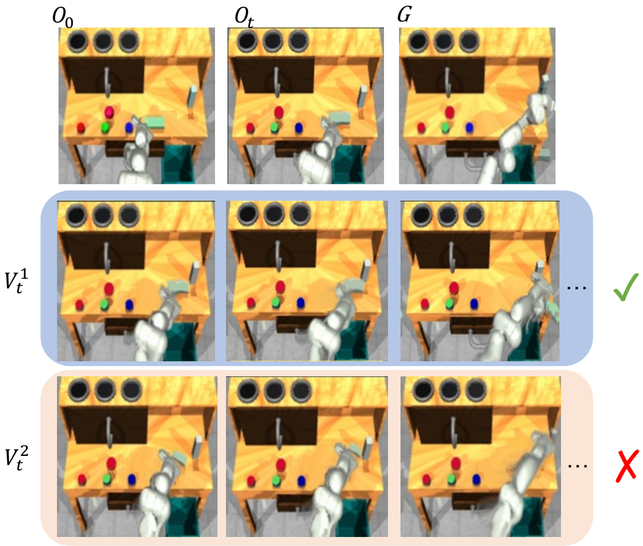

Fig. 6. Visualization of the action-conditioned video predictions assessed by the hierarchical cost function of VLMPC via knowledge reasoning.

and `robosuite` [81]. Considering the significant difference between the physical rendering of `robosuite` and real-world scenarios, we only use RoboDesk in this work. RoboDesk provides a physical environment with a Franka Panda robot arm, as well as a set of manipulation tasks. VP^2 conducts 7 sub-tasks: *push* $\{$*red, green, blue*$\}$ *button*, *open* $\{$*slide, drawer*$\}$, *push* $\{$*upright block, flat block*$\}$ *off table*. For each sub-task, VP^2 provides 30 goal images as task input.

In the second experiment, we compare VLMPC with 5 existing methods in the Language Table environment [47] on the *move to area* task following VLP [19]. Such a task is given by language instructions: *move all blocks to different areas of the board*. The 5 competing methods are UniPi [18], LAVA [41], PALM-E [17], RT-2 [10] and VLP [19]. We follow VLP [19] to compute rewards using the ground truth state of each block in the Language Table environment [47]. And we made the evaluation on 50 randomly initialized environments.

TABLE I
COMPARISON WITH EXISTING METHODS ON THE TASK OF *move to area* IN THE LANGUAGE TABLE ENVIRONMENT.

Method	Success Rate(%)	Reward
UniPi [18]	0	30.8
LAVA [41]	22	59.8
PALM-E [17]	0	36.5
RT-2 [10]	0	18.5
VLP [19]	64	87.3
VLMPC	**70**	**89.3**

2) Experimental Results: The experimental results on the VP^2 benchmark are listed in Fig. 5. It can be seen that VLMPC significantly outperforms the VP^2 baseline. We can see that for the tasks of *push* $\{$*red, green, blue*$\}$ *button*, both the VP^2 baseline and VLMPC achieve high performance. This is simply because such tasks contain no multiple sub-goals. Thus, once the robot arm reaches the specific button and pushes it, the task is completed. On the other hand, the remaining tasks are more challenging, which require the robot to identify and move among multiple sub-goals as well as avoiding the collision with interference objects. We can see that VLMPC significantly outperforms the VP^2 baseline in such challenging tasks, demonstrating its good reasoning and planning capability.

Fig. 6 shows the visual results for the most challenging sub-task *push flat*. This task requires pushing a flat green block off the table, while keeping other objects unmoved. We notice a slender block standing on the right edge of the table, which obviously serves an interference object. For the current observation O_t, we select two predicted videos for visualization. The second and the third rows of Fig. 6 show the predicted videos corresponding to different candidate action sequences. It can be seen that both candidate action sequences have the tendency to push the flat block off the table. It is noteworthy that the VP^2 baseline using a pixel-level cost and a simple state classifier assigns similar costs on both

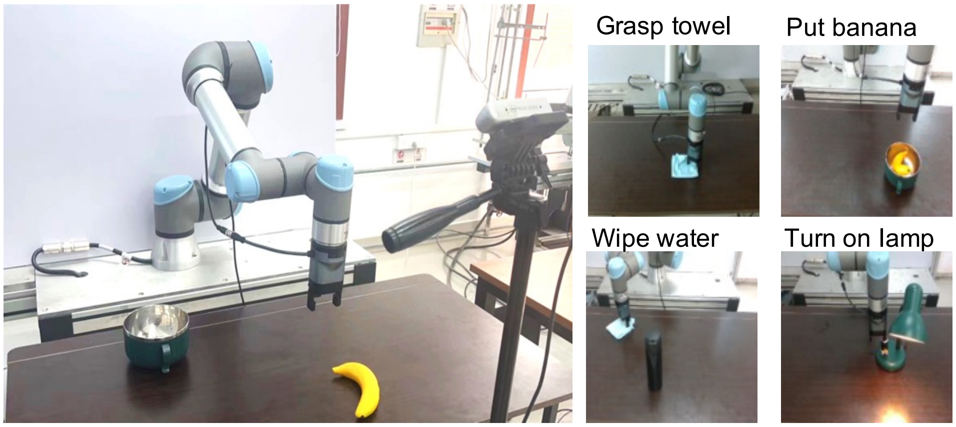

Fig. 7. The real-world experimental platform includes a UR5 robot arm and a monocular RGB camera. It also shows a goal image for each of the four tasks.

videos, which leads to the selection of an inappropriate action sequence. In contrast, VLMPC produces a higher cost for V_t^2 which contains a possible collision between the robot arm and the interference object. V_t^1 indicates a more reasonable moving direction and interaction with objects, and is thus assigned a lower cost. Such results demonstrate that the proposed hierarchical cost function can make desired assessment of the predicted videos on the knowledge level and facilitate VLMPC to select an appropriate action to execute.

Table I lists the quantitative results of the comparative experiment conducted in the Language Table environment [47], where the Reward metric is computed in accordance with the VLP reward [19]. It can be seen that the proposed VLMPC outperforms all competing methods. This is because VLMs are good at localizing specific areas. Therefore, through sampling actions towards the sub-goals, VLMPC enables the robot to successfully reach the sub-goals and complete the task.

C. Real-World Experiments

1) Experimental Setting: As shown in Fig. 7, we use a UR5 robot to conduct real-world experiments. A monocular RGB camera is set up in front of the manipulation platform to provide the observations. We design four manipulation tasks, including *grasp towel*, *put banana*, *turn on lamp*, and *wipe water*. It is noteworthy that the objects involved in these tasks are not included in the collected data for training the video prediction model.

In each manipulation task, the position of the objects is initialized randomly within the reachable space of the action, yielding different goal images. Fig. 7 shows some example goal images for the 4 tasks.

2) Experimental Results: To properly evaluate VLMPC in real-world tasks, we repeat each task 30 times by randomly initializing the position of all objects and changes the color of the tablecloth every 10 times. We calculate the success rate and the average time for each task respectively. The results are listed in Table II. It can be seen that VLMPC achieves high success rates on the tasks of *grasp towel* and *turn on lamp*. These two tasks are relatively simple as there is no interference object in the scene. The success rates for the tasks of *put banana* and *wipe water* are low because they are more challenging. *put banana* contains multiple sub-goals, and *wipe water* is even more difficult as it involves both interference objects and multiple sub-goals. Such results demonstrate that VLMPC generalizes well to novel objects and scenes unseen in the training dataset.

TABLE II
RESULTS OF VLMPC USING GOAL IMAGE OR LANGUAGE INSTRUCTION AS INPUT IN REAL-WORLD EXPERIMENTS.

Tasks	Goal Image		Language Instruction	
	Success Rate(%)	Time(s)	Success Rate(%)	Time(s)
grasp towel	76.67	162.4	73.33	184.6
put banana	60.00	203.9	46.67	230.7
turn on lamp	83.33	128.4	86.67	142.8
wipe water	36.67	289.3	23.33	331.9

We also provide the visual results for two challenging tasks *put banana in the bowl* and *wipe water*. As shown in Fig. 8, in the *put banana in the bowl* task, VLMPC correctly identifies the first sub-goal, *i.e.*, the banana, based on the current observation, and drives the robot arm moving towards and finally grasping it. Then, VLMPC dynamically finds the next sub-goal, *i.e.*, the bowl, and subsequently guides the robot to move to the area above it and opens the gripper. This example demonstrates VLMPC has the capability of dynamically identifying the sub-goals during the task.

The *wipe water* task requires the robot arm to wipe off the water on the table with the towel while watching out the bottle. It is clear that this task contains two sub-goals *towel* and *water*, and an interference object *bottle*. Fig. 8 shows that our method successfully identifies all of them, and guides the robot to select appropriate actions to execute while avoiding the collision with the interference object.

We provide more visualization results on four sub-tasks with both successful and failure cases, as well as related discussion in the supplementary material. We also provide video demon-

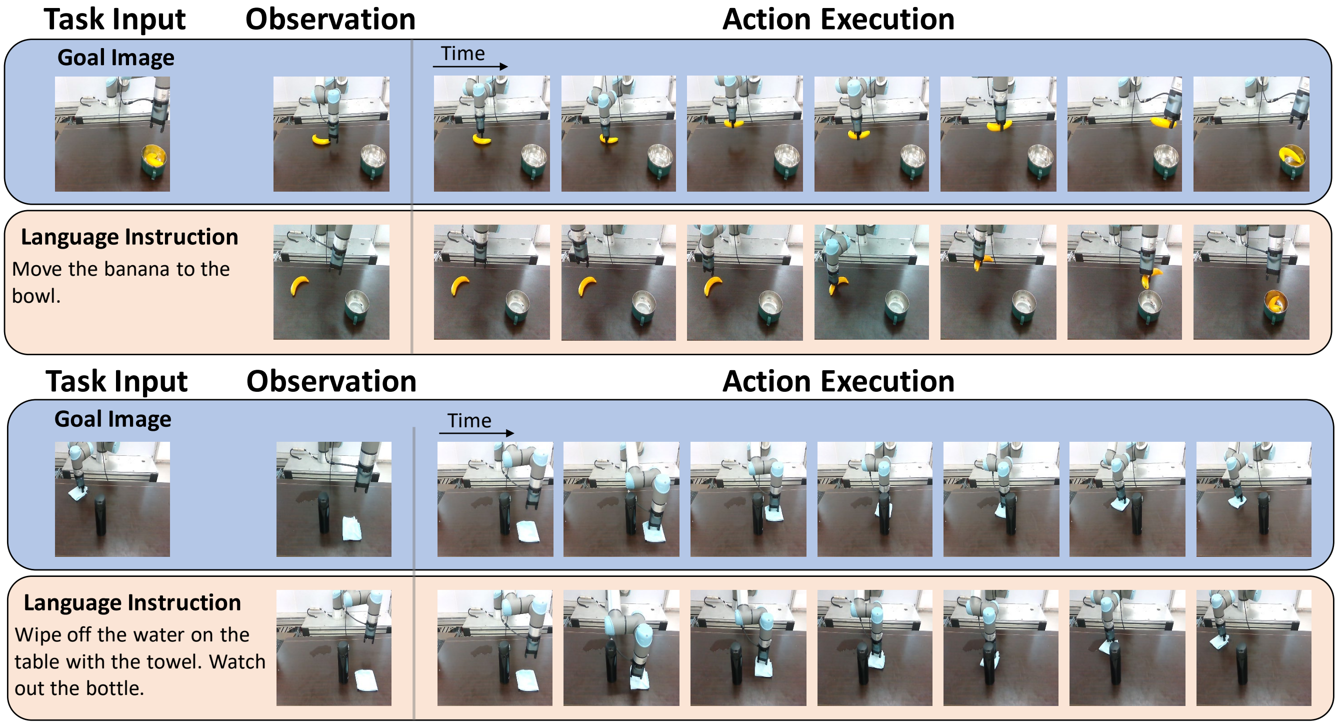

Fig. 8. Visualization of the action execution for two challenging real-world manipulation tasks *put the banana in the bowl* and *wipe water*.

strations in both simulated and real-world environments.

D. Ablation studies

We have conducted ablation study to demonstrate the effectiveness of each core component of VLMPC. In the experiments, we compare VLMPC with 4 variants described as follows:

VLMPC-rs: This is an ablated version of VLMPC where the conditional action sampling module is replaced with random sampling which simply sets the sampling mean μ_t to zero.

VLMPC-PD: This variant of VLMPC only uses the pixel distance cost as the cost function.

VLMPC-VS: This variant of VLMPC only uses the VLM-assisted cost as the cost function.

VLMPC-MCVD: In this variant of VLMPC, we replace DMVFN-Act with the action-conditioned video prediction model MCVD [67, 70].

The results are shown in Table III. First, compared with random sampling, our conditional action sampling module makes the robot complete various tasks more quickly and achieve higher success rates. This is because random sampling cannot make the sampled action sequences focus on the direction to sub-goals. Second, when VLMPC only uses the pixel distance cost, we found that the robot directly moves to the goal position and ignores intermediate sub-goals, leading to low success rates in the tasks *put banana* and *wipe water*. Besides, when VLMPC only uses the VLM-assisted cost, we found that VLM sometimes localizes incorrect sub-goals, which also leads to low success rates. Third, compared with DMVFN-Act, the diffusion-based video prediction model MCVD leads to much lower efficiency in all testing tasks.

V. Conclusion

This paper introduces VLMPC that integrates VLM with MPC for robotic manipulation. It prompts VLM to produce a set of candidate action sequences conditioned on the knowledge reasoning of goal and observation, and then follows the MPC paradigm to select the optimal one from them. The hierarchical cost function based on VLM is also designed to provide an amenable assessment for the actions by estimating the future frames generated by a lightweight action-conditioned video prediction model. Experimental results demonstrate that VLMPC performs well in both simulated and real-world scenarios.

A limitation of VLMPC lies in its process of video prediction, where a mismatch between the predicted video and the action sequence may occur and thereby affect the evaluation of the action sequences. Moreover, incorporating a large-scale VLM into each step of the MPC loop introduces higher computing cost inevitably. Without assessing the predicted video at each step, VLMPC cannot perfectly handle the tasks where the space of motion is strictly constrained. Hence, developing a more reliable video prediction model and designing a more efficient scheme for integrating VLM with MPC are of interest in the future work.

TABLE III
Ablation study using the variants of VLMPC on different tasks in real-world environments.

VLMPC Variant	*grasp towel*		*put banana*		*turn on lamp*		*wipe water*	
	Success Rate(%)	Time(s)	Success Rate(%)	Time(s)	Success Rate(%)	Time(s)	Success Rate(%)	Time(s)
VLMPC-rs	63.33	302.5	40	389.5	73.33	256.7	13.33	573.9
VLMPC-PD	26.67	178.3	0	-	60.00	**123.6**	0	-
VLMPC-VS	56.67	201.5	46.67	297.3	56.67	243.7	10.00	543.9
VLMPC-MCVD	33.33	509.3	23.33	689.4	46.67	553.8	6.67	803.5
VLMPC	**76.67**	**162.4**	**60.00**	**203.9**	**83.33**	128.4	**36.67**	**289.3**

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 61991411, U22A2057, and 62076148, in part by the National Science and Technology Major Project under Grant 2021ZD0112002, in part by the Shandong Excellent Young Scientists Fund Program (Overseas) under Grant 2022HWYQ-042, in part by the Young Taishan Scholars Program of Shandong Province No.tsqn201909029, and in part by Project for Self-Developed Innovation Team of Jinan City under Grant 2021GXRC038.

References

[1] Gpt-4v(ision) system card. 2023.

[2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35: 23716–23736, 2022.

[3] Guillaume Allibert, Estelle Courtial, and François Chaumette. Predictive control for constrained image-based visual servoing. *IEEE Transactions on Robotics*, 26(5):933–939, 2010.

[4] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 2023.

[5] Yutong Bai, Xinyang Geng, Karttikeya Mangalam, Amir Bar, Alan Yuille, Trevor Darrell, Jitendra Malik, and Alexei A Efros. Sequential modeling enables scalable learning for large vision models. *arXiv preprint arXiv:2312.00785*, 2023.

[6] Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. V-jepa: Latent video prediction for visual representation learning. 2023.

[7] Mohak Bhardwaj, Balakumar Sundaralingam, Arsalan Mousavian, Nathan D Ratliff, Dieter Fox, Fabio Ramos, and Byron Boots. Storm: An integrated framework for fast joint-space model-predictive control for reactive manipulation. In *Conference on Robot Learning*, pages 750–759. PMLR, 2022.

[8] Kaifeng Bi, Lingxi Xie, Hengheng Zhang, Xin Chen, Xiaotao Gu, and Qi Tian. Accurate medium-range global weather forecasting with 3d neural networks. *Nature*, 619 (7970):533–538, 2023.

[9] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[10] 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.

[11] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33: 1877–1901, 2020.

[12] 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 *IEEE International Conference on Robotics and Automation*, pages 11509–11522, 2023.

[13] Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *arXiv preprint arXiv:2310.09478*, 2023.

[14] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24 (240):1–113, 2023.

[15] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in Neural Information Processing Systems*, 36, 2024.

[16] Yan Ding, Xiaohan Zhang, Chris Paxton, and Shiqi

Zhang. Task and motion planning with large language models for object rearrangement. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2086–2092, 2023.

[17] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

[18] Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *Advances in Neural Information Processing Systems*, 36, 2024.

[19] Yilun Du, Sherry Yang, Pete Florence, Fei Xia, Ayzaan Wahid, brian ichter, Pierre Sermanet, Tianhe Yu, Pieter Abbeel, Joshua B. Tenenbaum, Leslie Pack Kaelbling, Andy Zeng, and Jonathan Tompson. Video language planning. In *International Conference on Learning Representations*, 2024.

[20] Frederik Ebert, Sudeep Dasari, Alex X Lee, Sergey Levine, and Chelsea Finn. Robustness via retrying: Closed-loop robotic manipulation with self-supervised learning. In *Conference on Robot Learning*, pages 983–993, 2018.

[21] Frederik Ebert, Chelsea Finn, Sudeep Dasari, Annie Xie, Alex Lee, and Sergey Levine. Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. *arXiv preprint arXiv:1812.00568*, 2018.

[22] Kuan Fang, Patrick Yin, Ashvin Nair, and Sergey Levine. Planning to practice: Efficient online fine-tuning by composing goals in latent space. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4076–4083, 2022.

[23] Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *IEEE International Conference on Robotics and Automation*, pages 2786–2793, 2017.

[24] Ruben Grandia, Farbod Farshidian, René Ranftl, and Marco Hutter. Feedback mpc for torque-controlled legged robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4730–4737, 2019.

[25] 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, 2023.

[26] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020.

[27] Lukas Hewing, Kim P Wabersich, Marcel Menner, and Melanie N Zeilinger. Learning-based model predictive control: Toward safe learning in control. *Annual Review of Control, Robotics, and Autonomous Systems*, 3:269–296, 2020.

[28] Noriaki Hirose, Fei Xia, Roberto Martín-Martín, Amir Sadeghian, and Silvio Savarese. Deep visual mpc-policy learning for navigation. *IEEE Robotics and Automation Letters*, 4(4):3184–3191, 2019.

[29] Thomas M Howard, Colin J Green, and Alonzo Kelly. Receding horizon model-predictive control for mobile robot navigation of intricate paths. In *International Conference on Field and Service Robotics*, pages 69–78, 2010.

[30] Xiaotao Hu, Zhewei Huang, Ailin Huang, Jun Xu, and Shuchang Zhou. A dynamic multi-scale voxel flow network for video prediction. In *Conference on Computer Vision and Pattern Recognition*, pages 6121–6131, 2023.

[31] 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.

[32] 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, 2022.

[33] Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, Pierre Sermanet, Tomas Jackson, Noah Brown, Linda Luu, Sergey Levine, Karol Hausman, and brian ichter. Inner monologue: Embodied reasoning through planning with language models. In *Conference on Robot Learning*, 2022.

[34] 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. In *Conference on Robot Learning*, 2023.

[35] Wenlong Huang, Fei Xia, Dhruv Shah, Danny Driess, Andy Zeng, Yao Lu, Pete Florence, Igor Mordatch, Sergey Levine, Karol Hausman, et al. Grounded decoding: Guiding text generation with grounded models for embodied agents. *Advances in Neural Information Processing Systems*, 36, 2024.

[36] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916, 2021.

[37] Harini Kannan, Danijar Hafner, Chelsea Finn, and Dumitru Erhan. Robodesk: A multi-task reinforcement learning benchmark. https://github.com/google-research/robodesk, 2021.

[38] Ian Lenz, Ross A Knepper, and Ashutosh Saxena. Deepmpc: Learning deep latent features for model predictive control. In *Robotics: Science and Systems*, volume 10, page 25. Rome, Italy, 2015.

[39] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *Conference on Computer Vision*

and Pattern Recognition, pages 8971–8980, 2018.

[40] 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. In *IEEE International Conference on Robotics and Automation*, pages 9493–9500, 2023.

[41] Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection, 2023.

[42] 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.

[43] Bo Liu, Yuqian Jiang, Xiaohan Zhang, Qiang Liu, Shiqi Zhang, Joydeep Biswas, and Peter Stone. Llm+ p: Empowering large language models with optimal planning proficiency. *arXiv preprint arXiv:2304.11477*, 2023.

[44] Fuxiao Liu, Kevin Lin, Linjie Li, Jianfeng Wang, Yaser Yacoob, and Lijuan Wang. Aligning large multi-modal model with robust instruction tuning. *arXiv preprint arXiv:2306.14565*, 2023.

[45] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in Neural Information Processing Systems*, 36, 2024.

[46] Yujie Lu, Pan Lu, Zhiyu Chen, Wanrong Zhu, Xin Eric Wang, and William Yang Wang. Multimodal procedural planning via dual text-image prompting. *arXiv preprint arXiv:2305.01795*, 2023.

[47] Corey Lynch, Ayzaan Wahid, Jonathan Tompson, Tianli Ding, James Betker, Robert Baruch, Travis Armstrong, and Pete Florence. Interactive language: Talking to robots in real time. *IEEE Robotics and Automation Letters*, 2023.

[48] Zhao Mandi, Shreeya Jain, and Shuran Song. Roco: Dialectic multi-robot collaboration with large language models, 2023.

[49] Michael Moor, Oishi Banerjee, Zahra Shakeri Hossein Abad, Harlan M Krumholz, Jure Leskovec, Eric J Topol, and Pranav Rajpurkar. Foundation models for generalist medical artificial intelligence. *Nature*, 616(7956):259–265, 2023.

[50] Ashvin V Nair, Vitchyr Pong, Murtaza Dalal, Shikhar Bahl, Steven Lin, and Sergey Levine. Visual reinforcement learning with imagined goals. *Advances in Neural Information Processing Systems*, 31, 2018.

[51] Suraj Nair, Eric Mitchell, Kevin Chen, Silvio Savarese, Chelsea Finn, et al. Learning language-conditioned robot behavior from offline data and crowd-sourced annotation. In *Conference on Robot Learning*, pages 1303–1315, 2022.

[52] Zhe Ni, Xiao-Xin Deng, Cong Tai, Xin-Yue Zhu, Xiang Wu, Yong-Jin Liu, and Long Zeng. Grid: Scene-graph-based instruction-driven robotic task planning. *arXiv preprint arXiv:2309.07726*, 2023.

[53] Julian Nubert, Johannes Köhler, Vincent Berenz, Frank Allgöwer, and Sebastian Trimpe. Safe and fast tracking on a robot manipulator: Robust mpc and neural network control. *IEEE Robotics and Automation Letters*, 5(2): 3050–3057, 2020.

[54] OpenAI. Gpt-4 technical report, 2023.

[55] 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.

[56] Vishal Pallagani, Bharath Chandra Muppasani, Kaushik Roy, Francesco Fabiano, Andrea Loreggia, Keerthiram Murugesan, Biplav Srivastava, Francesca Rossi, Lior Horesh, and Amit P. Sheth. On the prospects of incorporating large language models (LLMs) in automated planning and scheduling (APS). In *International Conference on Automated Planning and Scheduling*, 2024.

[57] Jianing Qiu, Lin Li, Jiankai Sun, Jiachuan Peng, Peilun Shi, Ruiyang Zhang, Yinzhao Dong, Kyle Lam, Frank P-W Lo, Bo Xiao, et al. Large ai models in health informatics: Applications, challenges, and the future. *IEEE Journal of Biomedical and Health Informatics*, 2023.

[58] 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, 2021.

[59] 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.

[60] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

[61] Allen Z. Ren, Anushri Dixit, Alexandra Bodrova, Sumeet Singh, Stephen Tu, Noah Brown, Peng Xu, Leila Takayama, Fei Xia, Jake Varley, Zhenjia Xu, Dorsa Sadigh, Andy Zeng, and Anirudha Majumdar. Robots that ask for help: Uncertainty alignment for large language model planners. In *Conference on Robot Learning*, 2023.

[62] Hao Sha, Yao Mu, Yuxuan Jiang, Li Chen, Chenfeng Xu, Ping Luo, Shengbo Eben Li, Masayoshi Tomizuka, Wei Zhan, and Mingyu Ding. Languagempc: Large language models as decision makers for autonomous driving. *arXiv preprint arXiv:2310.03026*, 2023.

[63] David Hyunchul Shim, H Jin Kim, and Shankar Sastry. Decentralized nonlinear model predictive control of multiple flying robots. In *IEEE International Conference on Decision and Control*, volume 4, pages 3621–3626, 2003.

[64] 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 *IEEE International Conference on Robotics and Automation*, pages 11523–11530, 2023.

[65] 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 *International Conference on Computer Vision*, pages 2998–3009, 2023.

[66] Arun James Thirunavukarasu, Darren Shu Jeng Ting, Kabilan Elangovan, Laura Gutierrez, Ting Fang Tan, and Daniel Shu Wei Ting. Large language models in medicine. *Nature medicine*, 29(8):1930–1940, 2023.

[67] Stephen Tian, Chelsea Finn, and Jiajun Wu. A control-centric benchmark for video prediction. In *International Conference on Learning Representations*, 2022.

[68] Guillem Torrente, Elia Kaufmann, Philipp Föhn, and Davide Scaramuzza. Data-driven mpc for quadrotors. *IEEE Robotics and Automation Letters*, 6(2):3769–3776, 2021.

[69] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[70] Vikram Voleti, Alexia Jolicoeur-Martineau, and Christopher Pal. MCVD - masked conditional video diffusion for prediction, generation, and interpolation. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[71] 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.

[72] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

[73] Grady Williams, Nolan Wagener, Brian Goldfain, Paul Drews, James M Rehg, Byron Boots, and Evangelos A Theodorou. Information theoretic mpc for model-based reinforcement learning. In *IEEE International Conference on Robotics and Automation*, pages 1714–1721, 2017.

[74] Yaqi Xie, Chen Yu, Tongyao Zhu, Jinbin Bai, Ze Gong, and Harold Soh. Translating natural language to planning goals with large-language models. *arXiv preprint arXiv:2302.05128*, 2023.

[75] Zhenjia Xu, Zhanpeng He, Jiajun Wu, and Shuran Song. Learning 3d dynamic scene representations for robot manipulation. *arXiv preprint arXiv:2011.01968*, 2020.

[76] Yufei Ye, Dhiraj Gandhi, Abhinav Gupta, and Shubham Tulsiani. Object-centric forward modeling for model predictive control. In *Conference on Robot Learning*, pages 100–109. PMLR, 2020.

[77] Wenhao Yu, Nimrod Gileadi, Chuyuan Fu, Sean Kirmani, Kuang-Huei Lee, Montserrat Gonzalez Arenas, Hao-Tien Lewis Chiang, et al. Language to rewards for robotic skill synthesis. In *Conference on Robot Learning*, 2023.

[78] Haoqi Yuan, Chi Zhang, Hongcheng Wang, Feiyang Xie, Penglin Cai, Hao Dong, and Zongqing Lu. Plan4mc: Skill reinforcement learning and planning for open-world minecraft tasks. *arXiv preprint arXiv:2303.16563*, 2023.

[79] Andy Zeng, Maria Attarian, brian ichter, Krzysztof Marcin Choromanski, Adrian Wong, Stefan Welker, Federico Tombari, Aveek Purohit, Michael S Ryoo, Vikas Sindhwani, Johnny Lee, Vincent Vanhoucke, and Pete Florence. Socratic models: Composing zero-shot multimodal reasoning with language. In *International Conference on Learning Representations*, 2023.

[80] Yukun Zhou, Mark A Chia, Siegfried K Wagner, Murat S Ayhan, Dominic J Williamson, Robbert R Struyven, Timing Liu, Moucheng Xu, Mateo G Lozano, Peter Woodward-Court, et al. A foundation model for generalizable disease detection from retinal images. *Nature*, 622(7981):156–163, 2023.

[81] Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Soroush Nasiriany, and Yifeng Zhu. robosuite: A modular simulation framework and benchmark for robot learning. *arXiv preprint arXiv:2009.12293*, 2020.