IEEE International Conference on Robotics and Automation (ICRA), Atlanta, USA, 2025.

Prompt-responsive Object Retrieval with Memory-augmented Student-Teacher Learning

Malte Mosbach and Sven Behnke

***Abstract*— Building models responsive to input prompts represents a transformative shift in machine learning. This paradigm holds significant potential for robotics problems, such as targeted manipulation amidst clutter. In this work, we present a novel approach to combine promptable foundation models with reinforcement learning (RL), enabling robots to perform dexterous manipulation tasks in a prompt-responsive manner. Existing methods struggle to link high-level commands with fine-grained dexterous control. We address this gap with a memory-augmented student-teacher learning framework. We use the Segment-Anything 2 (SAM 2) model as a perception backbone to infer an object of interest from user prompts. While detections are imperfect, their temporal sequence provides rich information for implicit state estimation by memory-augmented models. Our approach successfully learns prompt-responsive policies, demonstrated in picking objects from cluttered scenes. Videos and code are available at `https://maltemosbach.github.io/promptable_object_manipulation/`**

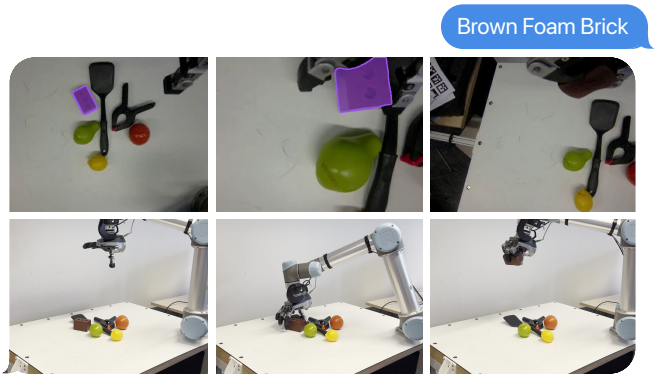


Fig. 1: We present a student-teacher framework that enables learning interactive, prompt-responsive policies for object-retreival from cluttered scenes (Figure inspired by [13]).

I. Introduction

Foundation Models (FMs) such as GPT-4 [1] and Segment Anything [2] represent a paradigm shift in the field of artificial intelligence. Trained on broad web-scale datasets, these models excel in generating contextually nuanced outputs across a diverse array of tasks [3]. This capability is typically implemented through prompt engineering, where human understandable inputs are used to prompt the model for a valid response to the task at hand [2], [4]. Thus, simple instructions can be used to condition a model to perform a myriad of downstream tasks.

Being able to control the behavior of dexterous robots in a similar manner is a long-standing goal [5], yet existing approaches mainly leverage FMs for high-level planning [5]–[7].. While this approach has yielded impressive capabilities in terms of developing versatile agents, it falls short in replicating the intricate, low-level dexterity required for complex manipulation tasks, such as dexterous grasping from clutter. It remains unclear how such approaches can scale to match the intuitive dexterity humans exhibit — often relying on tacit, hard-to-describe skills. [8].

In contrast, reinforcement learning (RL) bypasses the need for explicit, high-level instructions, opting instead to learn behaviors through trial-and-error. Recent works demonstrate that RL is capable of learning fine motor behaviors comparable to human dexterity [9]–[12], yet the learned policies tend to be task-specific and lack the adaptability of prompt-conditioned models.

Consider the scenario of a warehouse robot tasked with fulfilling an order by picking specific items from cluttered bins. Retraining for each new object or providing explicit models for every item to grasp is neither practical nor scalable. Instead, we aim to operate such a system using abstract, human-understandable instructions, like a packing list of items to retrieve. However, existing methods fail to integrate intuitive, language-guided instructions with the low-level control needed for dexterous manipulation.

To bridge this gap, we propose a novel approach that integrates the broad, open-vocabulary capabilities of vision foundation models (VFMs) with the precise motor control developed through RL. Specifically, we leverage the Segment Anything 2 (SAM 2) model as a perception backbone to segment objects of interest based on user prompts. While the representations generated by SAM 2 are inherently imperfect – prone to issues like occlusions or unstable segmentations – the sequence of outputs over an episode is highly informative and allows for an implicit estimation of the true object state.

This type of implicit inference of an underlying state from impaired perception via memory-augmented agents has recently been shown to be a powerful tool in navigation of simulated agents [14] and quadruped robots [15]. While prior works utilized synthetic simulations of imperfect perception, allowing them to train RL agents with high performance, having the SAM 2 model in the RL training loop pushes the compute requirements beyond what is currently feasible. Instead, we factor out learning how to act and learning

All authors are with the Autonomous Intelligent Systems Group, University of Bonn, Germany and the Lamarr Institute for ML and AI, Bonn, Germany. The corresponding author is Malte Mosbach `mosbach@ais.uni-bonn.de`

to infer the underlying state of an object of interest via a student-teacher formulation. After a teacher policy has been trained to master the task from privileged, simulator-state information, we distill its knowledge to a memory-augmented student policy that operates based on imperfect outputs from our VFM. This distillation process forces the student to implicitly learn the true state of the object to imitate the teacher's actions, even when faced with incomplete or faulty perception data. We explore both LSTMs and Transformers as sequence processing modules, to understand the impact of history-awareness on the student's performance.

Our key insight is that while SAM 2 does not allow for reconstructions of Markovian, ground-truth states directly that are needed to deploy a privileged policy, learning to match the teacher's actions forces the student to implicitly learn the associations between the detection sequence and the current state of the object of interest.

Our results demonstrate that this approach successfully learns dexterous, prompt-responsive policies capable of generating complex, targeted manipulation in cluttered environments. Further, the observation-space of the student policy allows for zero-shot real-robot transfer. By harnessing the synergy between high-level instruction and low-level robot action, transforming the warehouse robot into a system that picks out recyclable cans from trash or selectively removes rotten fruits from a box of produce can be achieved simply through conditioning on human-understandable prompts.
In summary, we address the following research questions:

1) *Can VFMs be used as perception backbones for prompt-responsive manipulation policies?* Yes, we find that the proposed method yields agents that are highly effective at grasping a wide variety of objects on the basis of human-understandable prompts.
2) *What are the necessary mechanisms for this approach?* The two core mechanisms are first, keeping the FM out of the RL loop via student-teacher learning, and second, using history-aware architectures to implicitly infer underlying states from imperfect detections.
3) *Does this method transfer to real-robot systems?* Yes, we demonstrate that the strong performance of the policies in simulation transfers to our real-robot system.

II. Related Work

A. Learning-based Robotic Grasping

Reliable robotic grasping has been a prominent challenge in robotics for decades [16]. The control aspect of this problem has been formulated mainly as either (1) a problem of grasp-pose prediction or (2) as closed-loop continuous control.

Grasp-pose prediction is typically solved through supervised learning, where a model is trained to predict the best grasp pose for a given object. Redmon and Angelova [17] introduced a deep learning approach to predict grasp positions on objects using convolutional neural networks.

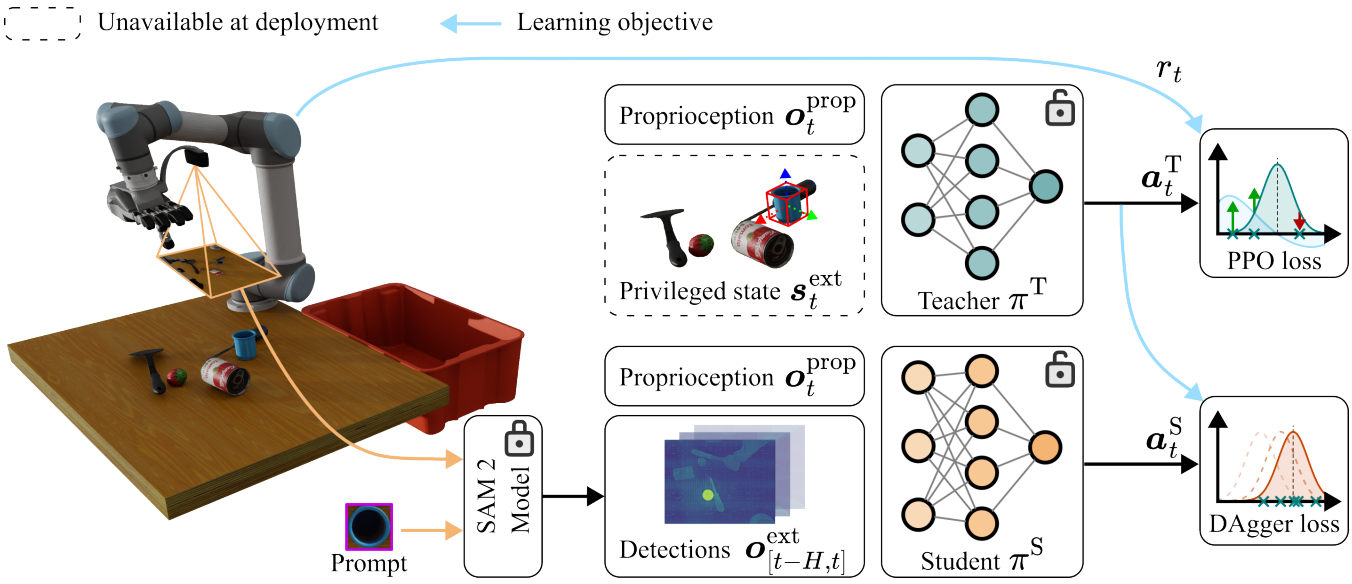
The latter group of continuous-control approaches typically utilize reinforcement learning. Levine et al. [18] demonstrated the use of deep RL for end-to-end training of robotic grasping policies. More recently, Kalashnikov et al. [19] developed QT-Opt, a scalable RL algorithm that significantly improves grasping performance. Mosbach et al. [20] utilize the Segment-Anything model for prompt-based robotic grasping. However, they consider only tabletop grasping and rely on unobstructed tracking of the object throughout the episode. Our approach eliminates this requirement, allowing target objects to go out of view, be occluded, or be misdetected by the model.

B. Learning from Privileged Information

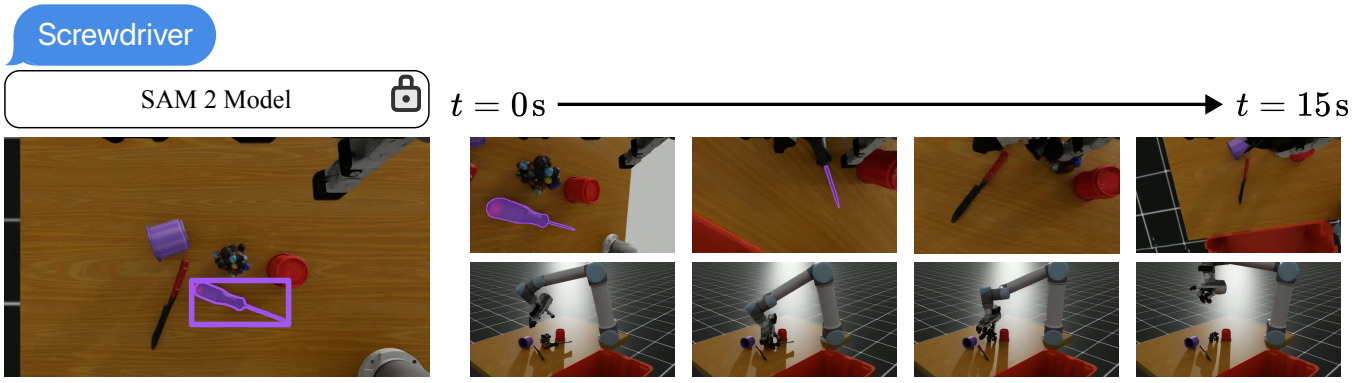
Chen et al. [21] observed that learning to imitate expert drivers from visual perception conflates two difficult problems: learning to perceive the environment and learning to control the vehicle. They proposed a two-stage approach where a teacher policy is trained to imitate the expert's actions based on the environment's ground-truth state. The knowledge from the teacher is then distilled into a vision-based student policy. Recently, Chen et al. [9], [10] extended this strategy to reinforcement learning. In their approach, a teacher policy is trained using simulator state information to solve the control problem. Subsequently, a visual student policy is trained more efficiently in a supervised manner to imitate the teacher from realistic observations. Kumar et al. [22] identified domain randomization parameters as a special kind of privileged information. Their method, Rapid Motor Adaptation, enables a student policy to infer domain parameters from observation history, which are made available to the teacher policy. Building on this framework, Margolis et al. [23] developed a method to learn agile locomotion over diverse terrains. The domain randomization parameters are made available to the teacher policy, to avoid learning overly conservative behaviors. The student policy, which cannot access these parameters directly, utilizes a history-aware approach, $\pi_S(\boldsymbol{x}_t, \boldsymbol{x}_{[t-h:t-1]})$, to implicitly deduce the domain parameters from the observation history. Zhang et al. [15] recently utilized an asymmetric actor-critic architecture to achieve resilient navigation. In their method, the critic has perfect observability of obstacles, while the history-aware actor uses exteroception along with an imperfect map for navigation. This approach enables policies to learn about unknown obstacles by integrating their history of observations from interactions such as bumping into or touching them. Similar to our approach, Kumar et al. and Margolis et al. [22], [23] use student-teacher learning to overcome a problem of explicit versus implicit observability. While their focus is on domain randomization, we tackle the problem associating imperfect sequences of detections with the underlying state of the scene.

C. Prompt-Guided Manipulation

Recent research efforts strive to transfer grounded world knowledge from vision-language models to robotics by training large foundation models on a wide array of behavioral data [3], [24], [25]. Additionally, language models have been employed for high-level planning in robotic manipulation [5],



(a) Training in simulation is performed in two stages via student-teacher learning.



(b) At deployment, the student policy can be prompted with points or language to infer the target object.

Fig. 2: We propose to train prompt-guided policies in two stages. First, the teacher policy is trained with model-free RL to solve the control problem from privileged information s_t^{ext}. Thereafter, the student policy is trained to imitate the teacher without access to s_t^{ext}, forcing it to implicitly infer the object state from the history of visual observations $o_{[t-H,t]}^{\text{ext}}$.

[24]. Recently, Shen et al. [13] distilled knowledge from language-supervised image models into a 3D representation of the scene, enabling the learning of 6-DOF grasping and placing of novel objects in a language-guided manner from only a few demonstrations. In contrast, our work does not rely on learning a representation of one specific scene beforehand.

III. Method

A. Overview

Consider a warehouse robot tasked with fulfilling an order by picking specific items from cluttered bins. At the start of each trial, the robot receives a description of the item to retrieve, which may be provided as an open-vocabulary text prompt, a selected point, or a detected bounding box. The robot uses this instruction to locate and grasp the specified object.

Successfully executing this task requires addressing two key challenges: (1) learning to interpret the provided instruction and visual observations, and (2) mastering the dexterous control needed to grasp objects in clutter. Instead of tackling both challenges simultaneously, we follow recent work in imitation and reinforcement learning [9], [10], [21] and decompose the problem into two stages. We first train a teacher policy using privileged simulator-state information, enabling it to acquire effective control strategies in an idealized setting. We then transfer these behaviors to a student policy that operates solely on real-world inputs. To bridge the gap between noisy perception and reliable control, we employ *memory-augmented student-teacher learning*, allowing the student to integrate outputs from perception modules like SAM 2 — despite occlusions and segmentation errors — into an implicit understanding of the object state.

B. Observations

To train our prompt-responsive grasping policies, we utilize three types of observations: proprioception, privileged exteroception, and VFM-based exteroception (see Table I).

Proprioception: Proprioception encompasses information that is available in both simulation and real-world deployment. This includes the joint states of the robotic arm and hand, the most recent action taken, and the 3D goal position for the target object.

Privileged Exteroception: During privileged training, the teacher policy has access to additional information unavailable in real-world deployment. This includes the oriented bounding box (OBB) of the target object, as well as a privileged heightmap centered around the gripper, which provides a structured representation of the surrounding clutter. Moreover, we provide additional information about the state of the manipulator including the fingertip poses and velocities.

VFM-based Exteroception: The student policy receives object detections from a VFM (SAM,2 in our case) to provide a prompt-responsive visual input space. Since these detections are imperfect and non-Markovian — due to occlusions and misdetections — we provide the student model with a history of recent detections, allowing it to infer the true object state over time.

C. Learning from Privileged Information

We train the teacher policy π^{T} using RL, where the agent maps observations $\boldsymbol{o}_t^{\mathrm{T}}$ to actions $\boldsymbol{a}_t$. To ensure generalization across diverse objects, we design the simulation as a multi-task learning problem, where each parallel environment contains a randomly selected subset of training objects. As a result, the teacher policy must learn to handle objects of varying shapes and sizes—both as targets and obstacles. To optimize the teacher policy, we require a stable RL algorithm capable of handling challenging continuous control tasks. We use PPO [26], which optimizes the policy to maximize the expected return:

$$J(\pi^{\mathrm{T}}) = \mathbb{E}_{\pi^{\mathrm{T}}}[\sum_{t=0}^{\infty} \gamma^t R(\boldsymbol{o}_t^{\mathrm{T}}, \boldsymbol{a}_t)]. \tag{1}$$

TABLE I: Observations combine robot proprioception with privileged or visual exteroception for the teacher and student, respectively.

Term	Dimensionality
Last actions	$11D$
Arm joint state	$18D$
Hand joint targets	$11D$
Goal position	$3D$
Fingertip poses	$35D$
Fingertip velocities	$30D$
Target object OBB corners	$24D$
Heightmap	$64D$
Target object velocity	$6D$
Target to goal pos	$3D$
SAM2 detected point-cloud	$4D * N_{points}$

Although the agent has access to privileged simulator-state information, the observation $\boldsymbol{o}_t^{\mathrm{T}}$ at a single time-step t does not convey the full state information, such as the exact shape of an object. Hence, we evaluate the use of LSTM architectures [27] alongside a standard MLP policy to enable the teacher policy to consider temporal dependencies for decision-making. The MLP policy comprises three layers with 768, 512, and 256 units, respectively. The LSTM variant adds a single LSTM layer with 768 units before the MLP.

The teacher policy observes regular proprioception alongside the privileged simulator-state information. The detailed makeup of the observation-space is given in Table I.

The policy controls the robot's joints at a frequency of 10 Hz. We use an exponential moving average (EMA) to control the joint velocities of the arm, formulated as $\dot{\boldsymbol{q}}_{t+1}^{\text{target}} = \alpha \boldsymbol{a}_t + (1-\alpha)\dot{\boldsymbol{q}}_t^{\text{target}}$, balancing smoothness and responsiveness. The Schunk SIH hand is controlled via servo-actuated tendons. A similar EMA formulation is used to set the servo target positions as $\boldsymbol{p}_{t+1}^{\text{target}} = \alpha \boldsymbol{a}_t + (1-\alpha)\boldsymbol{p}_t^{\text{target}}$.

Rewards and Termination Conditions: The reward function is designed to facilitate directed exploration without distracting from the overall objective. Initially, the agent is motivated to move its hand closer to the target object, where Δd^{grab} denote the change in distance between the fingertips and the object of interest. Once the agent reaches the object, this reward term is exhausted, allowing the agent to focus on manipulating the object. At this stage, we reward the agent for lifting the object from the table or bin, where h_t represents the height of the object as measured by the lowest point of its OBB. Finally, the agent is rewarded for moving the target object to the goal position and for reaching the goal position within a small threshold. The detailed reward function is shown in Table II.

Notably, we opted not to include reward terms that directly encourage safe behavior, such as contact or action penalties, as they can interfere with task-relevant exploration in hand-arm manipulation tasks. Our experiments showed that imposing penalties for hard contacts significantly hindered effective exploration. Instead, we employ termination conditions to enforce safety, which offers several advantages. Firstly, this eliminates the trade-off between task reward and

TABLE II: Reward terms used to train the teacher policy.

Term	Equation	Weight
Alive	1.0	0.01
Grab object	$-\Delta d^{\text{grab}}$	10.0
Lift object	$\min(h_t, h^{\text{lifted}})$	40.0
Reach goal	$-\Delta d^{\text{goal}}$	100.0
Goal bonus	$\mathbb{1}(d_t^{\text{goal}} < \bar{d}^{\text{goal}})$	10.0

TABLE III: Termination conditions, where $\mathcal{A}$ and $\mathcal{T}$ denote the bodies of the robot arm and the table (and bin), respectively.

Term	Condition
Arm contacts	$\max_{i \in \mathcal{A}} \|\boldsymbol{c}_t^i\|_2 > 5.0$
Tabletop or bin contacts	$\max_{i \in \mathcal{T}} \|\boldsymbol{c}_t^i\|_2 > 25.0$
Time-out	$t > T_{max}$

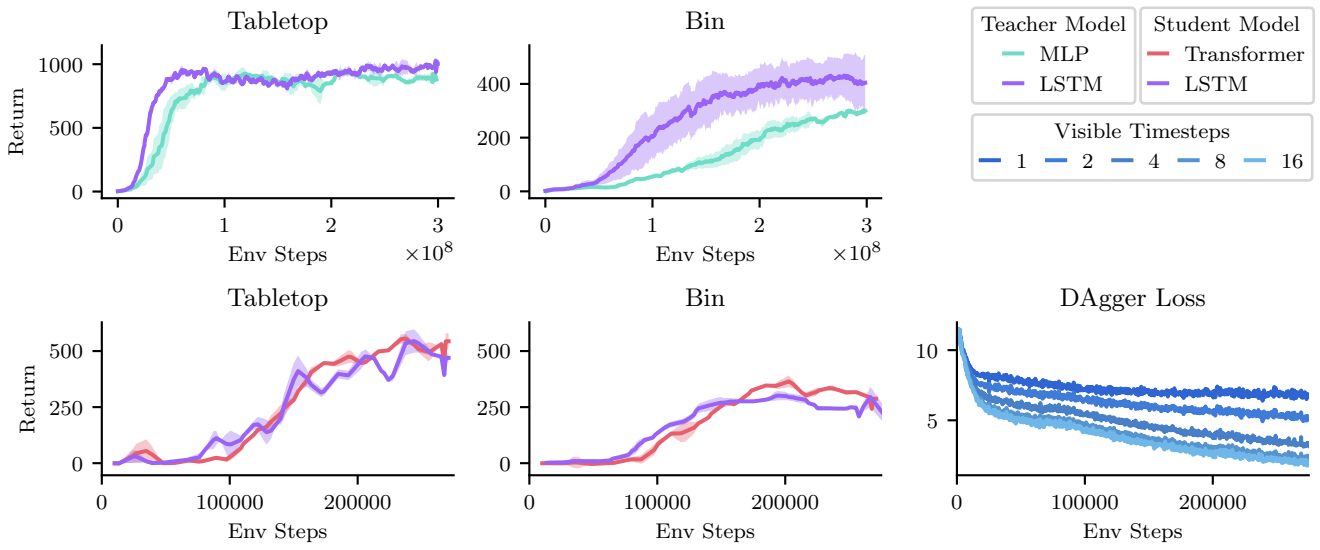


Fig. 3: Training performance of teacher policies (top) and student policies (bottom).

safety penalties, ensuring that unsafe behaviors are prohibited outright. Secondly, using terminations to punish unsafe behaviors creates a natural learning curriculum. Initially, when the agent expects to obtain low rewards, terminating an episode due to collision is less significant. As the agent improves and anticipates higher rewards, it will avoid these terminations due to the higher cost of lost future rewards. The termination conditions are shown in Table III.

D. Memory-augmented Student-Teacher Learning

We apply a memory-augmented student-teacher learning framework to transfer knowledge from the trained teacher policy to a student policy that operates solely on proprioception and detections from a VFM. SAM 2 understands the relationship between user prompts and objects. Using its detections as a prompt-responsive observation space removes the need for the agent to learn object-prompt associations explicitly. Instead, the student policy learns to interpret these detections as part of its perception pipeline.

The key challenge is that SAM 2's detections are non-Markovian — occlusions and segmentation inconsistencies can cause missing or unstable detections. Thus, for the student policy to successfully imitate the teacher, it must implicitly infer the true state of the target object by integrating past detections with object dynamics. This motivates the use of history-aware architectures, which allow the student to compensate for gaps in perception.

While our final policy should be controllable via user prompts, training requires automated prompting to ensure efficiency and consistency across parallelized simulation environments. To achieve this, we introduce two key extensions. First, we modify SAM 2 to handle batched image streams, enabling efficient processing across multiple parallel environments in Isaac Sim. This allows detections to be generated in parallel rather than sequentially, significantly reducing computational overhead. Second, we automate prompt generation by leveraging ground-truth object geometry from simulation. Specifically, we extract points on the mesh surface of the target object and project them into the camera frame. From these projected points, we compute a tight bounding box in the image space, which is then used as the prompt for SAM 2, ensuring alignment between the model's detections and the true target object defined in the task.

To process these detections, we transform the depth image from the RGB-D camera into a point cloud, providing a direct 3D representation of objects in the environment. Additionally, we append a feature dimension that indicates which subset of points SAM 2 assigns to the target object. For encoding these point clouds, we employ a PointNet-like encoder, which extracts an embedding of the scene. This embedding is then concatenated with proprioceptive observations and passed to the policy network. To integrate temporal information, we evaluate the follwoing memory-based student architectures within the DAgger imitation learning framework [28].

1D-CNN: Temporal convolution has been successfully applied in prior work [22] as a lightweight approach to capturing short-range dependencies. Here, we apply a three-layer 1D-CNN that convolves the feature representations across the time dimension to extract relevant temporal correlations The CNN layers use the following input channels, output channels, kernel size, and stride: [256, 256, 8, 4], [256,256, 5, 1], [256, 256, 5, 1].

LSTM: The LSTM variant mirrors the recurrent teacher policy, consisting of a single-layer LSTM of size 768..

Transformer: Given their proven effectiveness in capturing long-range dependencies in sequential data, test the use of a transformer encoder model to model. This transformer comprises 4 layers, each with 8 heads and a hidden dimension of 256.

The output from each of the sequence models is passed through a three-layer MLP to produce the action distribution.

TABLE IV: Simulation results with success and collision rate (SR/CR).

State	Model	Tabletop		Bin	
		SR ↑	CR ↓	SR ↑	CR ↓
Privileged	MLP	84.6 ± 4.0	2.5 ± 0.2	45.7 ± 5.2	5.6 ± 0.5
	LSTM	91.2 ± 0.9	1.2 ± 0.1	63.5 ± 5.7	5.1 ± 0.8
Visual	LSTM	87.1 ± 1.2	2.1 ± 0.2	59.0 ± 1.4	5.9 ± 0.9
	Transformer	88.3 ± 1.1	2.0 ± 0.2	59.2 ± 1.3	5.9 ± 0.8

TABLE V: Real-robot results evaluated for the tabletop scenario with different numbers of train or test objects present (n_{obj}).

Model	Train (n_{obj})		Test (n_{obj})	
	3	5	3	5
LSTM	$6/10$	$6/10$	$6/10$	$5/10$
Transformer	$4/10$	$6/10$	$6/10$	$5/10$

IV. Experimental Setup

A. Environments

We evaluate our method in simulated multi-object manipulation tasks using Nvidia Isaac Lab [29]. The simulation consists of multiple parallel instances of our robotic system interacting with randomly selected YCB objects [30]. In total, we use 60 YCB objects, of which 48 are included in the training set, while 12 are held out to assess generalization.

B. Evaluation and Metrics

Performance is measured based on two success criteria: lifting the target object from the tabletop or bin and moving it to a specified 3D goal position, which is considered successful if the object is within 5 cm of the target location.

V. Results

We deployed our trained controllers on a variety of unseen objects, including items of diverse shapes and sizes, such as a golf ball, a fork, and a cleanser bottle. Notably, the RL policies exhibited robust yet nimble behaviors. Despite the inherent safety challenges in deploying RL, our approach of terminating on unsafe interactions resulted in unrestricted yet careful manipulation. Additionally, the policies demonstrated implicit inference of the target object state over time, allowing for successful manipulation even when the target moves out of view or is not detected by SAM 2.

A. Grasping from Tabletop Scenes

First, we present results for the task of grasping a target object from a tabletop. In cluttered environments, a proficient control policy should reliably grasp the target item while ensuring safe operation. To evaluate these characteristics, we measured the success rate and collision rate, defined as the percentage of episodes where the robot induced undesirable collisions, as shown in Table IV. The results show that LSTM teacher outperforms the MLP variant, with the best LSTM policy achieving a 92.4% success rate averaged over all 48 training objects. In 1.2% of the episodes, the policy induced contacts that were larger than our desired threshold.

To identify failure modes, we manually reviewed rollouts and found that most failures involved objects that were difficult to grasp, such as thin, elongated items like knives, which require precise handling to avoid excessive contact forces. Failures also occurred when smaller objects were covered by larger ones. While we observed some meaningful pre-grasp behaviors, such as reorienting objects or pushing obstacles aside, long-horizon strategies – like intentionally moving a non-target object before retrieving the target – remain difficult to learn. A curriculum learning approach that scales the number of items over time as the policy improves might be an interesting avenue for future work. We depict the training progress on the left of Figure 3. Overall, we were able to learn highly effective teacher policies, that successfully handle diverse tabletop configurations. Failures are mainly limited to particularly difficult scenarios or slight exceedances of contact thresholds for hard-to-grasp items.

In Table IV, we can see that the visual student policies are able to recover much of the teacher's performance. This indicates, that the formulation via supervised learning from a sequence of imperfect observations is an effective approach to transfer the policy's abilities to a real-world deployable observation space. On real-robot deployment (see Table V), we observed that the policies exhibited similar success rates on seen and unseen objects, indicating, that the learned behaviors are not overfitting to the training subset.

B. Grasping from Cluttered Bins

The previous results demonstrate that, for tabletop scenes, RL policies can learn to grasp objects from cluttered environments with high success rates. We aim to extend this capability to the more challenging scenario of picking objects from cluttered containers. This setup presents additional challenges, such as tightly packed heaps causing unforeseen interactions and the need for precise maneuvering to avoid collisions with the container walls. Figure 3 center shows the training process.

The teacher policies for the bin-picking scenario can learn careful grasping of diverse items, but underperform the tabletop policies by a substantial margin (see Table IV). The learned policies again exhibit desirable behaviors like re-grasping and pre-grasp manipulation. Inspecting the learned behaviors revealed, that the complex kinematics required to manouver the arm to the desired object without causing collisions are most difficult to learn. The policies cause more terminations due to contacts and tend to behave overly conservative on objects that are difficult to grasp. The student policies for this scenario again track the performance of the teachers closely, indicating that the room for improvement lies in learning better teacher policies. Combining RL with explicit safety constraints is a promising avenue for future work in this scenario.

C. Implicit Inference through Time

We hypothesize that the student infers the underlying object state necessary for imitating the teacher by leveraging

the history of observations. If true, increasing the context length should enhance the student's ability to imitate the teacher's actions, resulting in lower loss. To investigate this, we log the loss curves of a history-aware student policy imitating an MLP teacher over different lengths of visible context in the bottom right of Figure 3. A clear pattern of decreasing loss with increasing context length is visible, showing that the student is better able to imitate the teacher when more context becomes available, which indicates that the student is indeed learning to infer the underlying object state from the history of observations.

VI. Conclusion

Summary: We have illustrated a way to condition RL policies on the output of SAM 2 to achieve closed-loop, prompt-guided grasping from clutter. Specifically, we have formulated the problem of learning from imperfect detections of a foundation model as a POMDP that can be solved efficiently through history-aware architectures in a student-teacher setting.

Limitations and Future Work: While we were able to create dexterous manipulation behaviors for cluttered bin-picking, the number of undesirable collisions is still higher than for the tabletop scenario, causing us to leave real-robot deployment for future work. Further, while we have tackled the problem of picking and repositioning unknown objects, it would be interesting to apply our methodology to additional manipulation tasks. Straightforward extensions might be reposing objects, or placing them in specific locations in the environment, such as on a shelf or in a container.

VII. Acknowledgment

This work has been funded by the German Ministry of Education and Research (BMBF), grant no. 01IS21080, project "Learn2Grasp: Learning Human-like Interactive Grasping based on Visual and Haptic Feedback".

References

[1] OpenAI, "GPT-4 technical report," *CoRR*, vol. abs/2303.08774, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2303.08774

[2] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," *arXiv preprint arXiv:2304.02643*, 2023.

[3] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid *et al.*, "RT-2: Vision-language-action models transfer web knowledge to robotic control," in *7th Annual Conference on Robot Learning*, 2023.

[4] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020.

[5] M. Parakh, A. Fong, A. Simeonov, T. Chen, A. Gupta, and P. Agrawal, "Lifelong robot learning with human assisted language planners," in *CoRL 2023 Workshop on Learning Effective Abstractions for Planning (LEAP)*, 2023.

[6] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch, "Language models as zero-shot planners: Extracting actionable knowledge for embodied agents," in *International Conference on Machine Learning (ICML)*. PMLR, 2022.

[7] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," *arXiv preprint arXiv:2204.01691*, 2022.

[8] H. L. Dreyfus, "From Socrates to expert systems: The limits of calculative rationality," *Bulletin of the American Academy of Arts and Sciences*, vol. 40, no. 4, 1987.

[9] T. Chen, J. Xu, and P. Agrawal, "A system for general in-hand object re-orientation," in *Conference on Robot Learning (CoRL)*. PMLR, 2022.

[10] T. Chen, M. Tippur, S. Wu, V. Kumar, E. Adelson, and P. Agrawal, "Visual dexterity: In-hand dexterous manipulation from depth," *arXiv preprint arXiv:2211.11744*, 2022.

[11] A. Petrenko, A. Allshire, G. State, A. Handa, and V. Makoviychuk, "Dexpbt: Scaling up dexterous manipulation for hand-arm systems with population based training," *arXiv preprint arXiv:2305.12127*, 2023.

[12] Y. J. Ma, W. Liang, G. Wang, D.-A. Huang, O. Bastani, D. Jayaraman, Y. Zhu, L. Fan, and A. Anandkumar, "Eureka: Human-level reward design via coding large language models," *arXiv preprint arXiv:2310.12931*, 2023.

[13] W. Shen, G. Yang, A. Yu, J. Wong, L. P. Kaelbling, and P. Isola, "Distilled feature fields enable few-shot language-guided manipulation," in *7th Annual Conference on Robot Learning (CoRL)*, 2023.

[14] E. Wijmans, M. Savva, I. Essa, S. Lee, A. S. Morcos, and D. Batra, "Emergence of maps in the memories of blind navigation agents," *AI Matters*, vol. 9, no. 2, 2023.

[15] C. Zhang, J. Jin, J. Frey, N. Rudin, M. E. Mattamala Aravena, C. Cadena, and M. Hutter, "Resilient legged local navigation: Learning to traverse with compromised perception end-to-end," in *41st IEEE Conference on Robotics and Automation (ICRA 2024)*, 2024.

[16] Z. Xie, X. Liang, and C. Roberto, "Learning-based robotic grasping: A review," *Frontiers in Robotics and AI*, vol. 10, 2023.

[17] J. Redmon and A. Angelova, "Real-time grasp detection using convolutional neural networks," in *IEEE international conference on robotics and automation (ICRA)*, 2015.

[18] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *The Journal of Machine Learning Research (JMLR)*, 2016.

[19] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke *et al.*, "QT-Opt: Scalable deep reinforcement learning for vision-based robotic manipulation," *arXiv preprint arXiv:1806.10293*, 2018.

[20] M. Mosbach and S. Behnke, "Grasp anything: Combining teacher-augmented policy gradient learning with instance segmentation to grasp arbitrary objects," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024.

[21] D. Chen, B. Zhou, V. Koltun, and P. Krähenbühl, "Learning by cheating," in *Conference on Robot Learning (CoRL)*. PMLR, 2020.

[22] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "Rma: Rapid motor adaptation for legged robots," *arXiv preprint arXiv:2107.04034*, 2021.

[23] G. B. Margolis, G. Yang, K. Paigwar, T. Chen, and P. Agrawal, "Rapid locomotion via reinforcement learning," *The International Journal of Robotics Research (IJRR)*, vol. 43, no. 4, 2024.

[24] A. Brohan, Y. Chebotar, C. Finn, K. Hausman, A. Herzog, D. Ho, J. Ibarz, A. Irpan, E. Jang, R. Julian *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," in *Conference on Robot Learning (CoRL)*. PMLR, 2023.

[25] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu *et al.*, "Rt-1: Robotics transformer for real-world control at scale," *arXiv preprint arXiv:2212.06817*, 2022.

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

[27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, 1997.

[28] S. Ross, G. J. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, ser. JMLR Proceedings, vol. 15, 2011.

[29] M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, R. Singh, Y. Guo, H. Mazhar, A. Mandlekar, B. Babich, G. State, M. Hutter, and A. Garg, "Orbit: A unified simulation framework for interactive robot learning environments," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, 2023.

[30] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The YCB object and model set: Towards common benchmarks for manipulation research," in *International Conference on Advanced Robotics (ICAR)*. IEEE, 2015.