Plug-And-Play Object-Centric Representations From “What” and “Where” Foundation Models

Junyao Shi*, Jianing Qian*, Yecheng Jason Ma*, Dinesh Jayaraman
University of Pennsylvania
{junys, jianingq, jasonyma, dineshj}@seas.upenn.edu

Abstract—**There have recently been large advances in the problem of segmenting unknown category objects in general images. To leverage these for improved robot learning, we propose a new framework for building object-centric representations (OCR) for robotic control. Building on theories of “what-where” representations in psychology and computer vision, we use segmentations from a pre-trained model to stably locate across timesteps, various task-relevant entities in the scene, capturing “where” information. To each such segmented entity, we apply other pre-trained models that build vector descriptions suitable for robotic control tasks, thus capturing “what” the entity is. Thus, our OCR is constructed by appropriately combining the outputs of off-the-shelf pre-trained models, with no new training. On various simulated and real robotic tasks, we show that imitation policies for robotic manipulators trained on our OCR perform better than prior OCRs that are typically trained from scratch, as well as the current state of the art unstructured representations.**

I. Introduction

One of the fundamental challenges of intelligence is how to represent and process the continuous and complex stream of sensory information that we receive from the world. The “what-where” representation theory [5, 12, 49] in cognitive science postulates that the brain uses different neural pathways to encode two types of information: “what” information, which refers to the identity, features, and properties of an entity; and “where” information, which refers to the location, direction, and distance of an entity. A growing literature on object-centric representations (OCRs) attempts to instantiate these ideas in artificial intelligence, commonly focusing on co-training the “what” and “where” pathways within a target domain.

We investigate an alternative, simpler route towards OCRs, paved by recent advances in adjacent disciplines. First, computer vision researchers have recently achieved dramatic advances on image segmentation [24, 63], the task of identifying groups of pixels that correspond to semantic objects and their parts. These pre-trained models can now reliably locate the discrete entities in in-the-wild images in arbitrary domains. Next, pre-trained unsupervised vector representation encoders have matured and are fast becoming the de facto standard descriptors of the contents of raw sensory inputs for downstream tasks in many domains: language and audio [2, 7, 11, 41], vision [11, 16, 41], and robotics [32, 33, 37, 55].

We propose to chain these foundation models together to create a new general-purpose pre-trained OCR for robot learning. Having located (“where”) the entity slots in an image observation with a pre-trained image segmentation model, we propose to describe the contents of each slot (“what”) with another pre-trained model, a control-aware unsupervised representation encoder.

We instantiate this plug-and-play OCR framework by picking two representative “where” and “what” foundation models: SAM [24] for segmentation, and LIV [33] for control-aware image representation, which have both individually been pre-trained on large and diverse datasets, and afterwards been shown to work well on many domains of interest. If the composite OCR, “SAM-LIV” inherits these generalization properties, it may be used off-the-shelf in arbitrary new tasks; see Figure 1 for a schematic overview.

Indeed, we evaluate SAM-LIV on unseen simulated and *real-world* robotic manipulation settings. We find that SAM-LIV not only provide better object representation than other OCR approaches, but more importantly, also enable significantly better policy learning compared to both pre-trained flat representations and representations learned in-domain from scratch. Through ablation studies, we show that our use of pre-trained models in both the “what” and “where” components are critical to achieving substantial gains. In addition, we demonstrate that policies trained with SAM-LIV outperforms prior representations on real-robot manipulation tasks, fully showcasing the practicality and generality of our plug-and-play object-centric representation for robotics paradigm.

II. Plug-And-Play OCRs For Robotic Manipulation

Towards chaining our “where” and “what” foundation models (SAM and LIV, respectively) into a useful representation for manipulation policy learning, there are three key questions to address: **where** are the relevant object regions, **what** are their contents, and how should **robots** act to accomplish manipulation tasks given such what-where object-centric representations.

A. The Where: Localizing and Assigning Objects to Slots

To go from SAM mask outputs to OCR slot masks l_i, we propose to match each mask to some pre-specified task-relevant object segments.

Specifying task-relevant objects in a reference image. Before we can begin to extract OCRs in a new domain, we collect some reference images, such as from the frames of expert demonstrations that may be required anyway to train a policy. We use these to compute the background mask following the procedure in [1]. Then, on one of these same reference

[*] Equal contribution

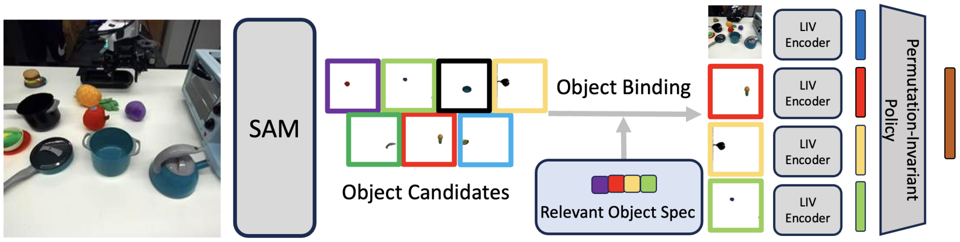


Fig. 1. **SAM-LIV**: plug-and-play object-centric representations for robotics by chaining "what" and "where" visual foundation models.

images o^{ref}, we run SAM to produce the set of segmentation masks m_i^{ref}. By design of SAM, m_i^{ref} is an over-complete set of segmentation masks corresponding to various entities i in the scene. This includes entities at various levels of granularity from objects all the way down to their parts and subparts, and also task-irrelevant entities in background regions of the scene.

To discard distractor entities among these reference image masks, we manually select k masks $\{l_1^{ref}, \dots, l_k^{ref}\}$ that most closely correspond to the task-relevant objects in the environment. In practice, this procedure requires very little annotation effort: about 1 minute for each environment. Note that this convenient specification interface is only made possible by the object representation.

Localizing task-relevant objects in each observation. Given these selected reference masks that encode task-relevant objects, the slots in our desired OCR must bind to these objects in each image. Towards this, we now overview a procedure to localize these objects.

- **Screening the object-level foreground entity candidates.** Given any new image observation o, we first identify background regions as described above. Then, we compute the SAM masks $\{m_i(o)\}_{i=1}^q$ and identify object-level foreground entities among these. We use a greedy non-maximum suppression algorithm: sorting the masks in decreasing order of foreground area, we iteratively select masks m_i that do not overlap with either previously selected masks or the background regions. The end result is a much shorter list of n SAM mask candidates $\{c_i(o)\}_{i=1}^n$ for slot binding.
- **Consistent slot binding:** Finally, to decide which candidates to bind to the k slots in our OCR representation of image o, we perform Hungarian matching [25] among n selected candidates $\{c_1(o), \dots, c_n(o)\}$, and the k task-relevant masks $\{l_1^{ref}, \dots, l_k^{ref}\}$. We compute as the matching costs the Euclidean distance between pre-trained DINO-v2 representations of each slot mask, obtained through ROI-pooling.

The final output is an ordered set of k masks $\{l_1(o), \dots, l_k(o)\}$, which will serve as the "where" component of our OCR.

B. The What: Representing The Image Contents in Each Slot

Given slot masks $\{l_1(o), l_2(o), \dots, l_k(o)\}$ for image o, we must compute, for each slot, its "slot vector" z_i. This slot vector captures the properties of the object visible in the scene regions specified by l_i, i.e., "what" is in l_i? As foreshadowed above, we will use a pre-trained LIV encoder to compute these slot vectors. For each slot i, we first generate a corresponding masked RGB image o_i by element-wise multiplying the binary mask l_i with the image o, and then compute LIV representations $z_i = \text{LIV}(I_i)$ over it. In addition to these slot-wise LIV features, we also compute the LIV representation of the unmasked original image $y(o) = \text{LIV}(o)$. Together, $\big(y(o), s(o) = \{(z_i(o), l_i(o))\}_{i=1}^k\big)$ constitutes our "plug-and-play" OCR, which we call SAM-LIV.

C. The How: Learning Robot Manipulation Policies from Demonstrations with SAM-LIV

So far, no learning has occurred as we have leveraged pre-trained representations to format visual observations into an OCR. For policy learning, we adopt an imitation learning paradigm, in which the policy network is trained to predict the expert actions in the provided demonstrations. With demonstrations, it is also easy to satisfy the assumption of task relevant object specficication as outlined in Section II-A. Now, we describe two straightforward methods for incorporating the extracted representation into policy learning.

Policies Over Concatenated Slot Vector Representations. One simple way to utilize the scene vector y and the object slot vectors ($\{z\}$) is to concatenate them into a fixed dimensional input vector and implement the policy architecture using a multi-layer perceptron (MLP):

$$\pi(y, \{z\}) := \text{MLP}(\texttt{Concatenate}(y, z_1, ..., z_k)) \quad (1)$$

This choice is simple to implement and suitable for visually simple environments in which object binding is unlikely to be inconsistent over different observations.

Policies Over Slot Permutation-Invariant Representations. Given that the object binding operation may be sensitive to noise and occasionally makes incorrect assignments, policy architectures that encode permutation invariance [23, 51, 52, 60] at the descriptor level may fare better for control [61]. We employ a self-attention (SA) [51] layer to process the OCR, and then aggregate the outputs to feed into an MLP policy.

$$\pi(y, \{z\}) := \text{MLP}\left(\sum \text{SA}(y, \{z\})\right) \quad (2)$$

III. Experimental Results

Our experiments aim to answer the following questions: 1) Does our method provide better off-the-shelf OCRs than alternative OCR approaches? 2) Does our method enable better policy learning compared to using flat pre-trained representations or curated masks alone? and 3) Does our method work on real robot? We begin by describing our simulation environments used to answer the first two questions, and then delve into

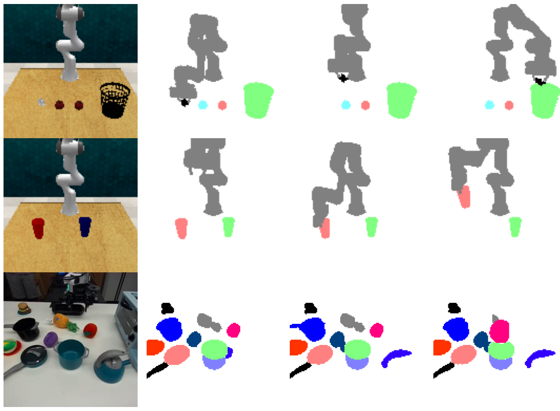


Fig. 2. SAM-LIV segmentation results over demonstrations.

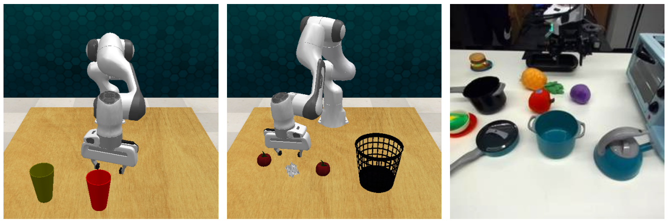
detailed experimental results answering all three questions affirmatively in Section III-A-III-C, respectively. Video and more visualization results: sites.google.com/view/sam-liv

Simulation Environments. We use RLBench [20] as our simulation testbed to validate our algorithmic design. More specifically, we have selected `Pick up Cup` and `Rubbish in Bin`, two challenging tasks from the RLBench suite that explicitly demands object-level inductive bias for successful learning (Figure 3(a) & 3(b)). `Pick up Cup` tasks the robot with picking up the red cup on the table in the presence of a distractor cup; the cup positions and the color of the distractor cup are randomized for each episode. `Rubbish in Bin` requires the robot to pick up the rubbish and place it inside the trash bin; the object locations, including the distractor apples, are randomized for each episode. Compared to `Pick up Cup`, this task also requires reasoning about object affordance (i.e., the desired rubbish location in the bin cannot be reached without first lifting the rubbish high off the table), and has been found empirically to be one of the most challenging tasks in RLBench for imitation learning [19].

A. Evaluating SAM-LIV Slot Masks

Quantitative Results. As discussed in Section A, prior deep OCRs typically require large domain-specific datasets for unsupervised training. This is unsuitable for sample-efficient policy learning in a new environment. For this experiment, we train AST-SEG [44], a state-of-the-art unsupervised OCR method, on our demonstrations in RLBench (about 1400 images for `Pick up Cup` and 2500 images for `Rubbish in Bin`). We report the quantitative results with foreground adjusted random index (FG ARI) [18, 42], a standard segmentation metric. SAM-LIV achieves 0.99 FG ARI scores (max is 1) on both tasks, while AST-SEG's fails to segment almost all the foreground objects, scoring only 0.2 on `Pick up Cup` and 0.1 on `Rubbish in Bin`.

Qualitative Results. We show the qualitative visualization of masks in various environments in Figure 2; see Appendix E for more qualitative results. To better understand the quality of these masks, we compute some qualitative metrics in simulation, exploiting the availability of ground-truth object masks. In particular, after we compute the masks for all observations in the policy learning dataset, we use pixel majority voting to decide SAM-LIV's slot assignment of each ground-truth object mask in an image. If the slot assignments are consistent with the pre-specified task-relevant mask subset (see section II-A), then we say they are correct for this image. Using this metric, we find that SAM achieves 94.3% accuracy on `Pick up Cup` and 87.8% accuracy on `Rubbish in Bin`, with an overall accuracy of 90.3%.



(a) Pick up Cup (b) Rubbish in Bin (c) RealRobot (3 Tasks)

Fig. 3. Evaluation Environments.

B. Policy Learning Simulation Experiments

Methods. To thoroughly validate our algorithm in controlled simulation setting, we ablate **SAM-LIV** along various axis. First, to assess the quality of our SAM-based object binding pipeline, we compare to using *ground truth* masks provided by the environment, denoted as **GT-LIV**. Note that this ablation cannot be implemented in real-world scenarios, but serves as an upper bound to assess the relative goodness of our method. Second, to stress the importance of explicit object reasoning, we compare **SAM-LIV** to **LIV**, keeping only the flat scene-level representation. Finally, to assess the value of a pre-trained model (LIV) for describing the contents of each object slot, we consider using a CNN network with and without *ground truth* masks, denoted as **CNN-RGB** and **GT-CNN-RGB**, trained from scratch as the visual descriptor. For this baseline, we use the official implementation from James and Davison [19], and train with imitation learning loss on in-domain demonstration data. As shown in Table I, we find both **CNN-RGB** and **GT-CNN-RGB** to struggle without privileged depth map input from the simulator (consistent with [19]). Therefore, we primarily consider a variant, **SAM-CNN-RGBD** that additionally incorporates depth maps to drive the CNN learning. Note that **any method that incorporates LIV does not use depth**, as our eventual goal is plug-and-play real-world usage in which accurate depth cannot be guaranteed.

Training & Evaluation. Our policy training and evaluation protocol mostly follows James and Davison [19]; in particular, for each task, we use 100 demonstrations collected using a motion planner, and train single-task policies using behavior cloning. The action space is Franka robot's 6-DOF end-effector pose and gripper state, and we use keyframe action representation to reduce the task horizon; see Appendix C for more details. In simulation, as we find the mask outputs of our object binding algorithm to closely match the ground-truth masks (see results below), we use the simple MLP architecture for the policy network to stay consistent with the original implementation. For each method, we train policies using 3 seeds and report the mean and the standard error of the maximum rewards each seed achieves during training on 100 evaluation rollouts, following standard practice [38].

Task	Pick up Cup	Rubbish in Bin
SAM-LIV	**126.0** ± 2.1	**50.0** ± 2.0
GT-LIV	137.3 ± 2.4	57.3 ± 3.3
LIV	98.7 ± 3.7	43.7 ± 1.5
SAM-CNN-RGBD	105.3 ± 21.0	25 ± 8.6
GT-CNN-RGBD	117.0 ± 15.5	16.7 ± 8.7
CNN-RGBD	98.3 ± 7.3	8.0 ± 2.3
GT-CNN-RGB	31.7 ± 2.0	8.7 ± 1.3
CNN-RGB	34.3 ± 4.9	5.3 ± 0.7

TABLE I
RLBench behavioral cloning mean episode reward averaged over 100 rollouts.

Results. As shown in Table I, SAM-LIV significantly outperforms all baselines, demonstrating the joint effectiveness of our object binding procedure and using pre-trained flat visual encoder as mask descriptors – all without any in-domain training of any component in the representation pipeline. Our mask generation procedure is generally effective, regardless of whether the curated masks are processed using LIV or a CNN trained from scratch, as the respective method closely tracks the variant that uses ground-truth mask. It is a priori not obvious that SAM-LIV or GT-LIV would outperform LIV, as LIV by itself is already a strong baseline and processing mask images using LIV may seem unnatural given LIV's training data. However, our results indicate that doing so is in fact quite effective. The benefit of using pre-trained flat vision encoder as mask descriptor further trickles down to downstream policy learning: LIV based methods exhibit far less variance compared to training-from-scratch CNN methods, all while delivering higher performance across the board without access to privileged depth information.

C. Real Robot Experiments

Given our encouraging simulation experiments, it is natural to ask whether our algorithm can work on real-world robotic manipulation tasks, which present the additional challenges of noisy image observations from imperfect camera sensors and increased object quantity and diversity.

Environment. To realize the stated challenges above, we design a real-world environment (referred to as RealRobot) that consists of a counter-top kitchen setup, in which a Franka robot is tasked with placing various fruits, {apple, eggplant, pineapple} in the green pot located on the far side of the table. Numerous distractors (e.g., toaster, black pot, black pan, burger plate) are placed on the table to create a more visually realistic kitchen scene, bringing the total number of objects to 10. We use a single 3rd-person monocular RGB camera for policy learning (see Figure 3(c) for the camera view), and this camera is placed on the far side of the table (see Appendix D for a side view of the scene), making object appearances smaller compared to more idealistic simulation setup.

Methods, Training & Evaluation. We compare SAM-LIV and LIV using behavior cloning with keyframe action representation as in our simulation experiment. For each task, we collect 100 trajectories using human teleoperation with the fruits randomly initialized in the center workspace of the table for each trajectory. As it is typical to train visuo-motor control policies in the real world with data augmentation to improve robustness, we train both methods with random cropping augmentation to attain best performance for all methods; for SAM-LIV, the random-cropping is consistently applied for both the raw RGB input and the masks input. To assess the raw generalization capability of respective representations, we also consider a setup without any data augmentation. As real-world mask outputs are noisier, our default SAM-LIV policy uses the attention policy architecture discussed in Section II-C. For each trained policy, we run 10 trials per task, randomizing the positions of all fruit objects, and we use the identical set of object randomizations for all policies. See Appendix D for more experimental details.

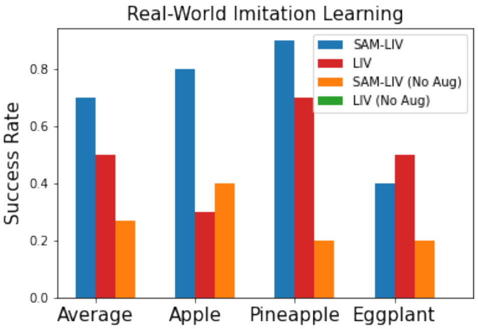


Fig. 4. RealRobot Imitation Results.

Results. As Figure 4 shows, SAM-LIV on average offers substantial gains compared to LIV; in the case of Apple, it achieves more than double the success rate. When trained without augmentation, SAM-LIV still achieves non-trivial performance, whereas LIV fails to solve any trial and overall exhibits degenerate reaching behavior, suggesting significant overfitting to the limited dataset size. We provide additional ablation results and analysis of BC losses in Appendix D. These results highlight the sensitivity of flat scene-level representations, even when they have been trained on large, diverse human videos. Several prior works [32, 38, 55] have demonstrated the capability of "what" foundation models on real-world visuomotor control tasks; however, their experiments all focus on single-task setting with limited object position variation. Given these models' lack of fine-grained object understanding, it is not surprising that they may struggle in more object-oriented tasks and overfit to just several motion trajectories in the limited data regime. However, as our experiments suggest, it is not that their representations are not compatible with fine-grained object reasoning, but rather that they are not been given the right input observations – the very issue that can be can be mitigated with our chaining approach that augments "what" foundation models by explicitly providing the "where" from a powerful off-the-shelf segmentation model.

IV. Conclusion

We have presented a simple yet effective framework for plug-and-play object-centric representations for visual robotic manipulation from "what" and "where" foundation models. Instantiated using state-of-art visual foundation models, SAM-LIV substantially outperforms baselines in simulation and real world without in-domain object-centric representation learning.

References

[1] Shirzad Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep vit features as dense visual descriptors. *ArXiv*, abs/2112.05814, 2021.

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

[3] Christopher P. Burgess, Loïc Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matthew M. Botvinick, and Alexander Lerchner. Monet: Unsupervised scene decomposition and representation. *ArXiv*, abs/1901.11390, 2019.

[4] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee, 2005.

[5] Edward HF de Haan and Alan Cowey. On the usefulness of 'what'and 'where'pathways in vision. *Trends in cognitive sciences*, 15(10):460–466, 2011.

[6] Coline Devin, P. Abbeel, Trevor Darrell, and Sergey Levine. Deep object-centric representations for generalizable robot learning. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7111–7118, 2017.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[8] Andrea Dittadi, Samuele S Papa, Michele De Vita, Bernhard Schölkopf, Ole Winther, and Francesco Locatello. Generalization and robustness implications in object-centric learning. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 5221–5285. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/dittadi22a.html.

[9] Martin Engelcke, Oiwi Parker Jones, and Ingmar Posner. Reconstruction bottlenecks in Object-Centric generative models. July 2020.

[10] Martin Engelcke, Oiwi Parker Jones, and Ingmar Posner. Genesis-v2: Inferring unordered object representations without iterative refinement. In *Neural Information Processing Systems*, 2021.

[11] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15180–15190, 2023.

[12] Melvyn A Goodale and A David Milner. Separate visual pathways for perception and action. *Trends in neurosciences*, 15(1):20–25, 1992.

[13] Klaus Greff, Sjoerd van Steenkiste, and Jürgen Schmidhuber. On the binding problem in artificial neural networks. December 2020.

[14] Siddhant Haldar, Jyothish Pari, Ananta Kant Rai, and Lerrel Pinto. Teach a robot to fish: Versatile imitation from one minute of demonstrations. *ArXiv*, abs/2303.01497, 2023.

[15] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask r-cnn. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42:386–397, 2017.

[16] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022.

[17] Negin Heravi, Ayzaan Wahid, Corey Lynch, Pete Florence, Travis Armstrong, Jonathan Tompson, Pierre Sermanet, Jeannette Bohg, and Debidatta Dwibedi. Visuomotor control in multi-object scenes using object-aware representations. *arXiv preprint arXiv:2205.06333*, 2022.

[18] Lawrence J. Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2:193–218, 1985.

[19] Stephen James and Andrew J Davison. Q-attention: Enabling efficient learning for vision-based robotic manipulation. *IEEE Robotics and Automation Letters*, 7(2): 1612–1619, 2022.

[20] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020.

[21] Thomas Kipf, Elise van der Pol, and Max Welling. Contrastive learning of structured world models. *ArXiv*, abs/1911.12247, 2019.

[22] Thomas Kipf, Gamaleldin F. Elsayed, Aravindh Mahendran, Austin Stone, Sara Sabour, Georg Heigold, Rico Jonschkowski, Alexey Dosovitskiy, and Klaus Greff. Conditional object-centric learning from video. *ArXiv*, abs/2111.12594, 2021.

[23] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[24] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross B. Girshick. Segment anything. *ArXiv*, abs/2304.02643, 2023.

[25] Harold W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics (NRL)*, 52, 1955.

[26] Tejas D. Kulkarni, Ankush Gupta, Catalin Ionescu, Sebastian Borgeaud, Malcolm Reynolds, Andrew Zisserman, and Volodymyr Mnih. Unsupervised learning of object keypoints for perception and control. *ArXiv*, abs/1906.11883, 2019.

[27] Sateesh Kumar, Jonathan Zamora, Nicklas Hansen,

Rishabh Jangir, and Xiaolong Wang. Graph inverse reinforcement learning from diverse videos. In *Conference on Robot Learning*, 2022.

[28] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behav. Brain Sci.*, pages 1–101, November 2016.

[29] Zhixuan Lin, Yi-Fu Wu, Skand Vishwanath Peri, Weihao Sun, Gautam Singh, Fei Deng, Jindong Jiang, and Sungjin Ahn. Space: Unsupervised object-oriented scene representation via spatial attention and decomposition. *ArXiv*, abs/2001.02407, 2020.

[30] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *ArXiv*, abs/2006.15055, 2020.

[31] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110, 2004.

[32] Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. Vip: Towards universal visual reward and representation via value-implicit pre-training. *arXiv preprint arXiv:2210.00030*, 2022.

[33] Yecheng Jason Ma, William Liang, Vaidehi Som, Vikash Kumar, Amy Zhang, Osbert Bastani, and Dinesh Jayaraman. Liv: Language-image representations and rewards for robotic control. *arXiv preprint arXiv:2306.00958*, 2023.

[34] Arjun Majumdar, Karmesh Yadav, Sergio Arnaud, Yecheng Jason Ma, Claire Chen, Sneha Silwal, Aryan Jain, Vincent-Pierre Berges, Pieter Abbeel, Jitendra Malik, et al. Where are we in the search for an artificial visual cortex for embodied intelligence? *arXiv preprint arXiv:2303.18240*, 2023.

[35] Toki Migimatsu and Jeannette Bohg. Object-centric task and motion planning in dynamic environments. *IEEE Robotics and Automation Letters*, 5:844–851, 2019.

[36] Matthias Minderer, Chen Sun, Ruben Villegas, Forrester Cole, Kevin P. Murphy, and Honglak Lee. Unsupervised learning of object structure and dynamics from videos. *ArXiv*, abs/1906.07889, 2019.

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

[38] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022.

[39] Samuele Papa, Ole Winther, and Andrea Dittadi. Inductive biases for object-centric representations in the presence of complex textures. In *UAI 2022 Workshop on Causal Representation Learning*, 2022.

[40] Simone Parisi, Aravind Rajeswaran, Senthil Purushwalkam, and Abhinav Gupta. The unsurprising effectiveness of pre-trained vision models for control. *arXiv preprint arXiv:2203.03580*, 2022.

[41] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[42] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66:846–850, 1971.

[43] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1137–1149, 2015.

[44] Bruno Sauvalle and Arnaud de La Fortelle. Unsupervised multi-object segmentation using attention and soft-argmax. *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3266–3275, 2022.

[45] Rutav Shah and Vikash Kumar. Rrl: Resnet as representation for reinforcement learning. *arXiv preprint arXiv:2107.03380*, 2021.

[46] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In *Conference on Robot Learning*, pages 894–906. PMLR, 2022.

[47] Maximilian Sieb, Xian Zhou, Audrey Huang, Oliver Kroemer, and Katerina Fragkiadaki. Graph-structured visual imitation. In *Conference on Robot Learning*, 2019.

[48] Jonathan Tremblay, Thang To, Balakumar Sundaralingam, Yu Xiang, Dieter Fox, and Stan Birchfield. Deep object pose estimation for semantic robotic grasping of household objects. In *Conference on Robot Learning*, 2018.

[49] Leslie G Ungerleider and James V Haxby. 'what'and 'where'in the human brain. *Current opinion in neurobiology*, 4(2):157–165, 1994.

[50] Sjoerd van Steenkiste, Klaus Greff, and Jürgen Schmidhuber. A perspective on objects and systematic generalization in model-based rl. *arXiv preprint arXiv:1906.01035*, 2019.

[51] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[52] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[53] Dequan Wang, Coline Devin, Qi-Zhi Cai, Fisher Yu, and Trevor Darrell. Deep object-centric policies for autonomous driving. *2019 International Conference on Robotics and Automation (ICRA)*, pages 8853–8859, 2018.

[54] Ted Xiao, Harris Chan, Pierre Sermanet, Ayzaan Wahid,

Anthony Brohan, Karol Hausman, Sergey Levine, and Jonathan Tompson. Robotic skill acquisition via instruction augmentation with vision-language models. *arXiv preprint arXiv:2211.11736*, 2022.

[55] Tete Xiao, Ilija Radosavovic, Trevor Darrell, and Jitendra Malik. Masked visual pre-training for motor control. *arXiv preprint arXiv:2203.06173*, 2022.

[56] Yafei Yang and Bo Yang. Promising or elusive? unsupervised object segmentation from real-world single images. *NeurIPS*, 2022.

[57] Yufei Ye, Dhiraj Gandhi, Abhinav Kumar Gupta, and Shubham Tulsiani. Object-centric forward modeling for model predictive control. *ArXiv*, abs/1910.03568, 2019.

[58] Jaesik Yoon, Yi-Fu Wu, Heechul Bae, and Sungjin Ahn. An investigation into pre-training object-centric representations for reinforcement learning. *arXiv preprint arXiv:2302.04419*, 2023.

[59] Sarah Young, Jyothish Pari, P. Abbeel, and Lerrel Pinto. Playful interactions for representation learning. *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 992–999, 2021.

[60] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017.

[61] Allan Zhou, Vikash Kumar, Chelsea Finn, and Aravind Rajeswaran. Policy architectures for compositional generalization in control. *arXiv preprint arXiv:2203.05960*, 2022.

[62] Yifeng Zhu, Abhishek Joshi, Peter Stone, and Yuke Zhu. Viola: Imitation learning for vision-based manipulation with object proposal priors. *ArXiv*, abs/2210.11339, 2022.

[63] Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Gao, and Yong Jae Lee. Segment everything everywhere all at once. *ArXiv*, abs/2304.06718, 2023.

APPENDIX

A. Problem Setup and Background

We are interested in sample-efficiently learning robotic manipulation policies in arbitrary multi-object scenes, with some task-relevant and some distracting objects. For example, in our real robot experiments, we task a robot arm attached to a cluttered kitchen counter-top with moving fruits and vegetables into various pots and pans around it, with only a few tens of demonstrations.

Object-Centric Representations (OCR). We propose to enable such tasks with object-centric representations (OCRs) of visual scenes. Like many prior works [3, 10, 21, 29, 30, 44], we target an OCR that at each time t summarizes the scene o^t in terms of various discrete "slots" s_i^t, that ideally correspond to the entities in the scene, i.e., objects and parts. To unclutter notation, will omit the time index t when it is not relevant. Each slot is a tuple $s_i = (l_i, z_i)$ with two components: (1) the location or "where" component l_i indicates the presence and location of an entity, such as through a segmentation mask [3, 10, 29, 30, 44], bounding box [6, 27, 47, 53, 62], or keypoint location [26, 36]. (2) the content or "what" component, often called a "slot vector" $z_i \in \mathbb{R}^D$ captures the properties of the object such as its texture, pose, and affordances, visible in the scene regions $o[l_i]$ identified by l_i.

The Pros and Cons of OCRs. OCRs disentangle scene objects, enabling improved systematic generalization, symbolic reasoning, sample-efficient learning, and causal inference starting from visual inputs [8, 13, 28, 50, 58] compared to unstructured "flat" vector representations of the scene. They can also serve as a shared representation interface[22] between humans and robots, which is potentially useful for task specification. For example, a system that understands the world in terms of objects can understand natural instructions such as "place object-1 upon object-4". In our approach, we will take advantage of this interface capability to identify task-relevant objects in a cluttered scene.

Despite all these potential advantages of OCRs, state-of-the-art approaches in robot learning today commonly use flat vector representations of the scene [14, 32, 38, 54, 59]. We argue that this is primarily because training deep neural networks to generate object-centric representations is difficult; they require non-standard architectures, and do not train as stably. This in turn means that current deep OCR encoders are restricted to be relatively small-capacity networks that are highly sensitive to architectural choices [9, 39]. They must therefore be trained on domain-specific data, and even then, on large image datasets in relatively small domains. Leave alone re-using pre-trained OCR encoders in new task domains, state-of-the-art OCR encoders perform poorly even in-domain in realistic, visually complex settings [56], as we also find in our experiments.

Thus today's OCRs trail flat representations in practical utility. For example, pre-trained flat representation encoders can enable robot learning in new domains [32, 33, 37, 55]. Indeed, in our attempt to build similarly re-usable pre-trained OCR encoders, we too will re-use one such flat representation encoder LIV [33], alongside another pre-trained model SAM [24], that specializes in segmenting images. We now briefly discuss these two models.

Segment Anything Model (SAM). Our approach exploits recent large advances in image instance segmentation [24, 63] for building an image representation for robotics. Specifically, our experiments uses the pre-trained SAM [24] model off-the-shelf, but our approach is more general and permits using arbitrary future instance segmentation. At inference time, given an RGB image of size HxWx3, SAM can generate a full set of segmentation masks $\{m_1, m_2, \ldots, \}$ that identify pixel groupings potentially corresponding to object-like entities at varying levels of granularity.

Language-Image Value (LIV) representations. Our approach also requires a pre-trained visual encoder that provides flat scene-level vector representations of images. In our experiments, we use LIV [33], a vision-language representation pre-trained on a large human video dataset. The pre-trained model contains a vision encoder and a language encoder; we are primarily concerned with the LIV vision encoder, which has been shown to work well as a state representation for vision-based robotic tasks in cluttered scenes.

B. Other Related Work

1) Traditional Uses Of Object Detectors In Robotics.: We have motivated OCRs and discussed recent work on deep OCRs in Sec A. In a way, our approach of combining pre-trained models into one OCR encoder without any training is reminiscent of more traditional and modular approaches to representing visual scenes in robotics, such as by computing hand-defined (e.g., SIFT, HOG) features over object detector outputs [4, 31]. Such approaches have continued to be useful since the advent of deep learning, e.g., recent works have employed detectors for object poses [35, 48, 57] and bounding boxes [6, 27, 47, 53, 62]. Given the abundance of research in object detection from the computer vision community, those works either leverage existing object detectors [27, 47, 53] such as Mask R-CNN [15] or incorporate vision backbones such as a region proposal network [43] for general object proposals and then train a policy that attends to the task-relevant information [6, 62]. However, these methods typically require fine-tuning on their datasets and require prior knowledge of object categories thus failing to handle previously unseen objects. Indeed, the growing literature on unsupervised deep object-centric representations (OCRs) is motivated by the desire to move beyond such domain-specific labeled datasets, but has its own disadvantages, as we motivated in Sec A. Powered by recent advances in category-agnostic segmentation, we have proposed a truly "off the shelf", general-domain OCR that can be reused by robot learners in arbitrary domains.

2) Pre-trained Flat Representations for Control: As discussed in Sec A, our work aims to fix the gap between flat and object-centric representations for control by presenting a general-domain pre-trained OCR, inspired by the many such solutions that provide pre-trained *flat* representations for control [32, 34, 38, 40, 45, 55]. These works have shown how

	RLBench	RealRobot
Self-Attention Architecture	N/A	4 Heads, 256 Hidden Dimension
MLP Architecture	[256, 256]	[256, 256]
Non-Linear Activation	Leaky ReLU	ReLU
Optimizer	Adam	Adam
Gradient Steps	250000	10000
Batch Size	128	64
Learning Rate	0.0005	0.001
Proprioception	Yes	No
Augmentation	Demo augmentation [19]	Random Cropping

TABLE II
Imitation Learning Hyperparameters.

frozen visual representations, pre-trained on out-of-domain data, can serve as effective visual encoder for policy learning on unseen robot tasks. However, flat image-level representations typically lose fine-grained object-centric information that is often necessary for solving tasks that require multi-object reasoning [17] or require training another specialized architecture on in-domain data for generalization [46].

C. Simulation Experimental Details

Keyframe action representation. Following the setup of James and Davison [19], we perform keyframe discovery on over demonstration dataset to reduce the task horizon. Iterating over each of the demo trajectories τ, we use a Boolean function to decide whether each trajectory point is a keyframe. The Boolean function is a disjunction of change in gripper state and velocities approaching near zero. In our real-world experiment, we use simpler heuristic to mine keyframe actions.

D. Real Robot Experimental Details

RealRobot Environment. In Figure 6, we show a side view of the RealRobot environment to better illustrate the position of the camera that is used for policy learning.

Task Specification. Each task is specified using text description (e.g., `apple in green pot`), which is natural given that both SAM-LIV and LIV has access to LIV's language encoder to enable language-based task specification. Then, the language embedding vector is treated as another input vector to the policy.

Imitation Learning Losses. Besides policy success rates, we also visualize the BC training and validation losses incurred during policy learning for SAM-LIV and LIV. In addition, we consider several SAM-LIV ablations such as removing the self-attention layer with MLP and reducing the learning rate. As shown in Figure 5, LIV clearly underfits SAM-LIV, even though both methods do use the same visual encoder and differ only in what inputs go through the encoder. The superior generalization of SAM-LIV, as shown in the validation loss, is not solely attributed to its improved expressivity. By decreasing the policy learning-rate by ten-fold, we see that **SAM-LIV (Small LR)** does now incur higher training loss than LIV, but still delivers lower validation loss than LIV, demonstrating that SAM-LIV's out-of-box generalization capability is robust to hyperparamter choices. Likewise, SAM-LIV without attention exhibits higher validation loss, indicating that the lack of permutation invariance inductive bias hurts generalization due to sensitivity to noisy mask outputs (Figure 2 shows an example where the mask output drops certain objects in the scene).

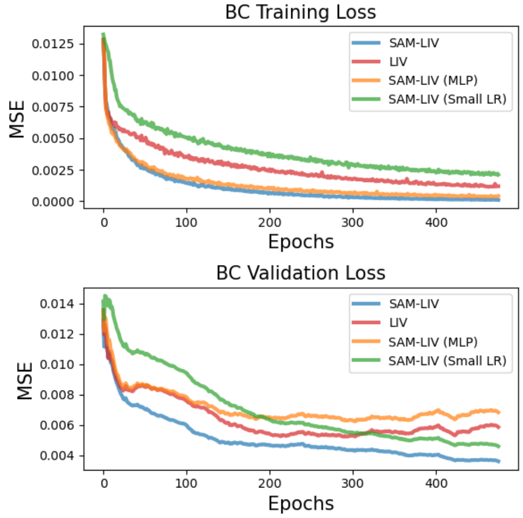


Fig. 5. BC Losses on RealRobot.

E. More Qualitative Results

Figures 7, 8, 9, 10, and 11 show additional qualitative visualizations of real-robot and simulation demonstration episodes. The leftmost column is the original RGB image, and the rest of slot assignments produced by SAM-LIV, with the second to the left column as background. Each row shows a keyframe point in the demonstration sequence discovered by our keyframe discovery procedure (See Appendix C for more details). Figures 12, 13, 14, 15, and 16 show the same demonstration sequences, but with the slot assignments visualized in one image. The top row are the RGB images of keyframes, and the bottom row are the corresponding overlay of SAM-LIV segmentations.

F. Limitations

With regard to limitations, our simulation experiments are limited in the number of tasks, and most of our tasks resemble

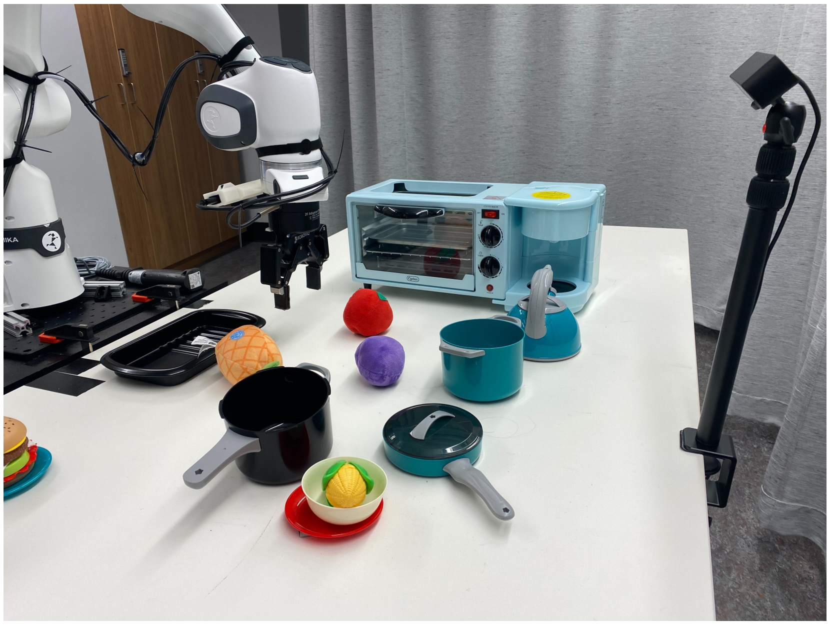


Fig. 6. RealRobot Environment Side View.

some form of pick-and-place motion. However, as our pipeline does not make assumption on task type, action space, and policy architecture, we aim to extend to more diverse tasks in both simulation and real-robot. Another limitation is that the performance of our pipeline is bottlenecked by the quality of the individual “what” and “where” foundation models. Furthermore, SAM-LIV policy inference is slower due to online generation of our object-centric representation of the current environment observation. However, these limitations can be addressed by the steady improvement in the quality of visual foundation models and by adopting a faster online OCR generation method that trades-off speed with accuracy.

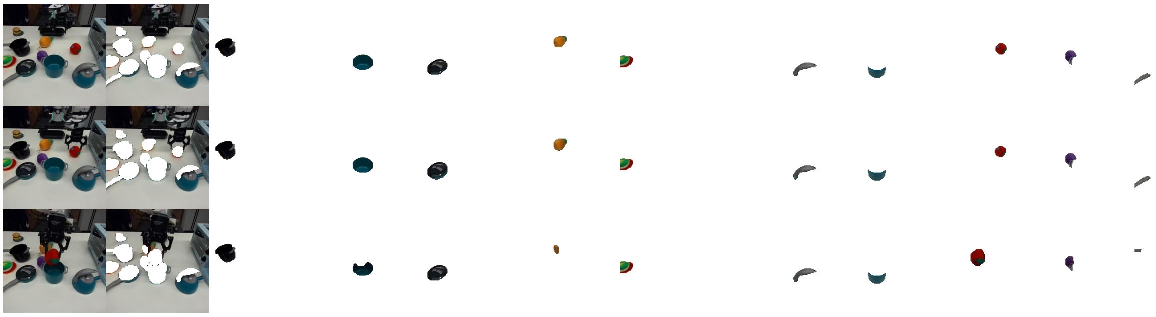


Fig. 7. Slot assignments breakdown of keyframes in one demonstration episode of RealRobot `Apple in Green Pot`

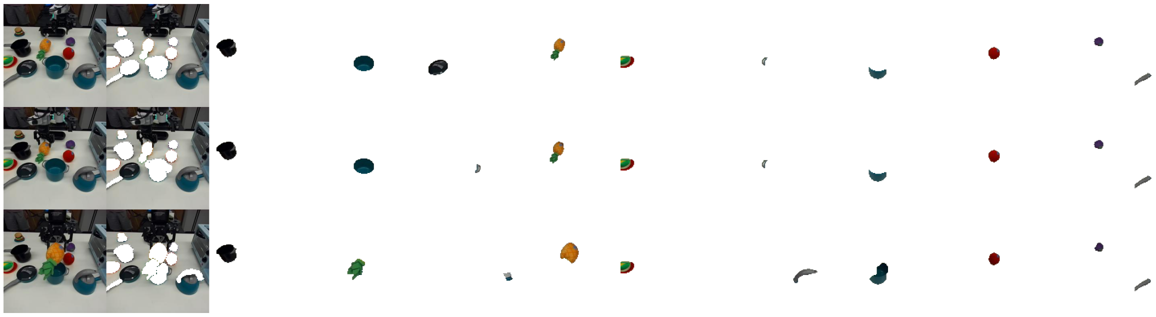


Fig. 8. Slot assignments breakdown of keyframes in one demonstration episode of RealRobot `Pineapple in Green Pot`

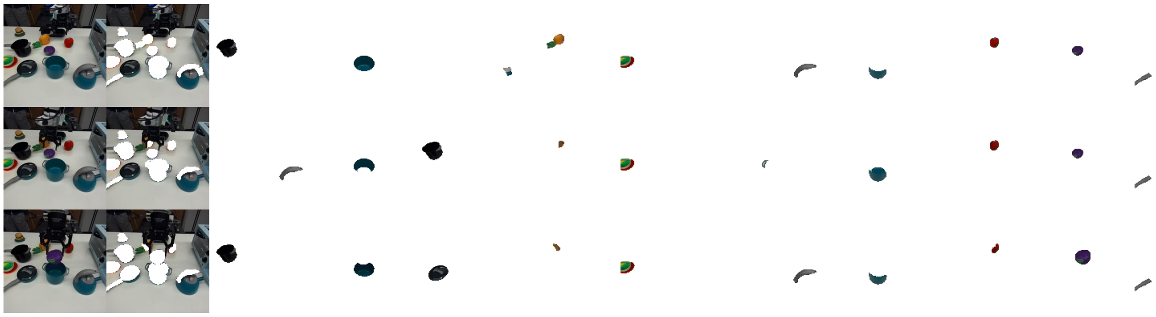


Fig. 9. Slot assignments breakdown of keyframes in one demonstration episode of RealRobot `Eggplant in Green Pot`

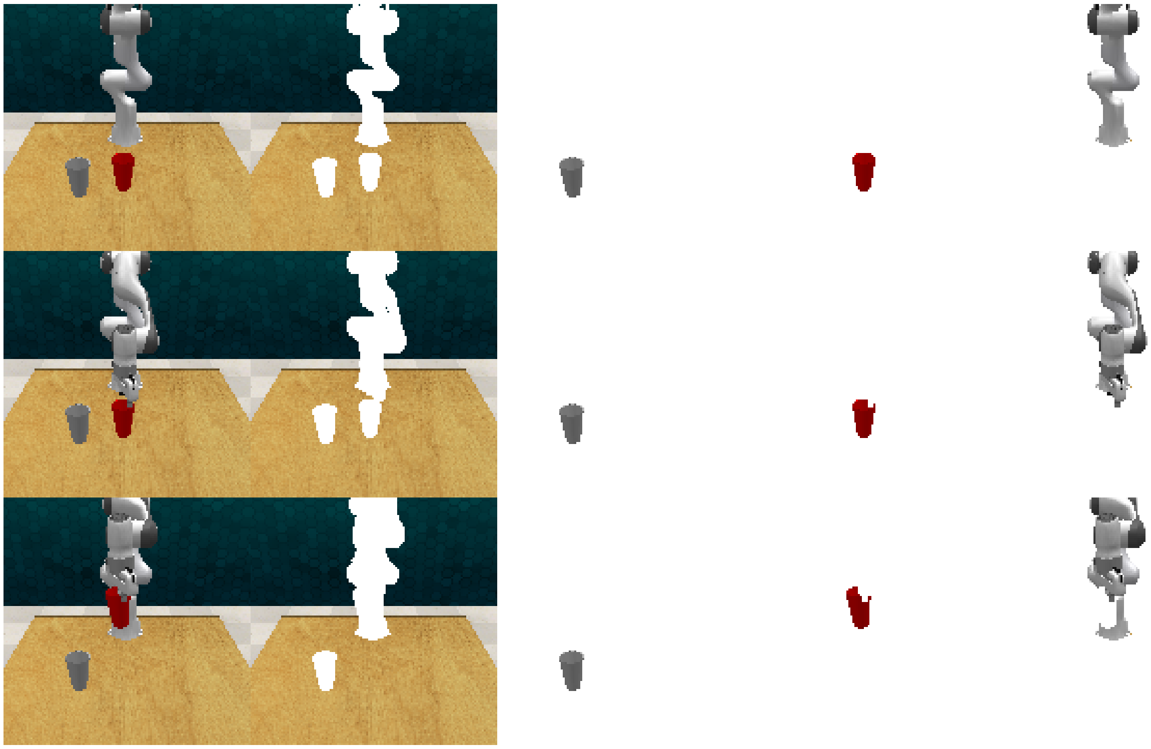


Fig. 10. Slot assignments breakdown of keyframes in one demonstration episode of RLBench `Pick up Cup`

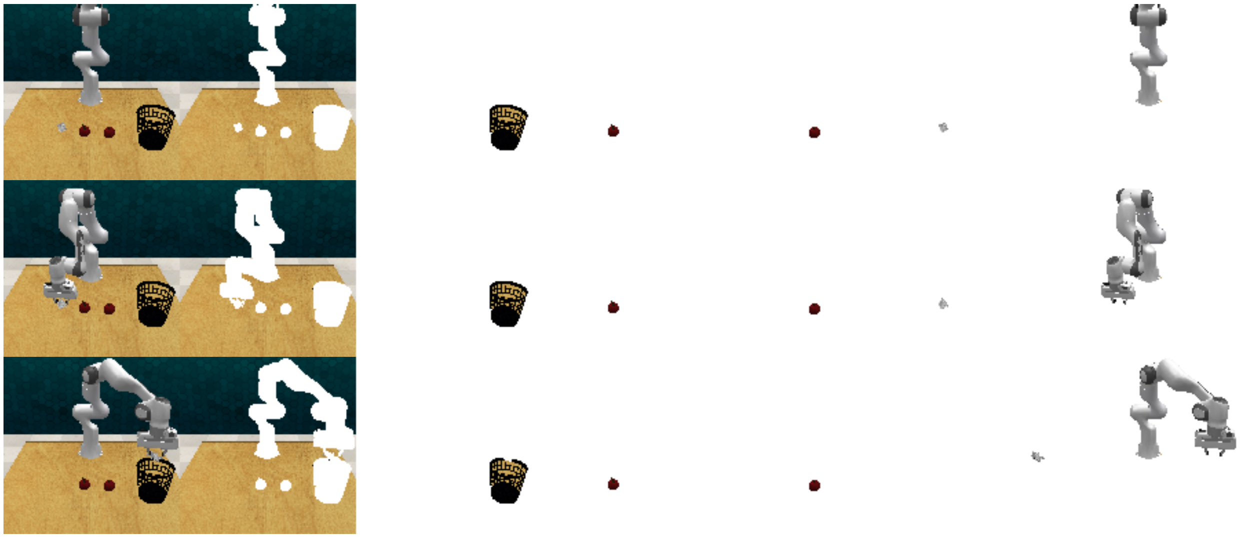


Fig. 11. Slot assignments breakdown of keyframes in one demonstration episode of RLBench `Rubbish in Bin`

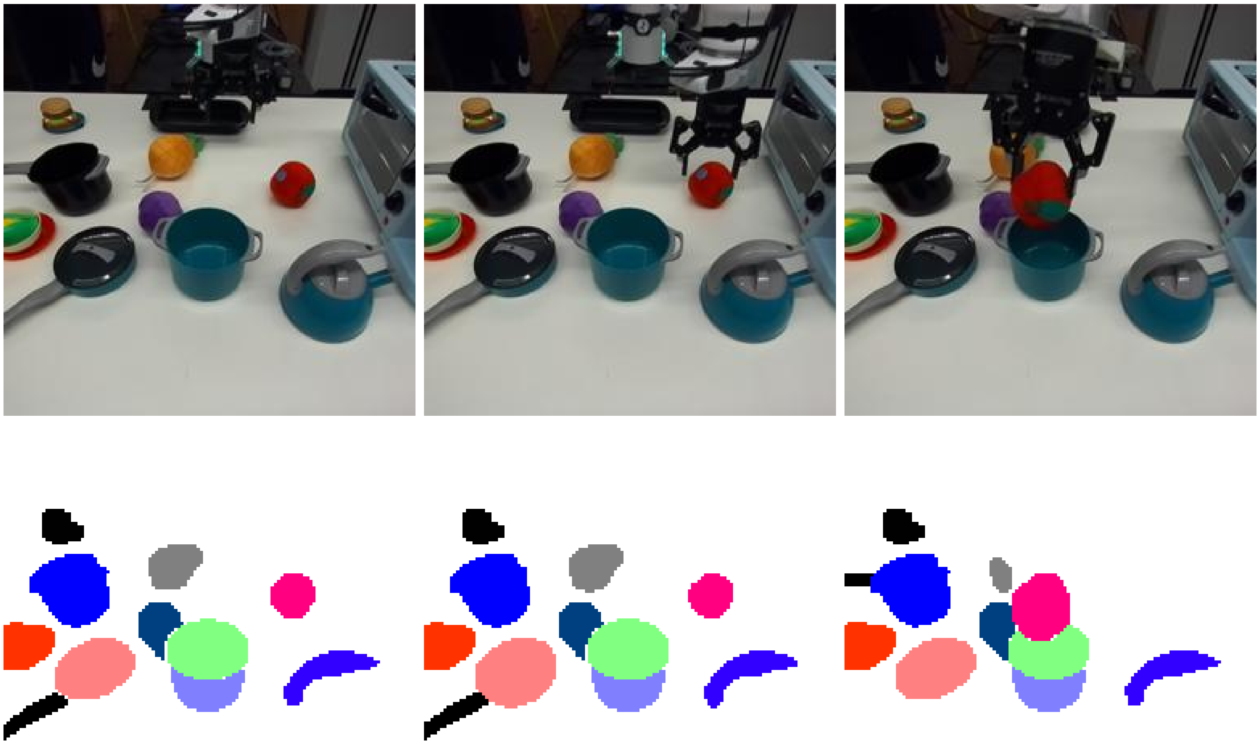


Fig. 12. Object masks overlay of keyframes in one demonstration episode of RealRobot `Apple in Green Pot`

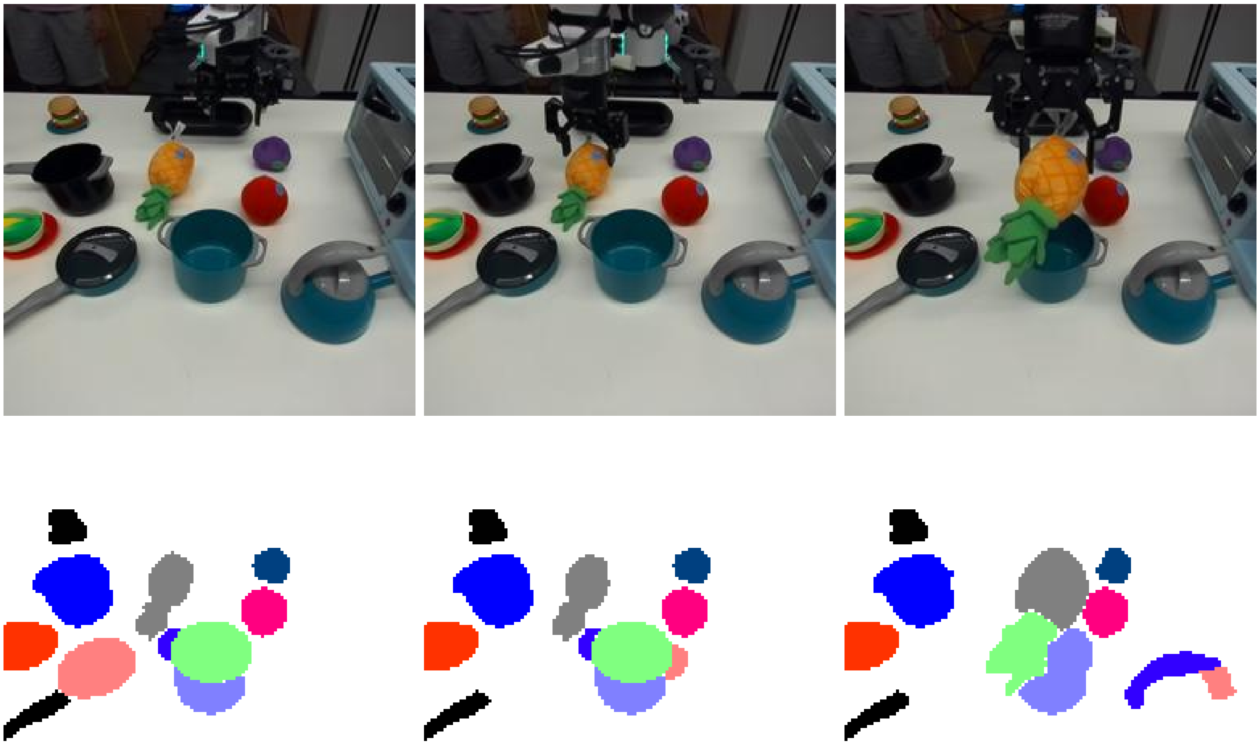


Fig. 13. Object masks overlay of keyframes in one demonstration episode of RealRobot `Pineapple in Green Pot`

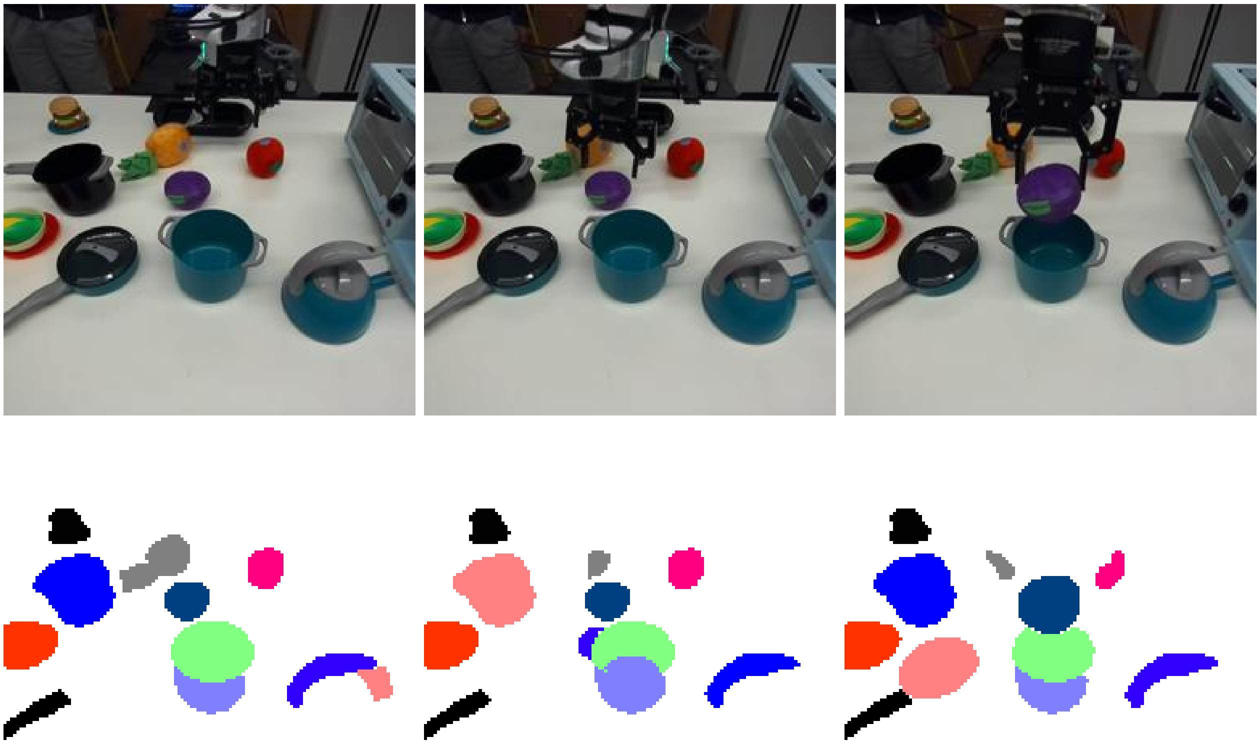


Fig. 14. Object masks overlay of keyframes in one demonstration episode of RealRobot `Eggplant in Green Pot`

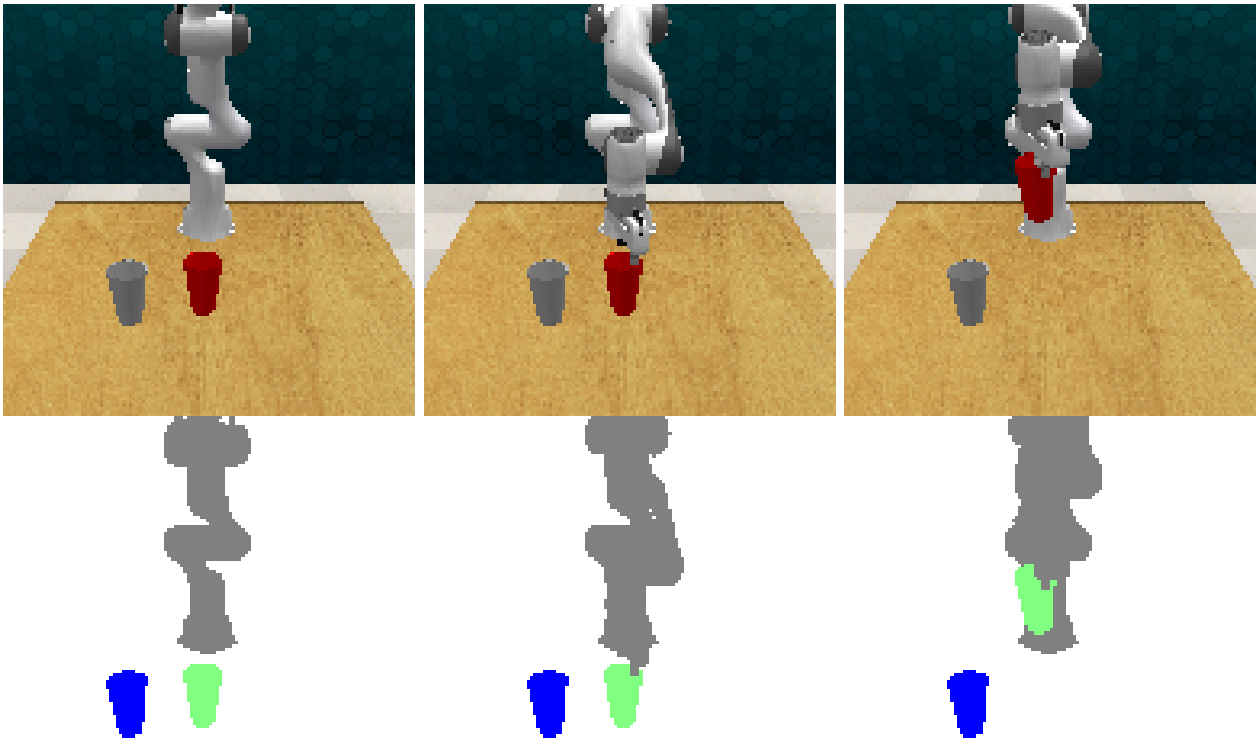


Fig. 15. Object masks overlay of keyframes in one demonstration episode of RLBench `Pick up Cup`

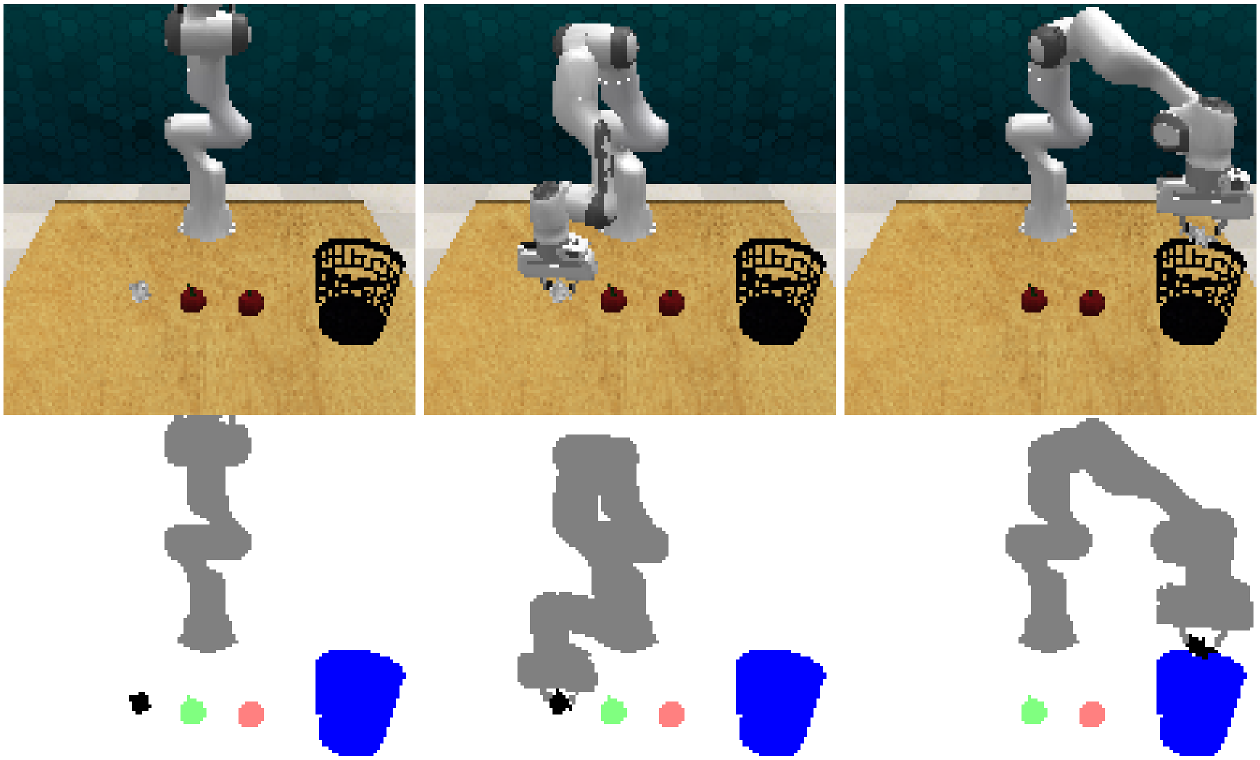


Fig. 16. Object masks overlay of keyframes in one demonstration episode of RLBench `Rubbish in Bin`