SIMPACT: Simulation-Enabled Action Planning using Vision-Language Models

Haowen Liu*,[1] Shaoxiong Yao*,[2] Haonan Chen[3] Jiawei Gao[3]
Jiayuan Mao[4,5] Jia-Bin Huang[1] Yilun Du[3]
[1]UMD, [2]UIUC, [3]Harvard, [4]Amazon FAR, [5]UPenn

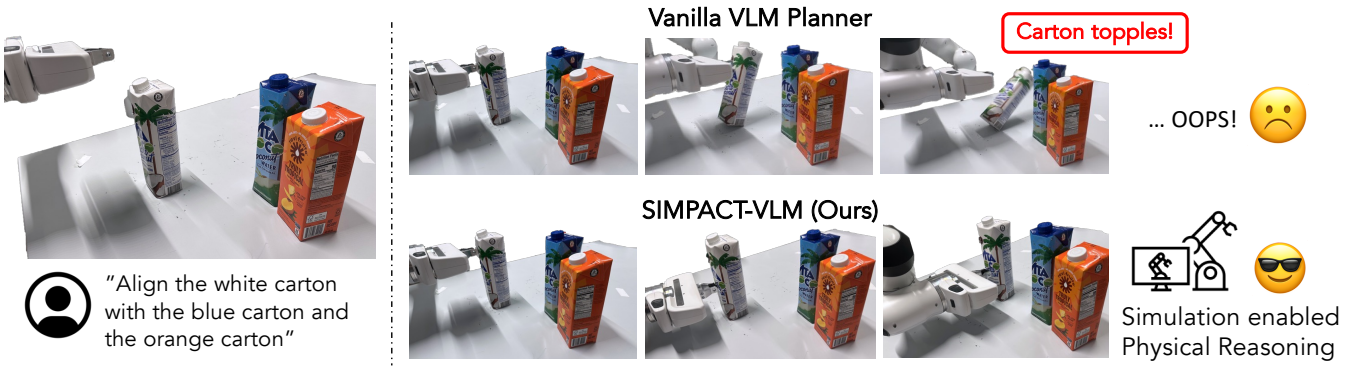

Fig. 1: **Simulation-Enable VLM Action Planning.** Given a single RGB–D image and a language task description (*left*), our method efficiently constructs a physics simulator that enables test-time VLM reasoning with physical grounding. This physically grounded reasoning allows the robot to succeed in fine-grained manipulation tasks (*bottom*), outperforming a vanilla VLM planner (*top*) that lacks awareness of physical dynamics.

***Abstract*— Vision-Language Models (VLMs) exhibit remarkable common-sense and semantic reasoning capabilities. However, they lack a grounded understanding of physical dynamics. This limitation arises from training VLMs on static internet-scale visual-language data that contain no causal interactions or action-conditioned changes. Consequently, it remains challenging to leverage VLMs for fine-grained robotic manipulation tasks that require physical understanding, reasoning, and corresponding action planning. To overcome this, we present SIMPACT, a test-time, SIMulation-enabled ACTion Planning framework that equips VLMs with physical reasoning through simulation-in-the-loop world modeling, without requiring any additional training. From a single RGB-D observation, SIMPACT efficiently constructs physics simulations, enabling the VLM to propose informed actions, observe simulated rollouts, and iteratively refine its reasoning. By integrating language reasoning with physics prediction, our simulation-enabled VLM can understand contact dynamics and action outcomes in a physically grounded way. Our method demonstrates state-of-the-art performance on seven challenging, real-world rigid-body and deformable manipulation tasks that require fine-grained physical reasoning, outperforming existing general-purpose robotic manipulation models. Our results demonstrate that embedding physics understanding via efficient simulation into VLM reasoning at test time offers a promising path towards generalizable embodied intelligence.**

I. Introduction

General-purpose robots hold significant promise for handling complex, labor-intensive tasks in unstructured environments, but realizing this potential requires advanced scene perception and robust action planning. Vision-Language Models (VLMs), trained on static internet-scale visual and language data, offer a promising solution by equipping robots to understand scenes and respond to diverse queries. These models can understand object semantics, infer task goals, and generate action descriptions aligned with human intent [1], [2], [3]. However, despite their remarkable commonsense and semantic reasoning capabilities, VLMs lack a **grounded understanding of physical dynamics**. They can describe what to do, but often fall short in predicting how actions will unfold when executed in the physical world.

As such, VLMs have shown limited capabilities in robotic manipulation, particularly for tasks involving rich physical interactions, such as turning an object in place or carefully stacking objects. These tasks require reasoning about how objects behave under forces and constraints, where small variations in contact or timing can lead to drastically different outcomes. Lacking physical understanding, VLMs often propose plans that appear reasonable in language but fail during execution.

To address this limitation, we propose a framework that augments VLMs with physical simulation rollouts as contextual feedback, enabling test-time physical reasoning for action planning. Our approach begins with a novel simulation generation pipeline that leverages pretrained visual foundation models – including segmentation, 3D generation, and pose estimation models, to build a physical simulator directly from a single-view RGB-D image efficiently. In addition, we use VLMs to automate the setup of a multi-physics simulator, enabling it to model the behavior of both rigid and deformable objects across diverse material properties. The resulting physical simulation characterizes intricate contact dynamics that are difficult to infer from static images and

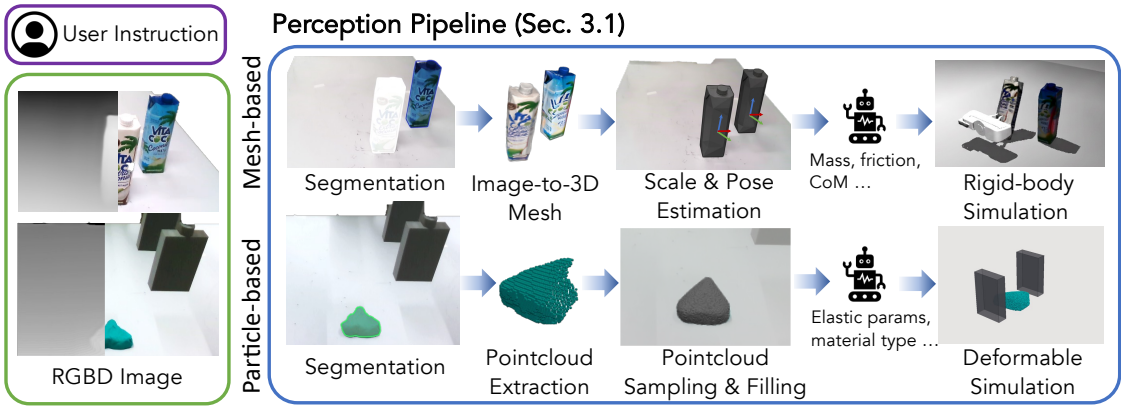

Fig. 2: **Simulation construction from single RGBD image.** Given an RGB-D image and a language task description, our pipeline automatically generates either a mesh-based simulation (*top*) for rigid objects or a particle-based simulation (*bottom*) for deformables. After segmenting objects-of-interest via GroundedSAM2 [4], we reconstruct either the 3D shape, scale, and pose of the object for rigid-body simulation, or perform dense sampling of particles within the volumes between the object surface and the table for the particle-based simulation pipeline. In both cases, we prompt the VLM to infer the relevant physical parameters required for simulation.

language alone, providing VLMs with physical insights for manipulation planning.

Powered by the generated simulation, we introduce a planning framework driven by VLMs' reasoning capabilities. Our key idea is to leverage the rich prior knowledge of VLMs to generate action sequence proposals, and to use simulated rollouts as context for the VLM to iteratively refine these proposals. This test-time reasoning paradigm, inspired by model-based control frameworks [5], [6], enables VLMs to reason not only about the world through language but also about its dynamics through simulated interaction. By augmenting VLMs with physical simulation, our framework enables them to anticipate action consequences, evaluate predicted outcomes, and iteratively adjust their decisions at test time, without any task-specific training. This process unlocks significantly stronger physical reasoning, enabling more reliable and robust real-world performance than state-of-the-art general-purpose manipulation models.

In summary, this paper makes the following contributions:

- We introduce a test-time, zero-shot framework enabling VLMs to plan physics-aware embodied actions;
- We present a pipeline for automatically generating multi-physics simulations from single RGB-D observation using visual foundation models and VLM;
- We propose a novel in-context learning approach for robot action generation, where physics simulation serves as context, enabling a new form of test-time reasoning in robotics.

II. Method

Our framework enables zero-shot robotic manipulation action generation from a single RGB-D image input I_0 and natural language instruction ℓ_{task} and outputs robot action sequence $\mathbf{a} = \{a_t\}_{1 \le t \le T}$, where $a_t \in \text{SE}(3) \times \mathbb{R}$, defining end-effector pose and gripper open width. For each task, the natural-language specification ℓ_{task} defines the task requirements, along with potential success and failure conditions, to guide the VLM in proposing plausible actions.

Our simulation-enabled VLM planning framework operates as illustrated in Fig. 3. First, we construct a physical simulator SIM using an automated perception pipeline that reconstructs complete 3D geometries and configures appropriate simulation parameters as shown in Fig. 2. Next, we instantiate a manipulation planner that integrates the simulator with a VLM as its core reasoning module. The planner begins by generating a scene context from an initial visual observation, which is augmented with robot proprioceptive data and object states. Based on this context and prior knowledge, the VLM proposes action sequences, which are evaluated through simulation rollouts. The resulting visual observations and object states from each rollout are then fed back to the VLM as additional context for iterative refinement. This process continues until a rollout is validated as successful. Finally, the optimized action sequence is executed as end-effector commands on the real robot system.

A. Simulation Construction

Our approach employs a physics-based simulator to predict the consequences of actions for manipulation planning. The simulation follows the discrete-time state transition:

$$s_t = \text{SIM}(s_{t-1}, a_t; \theta) \tag{1}$$

where s_t denotes the state at time step t, a_t represents the applied action, and θ comprises time-invariant simulation parameters. The state space captures all task-relevant information: rigid objects are represented by a 6DoF pose in $\text{SE}(3)$, while deformable objects are described by N particle positions in $\mathbb{R}^{3\times N}$. We initialize the state as s_0, assuming objects remain static prior to interaction, and construct parameters via $\theta = \text{CreateSim}(I_0)$ from the initial RGBD image I_0. Here, the simulation parameters are defined as $\theta = (\theta_{\text{geom}}, \theta_{\text{phys}})$, where θ_{geom} specifies the object shape and pose, and θ_{phys} characterizes its mechanical properties.

Our geometry pipeline begins by prompting a VLM to generate object labels based on the user's instructions, as shown in Fig. 2. We first apply a pretrained segmentation model, GroundedSAM2 [7], [4], to segment each identified

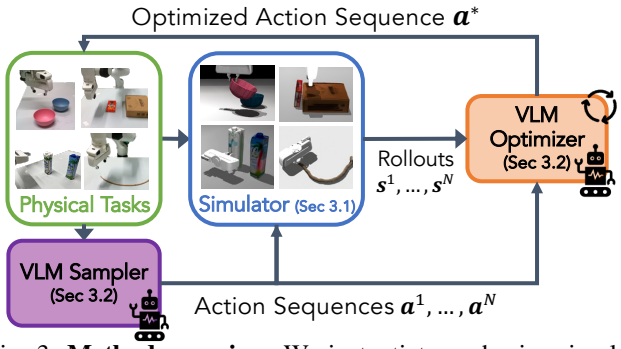

Fig. 3: **Method overview**. We instantiate a physics simulator given the real-world scene. Next, a VLM-based action sampler and optimizer iteratively refine the action sequence towards task success using simulated rollouts as context. The final optimized actions are then executed in the real world.

Algorithm 1: Action Planning Algorithm

Input: VLM, SIM, I_0, ℓ_{task}, s_0;
$\mathcal{A} = \emptyset$, $\mathcal{S} = \emptyset$;
for $k = 1..K$ **do**
  $\mathcal{A} \leftarrow \mathcal{A} \cup \{\mathbf{a}^i \leftarrow \text{SAMPLE}(I_0, \ell_{\text{task}}, s_0; \text{VLM})\}$;
  $\mathcal{S} \leftarrow \mathcal{S} \cup \{\mathbf{s}^i \leftarrow \text{SIMROLLOUT}(s_0, \mathbf{a}^i; \text{SIM})\}$;
end
for $k = K{+}1$ **to** $K_{\max}$ **do**
  $\mathbf{a}^k \leftarrow \text{OPTIMIZE}(\mathcal{A}, \mathcal{S}, \ell_{\text{task}}; \text{VLM})$;
  $\mathbf{s}^k \leftarrow \text{SIMROLLOUT}(s_0, \mathbf{a}^k; \text{SIM})$;
  if $\text{TASKSUCCESS}(\mathbf{s}^k; \text{VLM})$ **then**
    break;
  end
  else
    $\mathcal{A} \leftarrow \mathcal{A} \cup \{\mathbf{a}^k\}$, $\mathcal{S} \leftarrow \mathcal{S} \cup \{\mathbf{s}^k\}$;
  end
end
return $\mathbf{a}^k$;

object in I_0. We reconstruct complete triangle meshes for each object using a pretrained image-to-3D model [8]. We prompt the VLM to automatically select different physics engines based on object characteristics: MuJoCo [9] for rigid bodies, a variant of the projective dynamics [10] solver for stiff deformable objects that ensures numerical stability, and the Material Point Method [11] solver for soft objects to handle potential topological changes. We automate physical parameters θ_{phys} inference by prompting the VLM to leverage its commonsense reasoning for plausible predictions, following prior works [12], [13], [14].

B. Action Planning via Simulation-enabled VLM

Given the constructed simulator SIM, our action planning framework follows an iterative refinement process, as outlined in Fig. 3. As shown in Alg. 1, our planner takes as input the initial RGB-D observation I_0, the initial simulator state s_0, task description ℓ_{task}, VLM, and SIM. The planner begins by sampling an initial set of action sequences $\mathcal{A}$ from the VLM prior. For each action sequence $\mathbf{a}^i \in \mathcal{A}$, the SIMROLLOUT procedure iteratively applies each action a_t^i and uses the SIM function to obtain the next state s_{t+1}^i, adding simulation rollouts $\mathbf{s}^i \in \mathcal{S}$.

After initialization, each iteration proceeds as follows. Using both $\mathcal{A}$ and $\mathcal{S}$, a VLM-based optimizer refines the proposed action sequences and produces a new action sequence

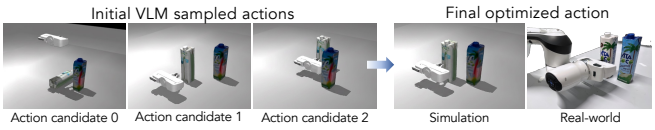

Fig. 4: **Action optimization process.** We show an example from the *non-toppling push* task. The left three images show simulation rollouts from initial VLM-sampled action sequence proposals, all of which fail due to insufficient/overshooting push, or because the bottle topples. From these proposals, the VLM optimizer reasons a non-trivial action update that pushes the bottle for the correct distance without toppling in both simulation and real-world execution.

$\mathbf{a}^k$. Based on the simulated rollout $\mathbf{s}^k$, the VLM model then evaluates whether $\mathbf{a}^k$ achieves the task goal. If successful, the corresponding action sequence $\mathbf{a}^k$ is executed on the real robot, and the process terminates. Otherwise, the planner proceeds to another round of action optimization by adding a newly generated $\mathbf{a}^k$ to $\mathcal{A}$ and $\mathbf{s}^k$ to $\mathcal{S}$, until either a successful plan is found or the maximum iteration limit $K_{\max}$ is reached. We leverage the pretrained knowledge of VLM to instantiate the SAMPLE, OPTIMIZE, and TASKSUCCESS modules. For each role, we define a corresponding system prompt ℓ_*, where $*$ denotes sample, opt, or eval, specifying the function that the VLM performs.

III. EXPERIMENTS

To evaluate the effectiveness of our framework, we design seven challenging, real-world, physics-aware, fine-grained manipulation tasks. We assess whether our method enables zero-shot planning on these tasks, comparing it against other state-of-the-art zero-shot methods. We evaluate our system using a Franka Research 3 robot arm with a parallel-jaw gripper. A single calibrated Intel RealSense D435i RGBD camera is used.

Tasks and Metrics. We design diverse tasks requiring fine-grained, physics-aware manipulation planning. The objects span rigid bodies (cartons, bowls, boxes) to deformable materials (rope, Play-Doh), enabling evaluation across different physical properties and manipulation strategies, including pushing, grasping, pivoting, squeezing, and sweeping. Success rate is our primary evaluation metric.

Baselines. We compare our approach against the following baselines: (1) *VLA models* that are trained on large-scale robot action datasets to directly predict joint velocities from visual observations and language instructions. We use $\pi_{0.5}$ [15], a recent open-source VLA model pretrained on the largest available robot manipulation dataset, as a representative baseline. (2) *VLM-based methods* that leverage geometric representations to augment VLM for manipulation planning. We compare against VoxPoser [16], which uses volumetric value maps to represent spatial affordances in 3D, and MOKA [17], which predicts keypoints and affordance regions to generate manipulation actions. For our pushing and squeezing scenarios, we extend MOKA to represent contact location with a target contact point and infer contact direction from pre- and post-contact positions.

TABLE I: **Success rates of our method and baselines.** For each task, we run 10 trials per method. Our approach consistently achieves a substantially higher success rate than baselines.

Method	Non-toppling push	Bowl stacking	Pivoting	Shape rope	Shape dough	Avoid obstacle	Sweeping
$\pi_{0.5}$[15]	0%	0%	0%	0%	0%	0%	0%
VoxPoser[16]	0%	20%	0%	0%	0%	0%	20%
MOKA[17]	0%	10%	0%	20%	0%	0%	0%
Ours	**80%**	**60%**	**40%**	**90%**	**80%**	**80%**	**70%**

Fig. 5: **Qualitative results.** The figure shows the initial state, execution progress, and final state for four tasks in both the real world (top) and the simulation (bottom). By leveraging VLM's powerful generalization, rendered simulation images can guide VLM's test-time reasoning for action planning despite the visual sim2real gap.

Results Table I shows the success rates of our method in comparison with baseline approaches. Overall, our method consistently outperforms baseline methods across all evaluated tasks, highlighting its strong performance in challenging tasks that require fine-grained, physics-aware manipulation. Fig. 5 shows simulation and real-world rollouts of six tasks. From the table, the VLA model $\pi_{0.5}$ consistently fails on all tasks. While we observe that $\pi_{0.5}$ can sometimes generate actions that approach the target object, it fails to complete the manipulation. This is because while VLA models can perform zero-shot on tasks similar to those seen during training, they generalize poorly to out-of-domain, challenging tasks used in our experiment. VLM-based methods, VoxPoser and MOKA, leveraging VLM's strong scene-understanding and reasoning capabilities, achieve non-zero success rates on tasks such as *bowl stacking*, *shape rope* and *sweeping*. However, they struggle with tasks that require precise action planning, where small errors, such as pushing the wrong part of an object (in *non-toppling push*) or squeezing an incorrect region of deformable materials (in *shape dough*) lead to failures. In contrast, our method integrates simulation-enabled reasoning with VLMs, enabling the robot to iteratively refine its action plan using simulation rollouts as context. This enables the system to identify and avoid physically unstable or ineffective strategies. For example, in *non-toppling push*, the simulation shows that pushing near the top of the carton would cause toppling, so the system adapts by pushing from a more stable point, as shown in Fig. 4.

IV. Conclusion

We introduce **SIMPACT**, a novel action-planning framework that leverages simulation-enabled VLM to enable zero-shot robotic manipulation without any task-specific training. Our approach is made possible by a foundation-model-enabled simulation construction pipeline and a test-time VLM reasoning framework that together unlock the rich commonsense knowledge and reasoning capabilities of VLMs for physics-aware, fine-grained robotic manipulation. Real-world experiments demonstrate that SIMPACT provides substantial improvements over state-of-the-art general-purpose manipulation models.

REFERENCES

[1] OpenAI, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[2] G. Comanici, E. Bieber, M. Schaekermann, I. Pasupat, N. Sachdeva, I. Dhillon, M. Blistein, O. Ram, D. Zhang, E. Rosen, *et al.*, "Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities," *arXiv preprint arXiv:2507.06261*, 2025.

[3] D. Driess, F. Xia, M. S. M. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, W. Huang, Y. Chebotar, P. Sermanet, D. Duckworth, S. Levine, V. Vanhoucke, K. Hausman, M. Toussaint, K. Greff, A. Zeng, I. Mordatch, and P. Florence, "Palm-e: An embodied multimodal language model," in *arXiv preprint arXiv:2303.03378*, 2023.

[4] T. Ren, S. Shen, and the IDEA-Research team, "Grounded-sam-2: Ground and track anything in videos," 2025, accessed: 2025-11-12.

[5] G. Williams, N. Wagener, B. Goldfain, P. Drews, J. M. Rehg, B. Boots, and E. A. Theodorou, "Information theoretic mpc for model-based reinforcement learning," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 1714–1721.

[6] J. B. Rawlings, D. Q. Mayne, M. Diehl, *et al.*, *Model predictive control: theory, computation, and design*. Nob Hill Publishing Madison, WI, 2020, vol. 2.

[7] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan, Z. Zeng, H. Zhang, F. Li, J. Yang, H. Li, Q. Jiang, and L. Zhang, "Grounded sam: Assembling open-world models for diverse visual tasks," 2024.

[8] T. H. Team, "Hunyuan3d 2.1: From images to high-fidelity 3d assets with production-ready pbr material," 2025.

[9] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 5026–5033.

[10] S. Bouaziz, S. Martin, T. Liu, L. Kavan, and M. Pauly, "Projective dynamics: fusing constraint projections for fast simulation," *ACM Trans. Graph.*, vol. 33, no. 4, July 2014. [Online]. Available: https://doi.org/10.1145/2601097.2601116

[11] C. Jiang, C. Schroeder, J. Teran, A. Stomakhin, and A. Selle, "The material point method for simulating continuum materials," in *ACM SIGGRAPH 2016 Courses*, ser. SIGGRAPH '16. New York, NY, USA: Association for Computing Machinery, 2016. [Online]. Available: https://doi.org/10.1145/2897826.2927348

[12] W. Xie, M. Valentini, J. Lavering, and N. Correll, "Deligrasp: Inferring object properties with LLMs for adaptive grasp policies," in *8th Annual Conference on Robot Learning*, 2024. [Online]. Available: https://openreview.net/forum?id=rY5T2aIjPZ

[13] H. Xia, Z.-H. Lin, W.-C. Ma, and S. Wang, "Video2game: Real-time interactive realistic and browser-compatible environment from a single video," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 4578–4588.

[14] B. Chen, H. Jiang, S. Liu, S. Gupta, Y. Li, H. Zhao, and S. Wang, "Physgen3d: Crafting a miniature interactive world from a single image," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2025, pp. 6178–6189.

[15] K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. R. Equi, C. Finn, N. Fusai, M. Y. Galliker, D. Ghosh, L. Groom, K. Hausman, b. ichter, S. Jakubczak, T. Jones, L. Ke, D. LeBlanc, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, A. Z. Ren, L. X. Shi, L. Smith, J. T. Springenberg, K. Stachowicz, J. Tanner, Q. Vuong, H. Walke, A. Walling, H. Wang, L. Yu, and U. Zhilinsky, "$\pi_{0.5}$: a vision-language-action model with open-world generalization," in *Proceedings of The 9th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, J. Lim, S. Song, and H.-W. Park, Eds., vol. 305. PMLR, 27–30 Sep 2025, pp. 17–40. [Online]. Available: https://proceedings.mlr.press/v305/black25a.html

[16] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei-Fei, "Voxposer: Composable 3d value maps for robotic manipulation with language models," in *7th Annual Conference on Robot Learning*, 2023. [Online]. Available: https://openreview.net/forum?id=9_8LF30mOC

[17] K. Fang, F. Liu, P. Abbeel, and S. Levine, "Moka: Open-world robotic manipulation through mark-based visual prompting," *Robotics: Science and Systems (RSS)*, 2024.