Prompting with the Future: Open-World Model Predictive Control with Interactive Digital Twins

Chuanruo Ning Kuan Fang* Wei-Chiu Ma*
Cornell University

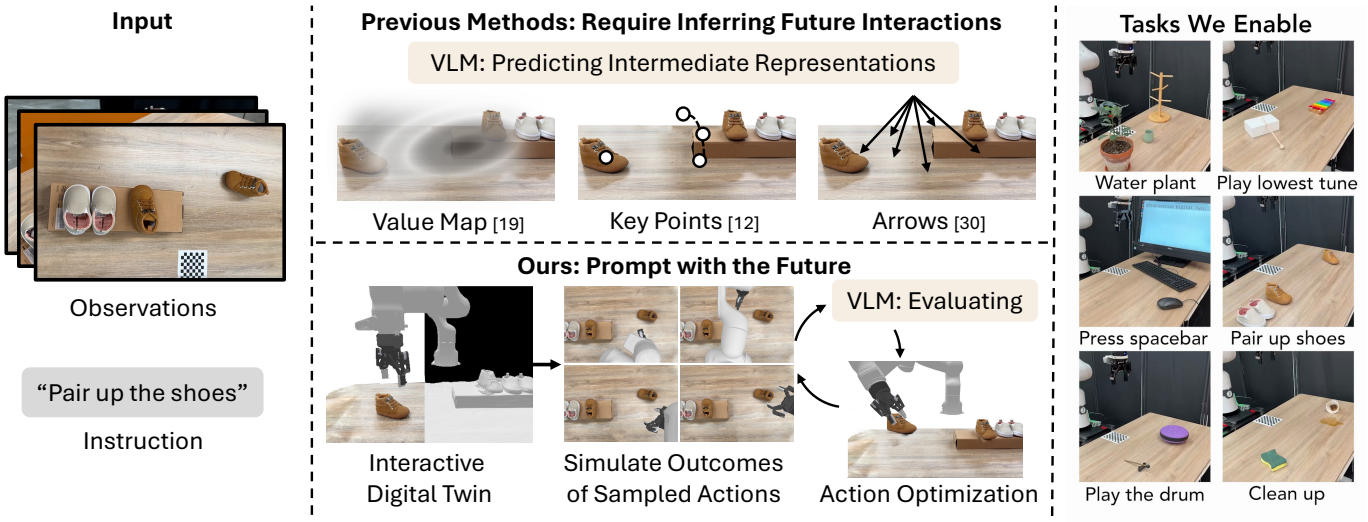

Fig. 1: Unlike previous methods that predict actions or intermediate representations given the instruction and current observations, which requires VLMs to implicitly imagine the low-level interactions, we build interactive digital twins to directly provide the future states of different actions to the model and let VLM focus on result evaluation, unlocking the potential of open-world motion planning on diverse tasks. Please use *Adobe Acrobat Reader* to view embedded video demonstrations.

***Abstract*—Open-world robotic manipulation requires robots to perform novel tasks specified by natural language instructions in unstructured environments. Although vision-language models (VLMs) with emergent high-level semantic reasoning capabilities have provided promising tools for this problem, they often lack the nuanced physics understanding critical for precise low-level control. To address this gap, we present Twin-Assisted Reasoning for Planning (TARP), a model predictive control framework that decouples semantic and physical reasoning for open-world robotic manipulation. TARP builds an interactive digital twin of the real-world environment from video scans, enabling prediction of future observations under candidate actions. Rather than relying on VLMs to infer dynamics, our method simulates potential outcomes and renders them as visual prompts. A sampling-based planning procedure then selects the optimal actions. By adaptively adjusting camera viewpoints and rendering parameters, our approach generates visual prompts that effectively convey physical context to the VLM. We evaluate TARP on eight real-world manipulation tasks involving intricate contact, object reorientation, and clutter resolution, demonstrating significantly higher success rates than state-of-the-art VLM-based control methods. Through thorough ablation studies, we validate the key design choices that drive TARP 's robust performance, underscoring the importance of explicitly modeling physics while leveraging the strengths of modern vision-language models.**

* Equal advisory

I. Introduction

Open-world robotic manipulation presents a significant challenge: generalist robots must solve novel tasks commanded by humans in diverse and complex environments. While recent advances in vision-language models (VLMs) have shown promise for high-level semantic reasoning and zero-shot visual question answering [2, 10, 19], these models often lack the intricate understanding of the physical world necessary for precise, low-level robotic control. Consider, for example, the task of organizing a cluttered shoe rack (as shown in Figure 1). This seemingly simple task requires not only recognizing individual shoes and understanding the desired rearranging goal, but also precisely controlling the robot's gripper to navigate the workspace and manipulate objects in various ways. Solving open-world manipulation requires rethinking how to effectively integrate the reasoning capabilities of VLMs with the demands of real-world physics.

Recent works have investigated various ways to employ VLMs for robotic control. Fine-tuned on massive demonstration trajectories through imitation learning, VLMs can serve as the backbone of policies to ground language instructions in visual observations [6, 24, 33]. However, robotics

datasets remain orders of magnitude smaller than the question-answering datasets used to train VLMs, limiting the generalization of learned policies to novel environments and tasks. Alternatively, an increasing number of works propose using VLMs in zero-shot settings with carefully crafted intermediate representations for actions such as value map [19], key points [12], and action arrows [30]. While these approaches show promise in simple manipulation tasks, they require VLM to infer future interactions, thus struggling with more complex scenarios involving intricate contact, dynamics, and motion due to insufficient physical understanding.

In this work, we propose to tackle this challenge following a fundamentally different strategy by decoupling semantic and physical reasoning. In contrast to prior works which demand VLM to implicitly reason dynamics, we employ interactive digital twins of the real-world environment to complement the VLM's capabilities. Through physical simulation, the digital twin explicitly models the dynamics of the environment, bypassing the need for forcing VLM to implicitly reason about low-level interaction. By rendering result images using the digital twin, we can interface the VLM with the information provided by the digital in a way that aligns with the expertise of VLM.

To this end, we present Twin-Assisted Reasoning for Planning (TARP), an approach that solves open-world manipulation in a model-predictive control (MPC) by leveraging a VLM and an interactive digital twin. As shown in Figure 1, given a video scan of the real-world environment, our method first builds an interactive digital twin with a controllable robot and interactable objects to enable physical simulation of possible outcomes of sampled actions. Following sampling-based planning procedure, our method generates candidate actions, simulates their outcomes within the digital twin, and utilizes the VLM to evaluate the resulting future states. Rather than requiring the VLM to reason about physics directly, we render RGB images from the digital twin to provide observations of predicted outcomes as visual prompts to the VLM. Unlike prior work that relies on fixed camera viewpoints, we adaptively adjust camera poses to optimize the visual inputs for the VLM, facilitating more effective reasoning. We design a visual prompting mechanism that enables the VLM to assess the feasibility and desirability of different action outcomes, enabling planning of the optimal actions to solve the task, Thereby, we enhance physical reasoning in VLM-driven robotic control without requiring additional robot data collection, VLM fine-tuning, or any in-context examples.

We evaluate TARP in eight real-world manipulation tasks specified by natural language instructions. Our approach demonstrates to be able to effectively solve these tasks with success rates superior to baseline methods that use VLM for robotic control. Additionally, we conduct thorough ablation studies and failure analysis to investigate the key factors contributing to our framework's performance and robustness.

II. Related Work

Open-world motion planning with VLM. Recent advances in VLMs have opened pathways toward open-world robot manipulation without any robotics demonstration by leveraging their strengths in semantic understanding and common-sense reasoning [9, 2, 14, 43, 34]. However, VLMs, primarily trained on visual question-answering tasks, often struggle to reason about the physical effects of interactions necessary for motion planning. To address this limitation, prior works have explored intermediate representations, such as reward functions, 2D keypoints, action vectors, and affordance [19, 12, 30, 20], as outputs for VLMs. While these approaches show promising results, these settings diverge from VLMs' training distribution. Another line of work involves carefully designed chain-of-thought processes [10, 29, 48], which decompose tasks into smaller, predefined steps. Although effective for specific scenarios, this approach struggles to generalize to tasks that cannot be easily divided into such reasoning procedures. In contrast, we aim to make open-world manipulation an in-distribution task for VLMs leveraging digital twins to simulate future results. This allows VLMs to focus on evaluating possible future observations, aligning seamlessly with their strengths in description and judgment.

Scene reconstruction for robot control. 3D reconstruction methods, including neural radiance fields (NeRFs) [3, 4, 27], neural implicit surfaces [44, 45, 32], and Gaussian Splatting [17, 22], have revolutionized 3D scene modeling by enabling photorealistic rendering and dense geometry reconstruction. These method provide a lot of potential for application in robot motion planning [46, 40, 23, 25, 36]. Previous work leveraging scene reconstruction for motion planning either try to distill 2D features generated by foundational models into 3D for spatial perception or enhance the demonstration to enable few-shot learning. However, these work usually assume the scene to be static and do not handle interactions over the reconstructed representation. In this work, we aim to enable dynamic and interactive modeling between the robot and its environment. To achieve this, we combine Gaussian splatting, known for its efficient and realistic rendering, with mesh for accurate physical modeling. Our hybrid representation leverages the strengths of both approaches, introducing interactive capabilities for robot control applications. This enables dynamic, physically grounded interactions in real-world scenarios, bridging the gap between static scene reconstruction and interactive manipulation.

Motion control with dynamic model. Model Predictive Control (MPC) is a widely adopted optimization-based control strategy that has proven highly effective in robotics [13, 11, 15, 31, 28]. MPC fundamentally consists of two key components: a dynamic model capable of predicting future system states given a sequence of actions, and an optimization process that determines the optimal action sequence by minimizing a predefined cost function. Recent MPC methods for open-world motion planning usually use image of video generation model to predict the future results given different actions [49, 18, 50]

leveraging the prior from large scale pre-training. However, video generation introduces artifact and fails to generalize the complex scenarios that are far from training distribution. Besides, generated video could not guarantee to be precisely based on the low-level action. In this paper, we introduce an accurate world model by building replica of the real-world environments. Besides, compared with traditional method of designing cost function for each task, we employ VLMs for flexible and comprehensive evaluation over the results, enabling open-world manipulation on a wide range of tasks.

Digital twin for robot manipulation. Building digital twin from the real-world scenes to enable manipulation in the real world (i.e., real-to-sim-to-real) is a well-explored problem. However, previous methods usually regard the built digital twin as a virtual playground for collecting manipulation data [35, 47, 21] or training robot policy through reinforcement learning [42, 34, 7, 41, 5], which are then transferred to the real world for manipulation. In this paper, we leverage digital twin as a world model that can provide results of actions that are not happen yet in the real world. By combining with VLM as a critic, we enable a zero-shot setting for real-to-sim-to-real manipulation.

III. Problem Formulation

Our goal is to enable robots to perform open-world manipulation tasks involving unseen objects and diverse goals. We focus on complex scenarios that involve intricate contact, dynamics, and motion. These setups require robots to possess a thorough physical and semantic understanding of the environment. We do not assume access to task-specific training data, in-context examples, or hard-coded motion primitives as used in prior work [19, 26, 12, 24].

We develop our approach using pre-trained vision language models (VLMs). We denote the VLM as $\mathcal{M}$, which takes inputs as a list of text and RGB images provided in a specific order, and outputs text responses. We employ $\mathcal{M}$ for various purposes in the proposed approach by designing different text prompts as part of the inputs.

We consider a table-top setting with one robotic arm. The framework's input consists of a natural language instruction l specifying the task, and an RGB video scan v of the scene. The output is an action sequence $\{a\}_{t=0}^{T}$ for achieving the task goal. Each action $a_t \in \mathbb{R}^7$ is defined as the 6-DoF gripper pose and the finger status.

IV. Method

We propose to tackle open-world motion planning by complementing VLM's high-level reasoning ability with dynamics modeling by the digital twins. At the heart of our framework are two key components: 1) We introduce a pipeline to automatically build interactive digital twins to support accurate modeling of diverse physical interactions and photorealistic rendering of the simulation outcomes. IV-A. 2) We formulate open-world manipulation as a model predictive control problem by prompting the VLM with futures provided by the digital twin, enabling adaptive observation and action optimization. IV-C.

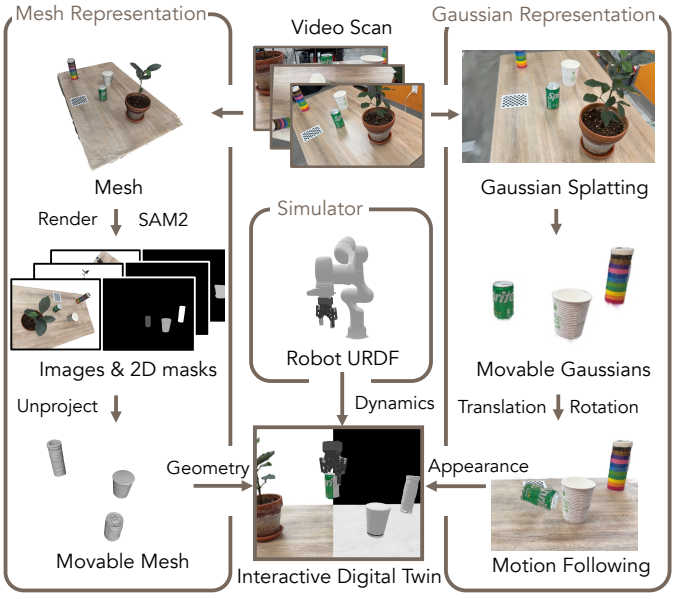

Fig. 2: **Interactive Digital Twin.** The building process begins by reconstructing Gaussian splatting and mesh from the input video. The generated mesh is then rendered into images, and a 2D segmentation method is applied to obtain 2D object masks. These masks are projected back into 3D to segment all movable object mesh. We anchor the Gaussian points to their corresponding object meshes to follow the object motion. Finally, a simulator with a robot URDF is integrated. The resulting interactive digital twin combines the mesh representation for geometry, the Gaussian representation for appearance, and the simulator for dynamics.

A. Interactive Digital Twin

In order to build digital twins that could provide VLM with the future, our construction pipeline needs to have two key stages: 1) reconstructing the static scene with photo-realistic appearance and precise geometries; 2) making the scene interactable for simulating results of diverse actions.

Reconstruction: As shown in Figure 2, we start by reconstructing the static scene given a video scan of the real world. The desired reconstruction should contain both accurate geometry for physical simulation and photorealistic appearance for prompting VLMs. Towards this goal, we employ a hybrid scene representation. Given the video scan, we first use 2D Gaussian splatting [17] to build a Gaussian representation of the scene as G . The creation of Gaussian representation G is supervised by the RGB values of the extracted images of the video. Then, we convert the Gaussian representation into a mesh representation M by TSDF volume integration, which preserves detailed geometry information of the real world.

Geometry: In contrast to previous reconstruction methods [39, 23], we enable the reconstructed G to support modeling of controllable robots and movable objects. Therefore, we

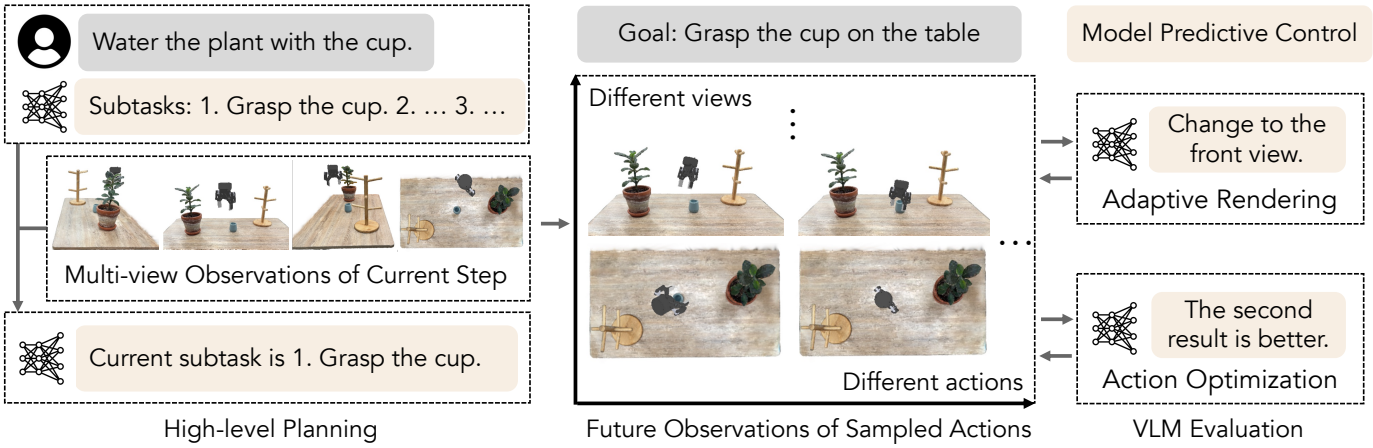

Fig. 3: **Model Predictive Control with VLM.** Our policy begins with high-level planning, generating subtasks based on language instructions and multi-view observations. Following this, low-level model predictive control is initiated. Future observations of diverse actions from different views are simulated by the digital twin, which are evaluated by VLM to adaptively change the camera view and identify the most promising results. Our system fits a distribution to the selected actions for further resampling, which finally leads to an optimized action. The numbers of sampled actions and views are just for illustration.

need to segment the reconstructed objects in G and assign them with physical properties. To achieve this, we render the reconstructed mesh M as multi-view images. The first image is used as the input of Molmo [8], a VLM that can generate 2D points on the images given language specification, to point out the movable objects given the task instruction l. With the points on the movable objects in the first image, we use SAM2 [37] to track the 2D segmentation mask of these objects in the multi-view images. The predicted 2D masks are then projected back to 3D space as labels on mesh vertices, which segments out the 3D meshes of movable objects in the mesh representation M.

Appearance: Having only movable meshes for geometric information is not enough; we also need high visual fidelity from any viewpoint and ensure realistic rendering after any possible movement. Therefore, the second step is to enable the Gaussian splats G to follow the movement of object meshes to provide a realistic and responsive appearance. Firstly, with the movable meshes, we segment out the Gaussian points based on their Euclidean distance to the object meshes. Then, we anchor these Gaussian points to their corresponding meshes, copying the translation and rotation of the object, thus keeping the appearance following the possible motions.

Dynamics: After enabling the scene to be responsive to changes, we introduce the dynamics that simulate the changes given actions. A simulator S [16] with a robot URDF file U is imported to apply diverse actions and provide accurate dynamics of these interactions based on the geometry and appearance provided.

Following these steps, an interactive digital twin is built with mesh representation providing object geometries, Gaussian representation providing appearance, and the simulator providing dynamics.

B. Prediction of Future Observations

We aim to create a setting that fully aligns with its core strength—describing and evaluating images by enabling the VLM to "see" the futures. Our digital twins leverage the simulator S to simulate the movement of the robot and objects $T, R = S(M, a)$ in the mesh representation M given different robot actions $a \in \mathbb{R}^7$, where T and R stand for translation and rotation. The movements are then copied to the Gaussian representation G to update the state after the interactions $G(R, T)$, providing physically grounded and realistic feedback of possible actions.

Since VLMs could only take 2D images, they inherently struggle with spatial reasoning. We propose to mitigate this limitation by allowing the model to cross-validate its evaluations across multiple viewpoints. With the updated state, our reconstruction could provide observations from any camera views at any manipulation step. To achieve this, we render the updated Gaussian representation with the given configurations $I_g = Render(G(R, T), C)$, which do not contain the robot. The robot images under action a are rendered by the simulator with URDF file $I_r = Render(S(U, a))$. The scene images and robot images are merged based on depth information to generate multi-view observations $I = \{I_i \in \mathbb{R}^{W \times H \times 3}\}$ of the manipulation scene at any time step, which could be fed into the VLM for 3D perception.

C. Model Predictive Control through Prompting

After building the interactive digital twin of the real world offering the opportunity to "see" the futures, the problem then becomes how to better leverage this setting to harness VLM for open-world manipulation.

High-level planning: As shown in Figure 3, we start with leveraging the high-level planning ability of VLM. We first input multi-view observations I_0 of the initial state and the task

instruction l to VLM to generate subtasks $T = \{T_1, T_2, ...\}$, which is a list of text containing structured information for the subgoals (e.g. grasp the cup on the table) that need to be achieved to complete the whole task (e.g. water the plant). At each planning step t, given the observations of the current state I_t and the subtasks T, the VLM is required to judge which stage the agent is in (i.e. what is the current subtask). The chosen subtask T_i will become the goal for low-level control of this step. This high-level decomposition of the task helps to better guide the direction of the following low-level optimization.

Adaptive rendering: With the selected subtask T_i, then it comes to low-level motion control of each planning step t. However, while robot action is in $SE(3)$ space, the VLM could only take 2D images, limiting the spatial reasoning ability when judging the results. Therefore, the selection of rendering view is crucial for planning. Thanks to our digital twin that can offer infinite views, for each planning step, we prompt VLM with the selected subtask T_i and observations of the current state I_t under different camera configurations C, and let the VLM choose the best camera view C_t for this step. For example, while a top-down view can not distinguish the two actions in Figure 3, VLM changes the view to the front, where the difference of actions is clearly shown for VLM to select better results.

Action sampling: After determining the subtask as well as the best view, we now discuss how to get the action for the current planning step. We propose to offload all interaction reasoning burden to the digital twin and let VLM focus on evaluation. In other words, VLM should control low-level actions in an action-agnostic way, which brings us to a model predictive control setting. With this aim, a straightforward method is to sample broadly and prompt VLM with diverse results to choose the best one as the output action for the current step. While this naive sampling-based method could theoretically work, it's impractical due to two reasons: randomly sampling actions in the action space is inefficient since most of the actions will not get any closer to the goal; prompting VLM with hundreds of results will increase the burden of the model for evaluation.

Action optimization: To overcome this issue, we apply a cross-entropy method [38] to improve sampling efficiency and decrease the conceptual burden of VLM. We first sample n actions $\{a_k\}$ from a multivariate Gaussian distribution, simulate their results and get observations $o_{t,k}$ showing the future states. To alleviate the conceptual burden of VLM, we batchify the results to ensure VLM is not overwhelmed by the number of images. We divide these observations into m groups. Images of each group are used to prompt a VLM with the subtask instruction T_i, where the VLM is asked to simply select the result that is closest to the goal from the group. By prompting the VLM of each group in parallel, we get m elite samples, which are used to optimize a Gaussian distribution representing the actions that lead to better results. Then we sample n actions again from this new distribution and prompt VLM for evaluation, which in theory leads to more optimized actions. We repeat this sampling and evaluation procedure for 3 times and use the mean value of the final distribution as the optimized action a_t for this step. As for the gripper motion, since our framework could offer contact information of future states, we simulate grasping or releasing after every arm movement and provide the resulting observations to VLM to decide whether to close/open the gripper or not.

After optimizing the action, the agent executes the action in the digital twin and enters the next step of high-level planning and low-level optimization. Our method will stop planning when the task is complete or the step budget is reached.

Note that, in both high-level planning and low-level action optimization, the VLM is never required to reason about interaction results or output anything that corresponds to an action. We keep the setting to the evaluation of visual observations, which fully aligns with the expertise of VLM for description and evaluation.

V. Experiments

To validate the effectiveness of our framework, in this section, we design eight real-world manipulation tasks that require 6 DoF control, semantic understanding, and diverse manipulation skills. We compare our approach against prior works on open-world manipulation that leverage VLMs or train on large-scale robotic data. Additionally, we conduct ablation experiments to evaluate the contribution of each component to the overall performance.

A. Experimental setup

Tasks. We introduce eight manipulation tasks that require intricate understanding of the physical world and diverse manipulation skills: `water the plant`, `play the drum`, `press spacebar`, `pair up shoes`, `put cucumber in the basket`, `play the lowest tune`, `unplug charger`, and `clean up`. For each task, we construct five manipulation scenes, featuring randomized object layouts and different distractors. Please see the supplementary material for more details about task design.

Baselines. We compare our model against several state-of-the-art methods. **VoxPoser*** [19] leverages a VLM to predict 3D value map for motion optimization. We enhance it by providing ground-truth segmented object point clouds from our digital twin, significantly improving its perception accuracy. **MOKA** [12] chooses the 2D keypoints as intermediate representations for VLM to predict, which are then converted into actions based on the depth information from a depth camera. **OpenVLA** [24] is a 7B-parameter open-source vision-language-action model fine-tuned from a VLM using 970k real-world robot demonstrations [33] .

Metrics. We use success rate as the evaluation metric. A task is considered a failure if the robot causes irreversible results or if the maximum step budget or time limit is reached. Please see the supplementary material for more details.

Methods	Water plant	Play the drum	Press spacebar	Pair up shoes	Cucumber in basket	Lowest tune	Unplug charger	Clean up
Voxposer* [19]	**6/10**	2/10	0/10	2/10	8/10	0/10	4/10	0/10
MOKA [12]	4/10	5/10	0/10	2/10	5/10	0/10	0/10	5/10
OpenVLA [24]	1/10	0/10	1/10	0/10	4/10	0/10	0/10	0/10
Ours	5/10	**6/10**	**5/10**	**6/10**	**9/10**	**4/10**	**7/10**	**5/10**

TABLE I: **Comparison against baselines**. TARP better leverages the reasoning ability of VLM and improve the performance on most of the tasks. Besides, our image-based evaluation on results provides more flexibility than value map or key points, enabling challenging tasks that are hard to be parametrized by previous representations (e.g. play lowest tune).

Implementation details. We adopt GPT-4o [1] for both our method and the baselines. During inference, we render four camera views and allow VLM to select observations from these perspectives. For the CEM optimization, we use 3 iterations with 90 samples per iteration. The planning policies are rolled out twice per scene to consider the randomness in VLM planning, resulting in 10 trials per task in total. Please see the supplementary material for more details about task and baseline design.

B. Quantitative results

Table I compares the success rates of our method against those of the baselines. Since Voxposer and MOKA rely on open-vocabulary detectors to detect objects before manipulation, they fail when the perception system cannot recognize specific object parts, such as the spacebar on a keyboard or the lowest key on a xylophone. In contrast, our method directly leverages the VLM to comprehend and reason about simulated future states, eliminating the need for a perception module and resulting in greater robustness. As for OpenVLA, while it can perform zero-shot on simple tasks due to its training on large-scale robotic datasets, its generalization is limited by the coverage of its training data, making it less effective for complex tasks. In contrast, our method benefits from the commonsense reasoning capabilities of the VLM, enabling broader task coverage and adaptability. We particularly excel in tasks requiring precise gripper pose alignment, as our approach allows for simulation-based "rehearsal" prior to execution.

C. Qualitative results

We visualize the action optimization process for a single planning step in the "clean up" task in Figure 4. Initially, the digital twin simulates a diverse set of actions with precise dynamics (*e.g.*, object rotating due to grasping, object dropping due to collisions, etc.) and photorealistic rendering. Based on the future simulation results and task instruction, the VLM selects elite actions and resamples actions for further simulation in the digital twin. This iterative process leads to an optimized action distribution that aligns more closely with the goal of wiping the spilled tea using the sponge.

Figure 5 shows randomly sampled rollout trajectories in both digital twins and the real world. By mirroring possible interactions in the simulated world, our framework provides a flexible and effective way for VLMs to guide the motion of the robot on diverse tasks from open-world environment. The digital twin and the real world are aligned based on planning steps. We highlight key planning steps where VLM chooses to change the observation view to better assess the results, showing the benefits of our adaptive rendering design. Please see the supplementary material for more visualization and videos.

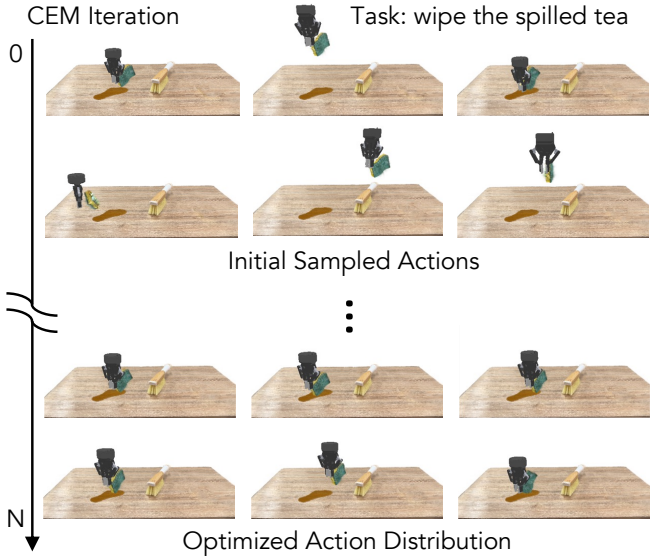

Fig. 4: **Example on action optimization.** We show the action optimization results of one planning step in subtask "wipe the spilled tea". Our digital twin could simulate diverse results with accurate motion and collision of the sponge in initial sampling and VLM could effectively optimize the action distribution to move the sponge towards the tea.

D. Ablation study

To assess the contribution of each component in our framework, we begin with the full system and systematically remove each component in turn. In the "w/o views" setting, we fix the camera to a static top-down perspective instead of allowing the VLM to select a view for each planning step. For "w/o subtasks," we use the user instruction directly as the goal for each planning step rather than first generating subtasks. In the "w/o CEM" setting, we simply take the mean value of the selected actions without optimizing the action distribution or resampling.

Fig. 5: **Example trajectories.** Example trajectories planned by our framework in both digital twin and real world (aligned). We highlight some key steps where VLM chooses to adaptively change the rendering view for better result evaluation (e.g., aligning the gripper from different perspectives for grasping, pressing, placing or hitting).

Methods	Water plant	Play the drum	Press spacebar	Pair up shoes	Cucumber in basket	Lowest tune	Unplug charger	Clean up
w/o. Views	1/10	0/10	0/10	3/10	8/10	0/10	3/10	1/10
w/o. Subtask	5/10	0/10	**6/10**	3/10	8/10	2/10	7/10	0/10
w/o. CEM	0/10	2/10	2/10	2/10	1/10	0/10	4/10	0/10
Ours Full	**5/10**	**6/10**	5/10	**6/10**	**9/10**	**4/10**	**7/10**	**5/10**

TABLE II: **Ablation study.** We validates the effectiveness of our components. Multi-view observations are essential for tasks that are sensitive to perspectives. Subtask division improves the performance on tasks that requires multi-stage planning. Most importantly, CEM significantly increases the sampling efficiency, facilitate effective planning.

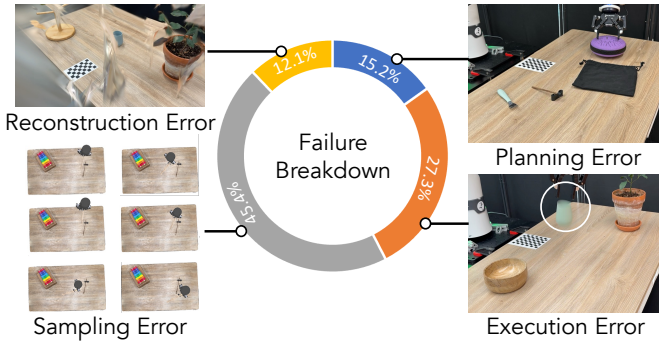

Fig. 6: **Failure analysis.** Our main failure cases can be divided into four categories. We show the percentage and provide an example for each failure type.

As shown in Table II, while performance varies across different tasks due to their diverse requirements, our full method achieves the best results in most of the tasks. Multi-view observations, enabled by our adaptive rendering, enhance spatial perception, particularly in tasks requiring precise control (e.g., Press spacebar, Play the drum). Dividing tasks into subtasks allows the VLM to focus on more concrete evaluation criteria for each state, which is beneficial for tasks involving multiple stages such as playing the drum and cleaning up. Lastly, the CEM process significantly improves sampling efficiency, producing action distributions that better align with the goal, which in general contributes the most to our model predictive control framework.

E. Failure breakdown

The failure cases can be categorized into four groups:

- **Reconstruction error:** The quality of our digital twin depends on the accuracy of camera pose estimation. Noisy poses often result in visual artifacts, which can further reduce the effectiveness of the VLM.
- **Planning error:** When subtasks are not properly defined or the model fails to recognize the current stage, the robot may execute actions incorrectly. For example, the robot may attempt to move the gripper directly to the drum without first picking up the drumstick.
- **Execution error:** In the real world, the gripper may misalign during grasping or drop objects during movement due to insufficient grip or friction.
- **Sampling error:** This is the primary source of failure in our framework. Due to the inherent randomness of action sampling and errors in VLM reasoning, the system may struggle to sample actions to achieve the goal within a limited step budget.

We show their percentage and some examples in Figure 6.

VI. Limitations

With a novel MPC formulation, we aim to study how far we can go in open-world manipulation via VLM without any robotic data or in-context examples. The current pipeline is far from fully solving all motion planning tasks. One notable limitation is the computational speed. Reconstructing 3D representation from real world is still time-consuming (e.g., about 20 minutes for each scene). Besides, prompting VLM also costs time for waiting response and generating tokens. The limitation on speed might hinder the real-time applications. However, we believe that advancements in 3D reconstruction techniques and using open-source VLMs could significantly reduce the inference time. Another limitation is brought by the ability of VLMs. Large models are not perfect on perception, which leads to some inaccurate evaluation on visual input, introducing randomness in our sampling-based planning. But one advantage of our framework is that our performance will benefit from the improvements in VLM capabilities. Our system with future models could be more robust and efficient.

VII. Conclusion

We propose to enhance the high-level reasoning capabilities of vision-language models (VLMs) by integrating interactive digital twins that simulate low-level physical interactions. By constructing a hybrid representation of the real world and enable interactions, our system provides VLMs with diverse, accurate, and photorealistic future scenarios under different actions to evaluate. Leveraging the inherent strengths of VLMs in description and judgment, our approach optimizes robotic actions within a model predictive control framework with adaptive rendering and iterative sampling, pushing the boundaries of open-world manipulation through VLMs. Comparisons with existing VLM- or VLA-based methods on a wide range of tasks demonstrate the effectiveness of our novel formulation for bridging the priors of VLM with robot control.

References

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

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

[3] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *ICCV*, 2021.

[4] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *CVPR*, 2022.

[5] Cristian Camilo Beltran-Hernandez, Nicolas Erbetti, and Masashi Hamaya. Sliceit!: Simulation-based reinforcement learning for compliant robotic food slicing. In *ICRA Workshop*, 2024.

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

[7] Tianyuan Dai, Josiah Wong, Yunfan Jiang, Chen Wang, Cem Gokmen, Ruohan Zhang, Jiajun Wu, and Li Fei-Fei. Automated creation of digital cousins for robust policy learning. *arXiv:2410.07408*, 2024.

[8] Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, Jiasen Lu, Taira Anderson, Erin Bransom, Kiana Ehsani, Huong Ngo, YenSung Chen, Ajay Patel, Mark Yatskar, Chris Callison-Burch, Andrew Head, Rose Hendrix, Favyen Bastani, Eli VanderBilt, Nathan Lambert, Yvonne Chou, Arnavi Chheda, Jenna Sparks, Sam Skjonsberg, Michael Schmitz, Aaron Sarnat, Byron Bischoff, Pete Walsh, Chris Newell, Piper Wolters, Tanmay Gupta, Kuo-Hao Zeng, Jon Borchardt, Dirk Groeneveld, Jen Dumas, Crystal Nam, Sophie Lebrecht, Caitlin Wittlif, Carissa Schoenick, Oscar Michel, Ranjay Krishna, Luca Weihs, Noah A. Smith, Hannaneh Hajishirzi, Ross Girshick, Ali Farhadi, and Aniruddha Kembhavi. Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models. *arXiv:2409.17146*, 2024.

[9] 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:2303.03378*, 2023.

[10] Jiafei Duan, Wentao Yuan, Wilbert Pumacay, Yi Ru Wang, Kiana Ehsani, Dieter Fox, and Ranjay Krishna. Manipulate-anything: Automating real-world robots using vision-language models. *arXiv:2406.18915*, 2024.

[11] 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:1812.00568*, 2018.

[12] Kuan Fang, Fangchen Liu, Pieter Abbeel, and Sergey Levine. Moka: Open-vocabulary robotic manipulation through mark-based visual prompting. In *RSS*, 2024.

[13] Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *ICRA*, 2017.

[14] Jensen Gao, Bidipta Sarkar, Fei Xia, Ted Xiao, Jiajun Wu, Brian Ichter, Anirudha Majumdar, and Dorsa Sadigh. Physically grounded vision-language models for robotic manipulation. In *ICRA*, 2024.

[15] Ruben Grandia, Farbod Farshidian, René Ranftl, and Marco Hutter. Feedback mpc for torque-controlled legged robots. In *IROS*, 2019.

[16] Jiayuan Gu, Fanbo Xiang, Xuanlin Li, Zhan Ling, Xiqiang Liu, Tongzhou Mu, Yihe Tang, Stone Tao, Xinyue Wei, Yunchao Yao, Xiaodi Yuan, Pengwei Xie, Zhiao Huang, Rui Chen, and Hao Su. Maniskill2: A unified benchmark for generalizable manipulation skills. In *ICLR*, 2023.

[17] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *SIGGRAPH*, 2024.

[18] Siyuan Huang, Zan Wang, Puhao Li, Baoxiong Jia, Tengyu Liu, Yixin Zhu, Wei Liang, and Song-Chun Zhu. Diffusion-based generation, optimization, and planning in 3d scenes. In *CVPR*, 2023.

[19] 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 *CoRL*, 2023.

[20] Wenlong Huang, Chen Wang, Yunzhu Li, Ruohan Zhang, and Li Fei-Fei. Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. In *CoRL*, 2024.

[21] Zhenyu Jiang, Yuqi Xie, Kevin Lin, Zhenjia Xu, Weikang Wan, Ajay Mandlekar, Linxi Fan, and Yuke Zhu. Dexmimicgen: Automated data generation for bimanual dexterous manipulation via imitation learning. *arXiv:2410.24185*, 2024.

[22] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM TOG*, 2023.

[23] Justin Kerr, Chung Min Kim, Mingxuan Wu, Brent Yi, Qianqian Wang, Ken Goldberg, and Angjoo Kanazawa. Robot see robot do: Imitating articulated object manipulation with monocular 4d reconstruction. In *CoRL*, 2024.

[24] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv:2406.09246*, 2024.

[25] Yunzhu Li, Shuang Li, Vincent Sitzmann, Pulkit Agrawal, and Antonio Torralba. 3d neural scene representations for visuomotor control. In *CoRL*, 2022.

[26] 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 *ICRA*, 2023.

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

[28] Maria Vittoria Minniti, Ruben Grandia, Kevin Fäh, Farbod Farshidian, and Marco Hutter. Model predictive robot-environment interaction control for mobile manipulation tasks. In *ICRA*, 2021.

[29] Yao Mu, Qinglong Zhang, Mengkang Hu, Wenhai Wang, Mingyu Ding, Jun Jin, Bin Wang, Jifeng Dai, Yu Qiao, and Ping Luo. Embodiedgpt: Vision-language pre-training via embodied chain of thought. In *NeurIPS*, 2024.

[30] Soroush Nasiriany, Fei Xia, Wenhao Yu, Ted Xiao, Jacky Liang, Ishita Dasgupta, Annie Xie, Danny Driess, Ayzaan Wahid, Zhuo Xu, et al. Pivot: Iterative visual prompting elicits actionable knowledge for vlms. *arXiv:2402.07872*, 2024.

[31] 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. *RA-L*, 2020.

[32] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *ICCV*, 2021.

[33] Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv:2310.08864*, 2023.

[34] Shivansh Patel, Xinchen Yin, Wenlong Huang, Shubham Garg, Hooshang Nayyeri, Li Fei-Fei, Svetlana Lazebnik, and Yunzhu Li. A real-to-sim-to-real approach to robotic manipulation with vlm-generated iterative keypoint rewards. In *CoRL Workshop*, 2024.

[35] Weikun Peng, Jun Lv, Yuwei Zeng, Haonan Chen, Siheng Zhao, Jichen Sun, Cewu Lu, and Lin Shao. Tiebot: Learning to knot a tie from visual demonstration through a real-to-sim-to-real approach. *arXiv:2407.03245*, 2024.

[36] Mohammad Nomaan Qureshi, Sparsh Garg, Francisco Yandun, David Held, George Kantor, and Abhisesh Silwal. Splatsim: Zero-shot sim2real transfer of rgb manipulation policies using gaussian splatting. *arXiv:2409.10161*, 2024.

[37] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv:2408.00714*, 2024.

[38] Reuven Rubinstein. The cross-entropy method for combinatorial and continuous optimization. *Methodology and computing in applied probability*, 1999.

[39] William Shen, Ge Yang, Alan Yu, Jansen Wong, Leslie Pack Kaelbling, and Phillip Isola. Distilled feature fields enable few-shot language-guided manipulation. In *CoRL*, 2023.

[40] William Shen, Ge Yang, Alan Yu, Jansen Wong, Leslie Pack Kaelbling, and Phillip Isola. Distilled feature fields enable few-shot language-guided manipulation. In *CoRL*, 2023.

[41] Marcel Torne, Arhan Jain, Jiayi Yuan, Vidaaranya Macha, Lars Ankile, Anthony Simeonov, Pulkit Agrawal, and Abhishek Gupta. Robot learning with super-linear scaling. *arXiv:2412.01770*, 2024.

[42] Marcel Torne, Anthony Simeonov, Zechu Li, April Chan, Tao Chen, Abhishek Gupta, and Pulkit Agrawal. Reconciling reality through simulation: A real-to-sim-to-real approach for robust manipulation. *arXiv:2403.03949*, 2024.

[43] Beichen Wang, Juexiao Zhang, Shuwen Dong, Irving Fang, and Chen Feng. Vlm see, robot do: Human demo video to robot action plan via vision language model. *arXiv:2410.08792*, 2024.

[44] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv:2106.10689*, 2021.

[45] Yiming Wang, Qin Han, Marc Habermann, Kostas Daniilidis, Christian Theobalt, and Lingjie Liu. Neus2: Fast learning of neural implicit surfaces for multi-view reconstruction. In *ICCV*, 2023.

[46] Yixuan Wang, Mingtong Zhang, Zhuoran Li, Tarik Kelestemur, Katherine Rose Driggs-Campbell, Jiajun Wu, Li Fei-Fei, and Yunzhu Li. D ^{3} fields: Dynamic 3d descriptor fields for zero-shot generalizable rearrangement. In *CoRL*, 2024.

[47] Yuxuan Wu, Lei Pan, Wenhua Wu, Guangming Wang, Yanzi Miao, and Hesheng Wang. Rl-gsbridge: 3d gaussian splatting based real2sim2real method for robotic manipulation learning. *arXiv:2409.20291*, 2024.

[48] Michał Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. Robotic control via embodied chain-of-thought reasoning. *arXiv:2407.08693*, 2024.

[49] Wentao Zhao, Jiaming Chen, Ziyu Meng, Donghui Mao, Ran Song, and Wei Zhang. Vlmpc: Vision-language model predictive control for robotic manipulation. *arXiv:2407.09829*, 2024.

[50] Xinxin Zhao, Wenzhe Cai, Likun Tang, and Teng Wang. Imaginenav: Prompting vision-language models as embodied navigator through scene imagination. *arXiv:2410.09874*, 2024.