Composing Pre-Trained Object-Centric Representations for Robotics From "What" and "Where" Foundation Models

Junyao Shi[*1], Jianing Qian[*1], Yecheng Jason Ma[1], Dinesh Jayaraman[1]

sites.google.com/view/pocr

Abstract— There have recently been large advances both in pre-training visual representations for robotic control and segmenting unknown category objects in general images. To leverage these for improved robot learning, we propose POCR, a new framework for building pre-trained object-centric representations 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 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 pre-trained object-centric representations for control are 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 POCR achieve better performance and systematic generalization than state of the art pre-trained representations for robotics, as well as prior object-centric representations that are typically trained from scratch.

I. INTRODUCTION

One of the fundamental challenges of developing general-purpose robots is how to build informative and generalizable visual representations that permit robots to acquire diverse manipulation skills. 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 language [1]–[4] and vision [3]–[5]. Recent works [6]–[9] have also shown that the same paradigm could be applied to robotics by leveraging internet images [10] and human videos [11]. Pre-trained representations for robotic control have clear advantages over representations trained in-domain: they can be flexibly deployed off-the-shelf, and they do not require large amounts of expensive task-specific data for learning.

While these pre-trained control-aware encoders have been proven useful for policy learning, they don't explicitly capture discrete and meaningful entities such as objects, which are essential for understanding the observation and reasoning about actions. In humans, per the "what-where" representation theory [12]–[14] in cognitive science, the brain uses specialized 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 embeddings (OCEs) [15]–[27] instantiates these ideas in

[*]Equal Contribution.

[1]Computer and Information Science, University of Pennsylvania. Email: `junys, jianingq, jasonyma, dineshj@seas.upenn.edu`

artificial intelligence, commonly focusing on co-training the "what" and "where" pathways within a target domain.

We investigate a simpler route to OCEs suitable for robots, paved by: (1) recent advances in image segmentation [28], [29], the task of identifying groups of pixels in an image 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, and (2) recently proposed pre-trained representations for control.

We propose pre-trained object-centric representations for robotics (POCR), a general-purpose ready-made model constructed by appropriately chaining segmentation and control-aware vision "foundation models". Each such model has individually been pre-trained on large and diverse datasets, and afterwards been shown to work well on many domains of interest. A composite representation that inherits these generalization properties may be used off-the-shelf in arbitrary new robotics tasks; see Figure 1 for a schematic overview.

In our experiments, we study various choices of foundation models to plug-and-play in the POCR framework. On unseen simulated and *real-world* robotic manipulation tasks, we find that the best POCR representations enable significantly better policy learning than all prior representations. POCR policies even demonstrate systematic generalization to unseen test-time variations in the scene. In summary, our findings suggest that POCR provides a simple framework for generating, to our knowledge, the very first generic pre-trained object-centric representations for robotics that can reliably be used off-the-shelf in new robotic environments and tasks.

II. PROBLEM SETUP AND BACKGROUND

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

Pre-Trained Representations for Control. Our work builds on top of the literature on pre-trained visual representations for control [6], [8], [9], [30]–[32]. These works have shown how frozen visual representations, pre-trained on out-of-domain data, can serve as effective visual encoders for policy learning on unseen robot tasks. However, flat image-level representations typically lose fine-grained object-centric information often necessary for solving tasks that require reasoning about multi-object relationships, object affordance, or object articulations [33]. Indeed, in our experiments, we discovered that they struggle with learning actions in multi-object scenes. Nevertheless, they serve as effective "what"
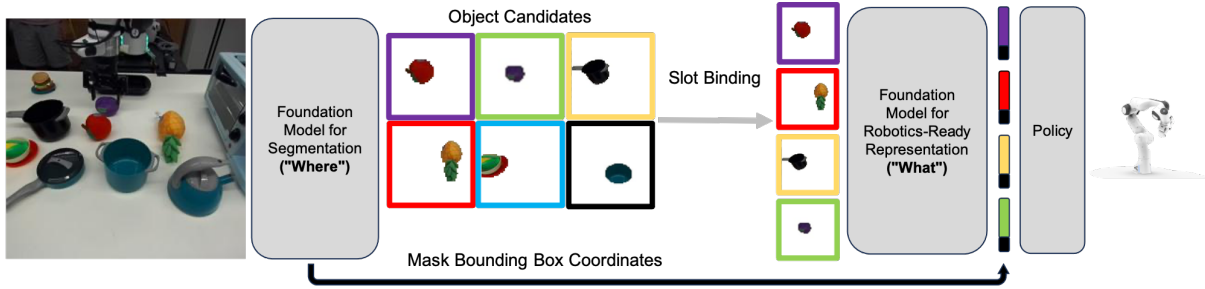
Fig. 1. **POCR**: **P**re-Trained **O**bject-**C**entric **R**epresentations for Robotics by chaining "what" and "where" foundation models. The "where" foundation model produces a set of masks representing objects in the scene. Slot binding selects which among them to bind to the slots in our OCE. Image contents in each slot are represented by the "what" foundation model and their mask bounding box coordinates. The robot learns policies over slot representations.

descriptors in our compositional "what-where" framework.

Object-Centric Embeddings (OCEs), Their Pros and Cons. We propose to enable manipulation tasks with OCEs of visual scenes. Many prior works [15]–[20] formulate OCEs using two components: (1) the location, or "where" component, l_i indicates the presence and location of an entity, such as through a segmentation mask [15]–[19], bounding box [21]–[25], or keypoint location [26], [27], and (2) the content, or "what" component, captures the properties of the object such as its texture, pose, and affordances.

OCEs disentangle scene objects, enabling improved systematic generalization, symbolic reasoning, sample-efficient learning, and causal inference starting from visual inputs [34]–[38] compared to unstructured "flat" vector representations of the scene. They can also serve as a shared representation interface [39] between humans and robots, which is potentially useful for task specification.

Despite all these potential functional advantages of OCEs, state-of-the-art pre-trained approaches in robot learning today commonly use flat vector representations of the scene [8], [30], [40]–[42]. We argue that this is primarily because training deep neural networks to generate OCEs is difficult; they require non-standard architectures and do not train as stably. This in turn means that current deep OCEs are restricted to be relatively small-capacity networks that are highly sensitive to architectural choices [43], [44]. 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 OCEs in new task domains, state-of-the-art OCEs perform poorly even in-domain in realistic, visually complex settings [45], as we also find in our experiments.

Thus today's OCEs trail flat representations in practical utility, whereas flat representations trail OCEs in functionality. Our attempt aims to get the best of both worlds by building functional and practical pre-trained OCEs for robotics.

III. PRE-TRAINED OBJECT CENTRIC REPRESENTATIONS FOR ROBOTIC MANIPULATION

We target an OCE that at each time t summarizes the scene o^t in terms of discrete "slots" s_i^t, that correspond to the entities in the scene, i.e., objects and parts. To unclutter notation, we 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 l_i and (2) the content, often called a "slot vector" $z_i \in \mathbb{R}^D$, visible in the scene regions $o[l_i]$ identified by l_i.

Since segmentations provide fine-grained information about object entities, they are an ideal choice for representing l_i. However, prior methods that use masks inevitably require extensive in-domain training, since no pre-trained models are designed to encode masks. Meanwhile, pre-trained vision encoders can be used off-the-shelf for representing image content, but their flat representations don't explicitly capture object entities. It is thus a priori non-obvious how to properly combine masks and pre-trained encoders to properly represent l_i and z_i in an OCE framework. Towards our unique approach to chain "where" and "what" foundation models into a useful representation for manipulation policy learning, we address three key questions: **where** are the object regions, **what** are their contents, and **how** should robots act to accomplish manipulation tasks given such what-where representations.

A. The Where: Localizing and Assigning Objects to Slots

To go from segmentation outputs to slot masks l_i, we propose a lightweight procedure to match each mask to object segments in a reference image (e.g., the initial frame).

Obtaining object segments in a reference image by screening the object-level foreground entity candidates. We collect some reference images, such as from the initial frames of demonstrations, and we use them to compute the background regions in the image following the procedure in [46]. Then, on one of these same reference images o^{ref}, we run the segmentation model to produce the set of masks m_i^{ref}. By design, a general-purpose segmentation model produces an over-complete set of segmentation masks corresponding to various entities i in the scene at various levels of granularity, and also irrelevant entities in background regions of the scene.

To discard background and incomplete entities among these reference image masks, we identify object-level foreground entities among m_i^{ref} as follows. 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 mask candidates $\{l_1^{ref}, \ldots, l_k^{ref}\}$ for object slot binding in the next step. The number of slots k is set to a large constant independent of the task or frame, and if segmentation proposals are fewer than k, the extra slots are treated as empty.

Localizing objects in each observation via consistent slot binding. Given these selected reference masks that

encode objects, the slots in our desired OCE must bind to these objects in each image. Towards this, we first apply the aforementioned procedure to screen the object-level foreground entity candidates to obtain a shorter list of mask candidates $\{m_1(o), \ldots, m_n(o)\}$ in image o. Next, to decide which among them to bind to the k slots in our OCE of the image o, we perform a Hungarian matching [47] between the n selected candidates $\{m_1(o), \ldots, m_n(o)\}$, and the k masks $\{l_1^{ref}, \ldots, l_k^{ref}\}$ representing object slots in the reference image. For the matching costs, we compute the cosine 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 $\{l_1(o), \ldots, l_k(o)\}$. In our method, each mask is represented by its bounding box coordinates, which serve as the "where" component of our OCE.

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

Given slot masks $\{l_1(o), \ldots, l_k(o)\}$ 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 at l_i? As foreshadowed above, we will use pre-trained vision encoders 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 image representations $z_i = \text{encoder}(o_i)$ over it. Together, $s(o) = \{(z_i(o), l_i(o))\}_{i=1}^k$ constitutes POCR, our "plug-and-play" OCE framework for robotics.

C. The How: Learning Robot Manipulation Policies from Demonstrations with POCR

So far, no learning has occurred as we have leveraged pre-trained models to format visual observations into OCEs. Given that the object binding operation may be sensitive to noise and occasionally make incorrect assignments, it is natural to use policy architectures that encode permutation invariance [48]–[52]. We employ a self-attention (SA) [49] neural layer to process the OCEs, and then aggregate the outputs to feed into an MLP policy.

$$\pi_\theta(\{(z_i, l_i)\}_{i=1}^k) := \text{MLP}_\theta \left(\sum_{i=1}^k \text{SA}_\theta[z_i, l_i] \right), \quad (1)$$

where $[\cdot, \cdot]$ is the concatenation operation. We train policies with a mean squared error (MSE) behavior cloning loss to predict the expert actions in the provided demonstrations.

IV. Other Related Work

Traditional Uses Of Object Detectors In Robotics. In a way, our approach of combining pre-trained models into one OCE encoder with no 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 [53], [54]. Such approaches have continued to be useful since the advent of deep learning, e.g., recent works have employed detectors for object poses [55]–[58] and bounding boxes [21]–[25], [59], [60]. Given the abundance of research in object detection

from the computer vision community, those works either assume ground-truth object states [61], leverage existing object detectors [22], [23], [25], [59], [60] such as Mask R-CNN [62] or incorporate vision backbones such as a region proposal network [63] for general object proposals and then train a policy that attends to the task-relevant information [21], [24]. However, these methods typically require prior knowledge of object categories thus failing to handle previously unseen objects. These methods are also data-intensive, requiring significantly larger in-domain task-specific datasets than ours to learn or refine the OCE. Indeed, the growing literature on unsupervised deep OCEs is motivated by moving beyond such domain-specific labeled datasets, but has its own disadvantages, as we motivated in Sec II.

Concurrent Works That Use Foundation Models for Segmentation. Concurrent works [64]–[67] also study the usage of general-purpose segmentation models such as SAM [28] for control. FOCUS [64] uses SAM to generate mask supervision for a model-based agent that learns an object-centric world model. MMPM [65] employs SAM to obtain language-grounded object masks given their bounding boxes. GROOT [66] utilizes SAM to construct object-centric 3D representations. RoboHop [67] leverages SAM to generate topological segment-based map representation for robot navigation. Among them, only GROOT [66] and RoboHop [67] handle object tracking in complex scenes, and none of them propose a viable pre-trained OCE for robotics. We set up baselines that are similar in spirit to GROOT and RoboHop: "SAM-Scratch", like GROOT and unlike POCR, embeds SAM masks with an encoder trained from scratch along with the task policy. "SAM-centroid", like RoboHop, uses segment centroids to represent each object segment.

We develop POCR to be the first reusable OCE that can be applied "off the shelf" to a large range of robotic environments and tasks. Concurrent with this work, SOFT [68] shows how OCEs can be constructed even from pre-trained vision models that do not directly output object segments.

V. Experimental Results

Our experiments aim to answer the following questions: 1) What is the appropriate "where" representation for POCR? 2) What is the appropriate "what" representation for POCR 3) Does POCR enable better policy learning compared to prior pre-trained representations or object-centric methods for robotics? 4) Does POCR enable systematic generalization? Video results and supplementary materials: sites.google.com/view/pocr.

A. Simulation Experiments Setup

Environments. We selected 7 tasks (Figure 3) in RL-Bench [69] as our simulation testbed to validate our algorithmic design. `Pick up Cup` and `Put Rubbish in Bin` are multi-object tasks with distractor objects. `Stack Wine`, `Phone on Base`, `Water Plants` are multi-object tasks that require reasoning about the geometry and affordance of objects to manipulate them into a desirable configuration. `Close Box` and `Close Laptop` are single-object tasks

that require reasoning about object articulations. The poses of all objects are randomized for each episode. In `Pick up Cup`, the color of the distractor cup is also randomized.

Policy Learning. Our policy training and evaluation protocol mostly follows [70]; in particular, for each task, we use 100 demonstrations collected using a state-based 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. For each method, we train policies using 3 seeds and report the mean and the standard error of the maximum success rates each seed achieves during training on 100 evaluation rollouts, following standard practice [30]. See Supplementary Materials I-A for more experimental details.

B. Investigating "Where" Representations for POCR

POCR's "where" representation involves Segment Anything Model (SAM) [28]. We first investigate the benefits of using SAM over prior approaches that train such segmenters in-domain. We compare SAM to the best such approach simulation: AST-SEG [19], an unsupervised method, and SAM [28], a pre-trained method. We train AST-SEG on our demonstrations in RLBench (about 1400 images for `Pick up Cup` and 2500 images for `Rubbish in Bin`) and use SAM directly out-of-the-box. We report the quantitative results with foreground adjusted random index (FG ARI) [71], [72], a standard segmentation metric. SAM achieves 0.99 FG ARI scores (max is 1) on both tasks, while AST-SEG fails to segment almost all the foreground objects, scoring only 0.2 on `Pick up Cup` and 0.1 on `Rubbish in Bin`. We reason that AST-SEG and prior segmentation methods [73], [74] typically require large domain-specific datasets for unsupervised training. This is unsuitable for sample-efficient policy learning in a new environment. Therefore, we employ SAM as the "where" representation for POCR. Figure 2 shows the visualizations of SAM masks after slot binding in various environments, illustrating that integrating SAM into our pipeline provides consistent and accurate tracking of objects. See Figure 4 and Supplementary Materials III for additional visualizations.

In Supplementary Materials II-A, we show that representing POCR's "where" component explicitly using SAM mask bounding box coordinates (**SAM-bbox**) leads to slightly better downstream control results than using SAM mask centroids (**SAM-centroid**). It also leads to significantly better results than using no explicit "where" representation, suggesting that the "where" information is not well captured implicitly in the "what" encodings. In Supplementary Materials II-D, we show that POCR's object binding procedure has near perfect accuracy when evaluated quantitatively using *ground-truth* masks. However, as presented in Supplementary Materials II-B, there still exists a small gap between policies trained with SAM masks and *ground-truth* masks.

C. Investigating "What" Representations for POCR

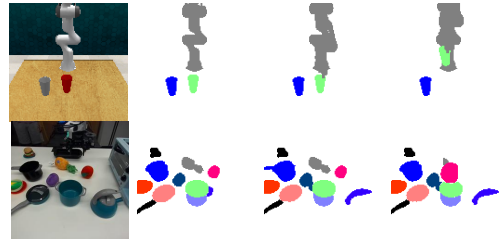**Representations.** Given that SAM masks, represented as the bounding box coordinates of each mask, provide

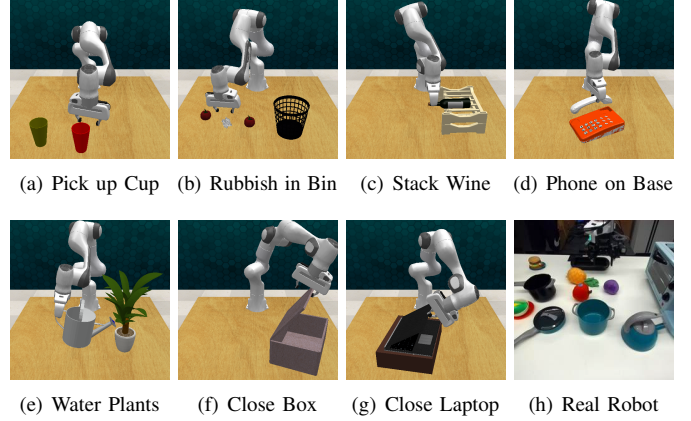Fig. 2. POCR segmentation results over demonstrations.

(a) Pick up Cup (b) Rubbish in Bin (c) Stack Wine (d) Phone on Base

(e) Water Plants (f) Close Box (g) Close Laptop (h) Real Robot

Fig. 3. Evaluation Environments.

the best "where" representation for POCR, we thoroughly investigate which "what" representation provides the best control performance by ablating various pre-trained vision encoders for control in simulation. **LIV** [9] and **R3M** [30] are visual representations for robot control pre-trained on large-scale in-the-wild human videos. **ImageNet** refers to a ResNet-50 network [75] pre-trained on the ImageNet dataset [10]. We also consider training a CNN network from scratch as the visual encoder of the masks (**Scratch**), using the official implementation from [70]. Finally, we consider using only the bounding box of SAM masks without additional "what" slot vector representations (**None**).

Results. As shown in Table I, when combined with SAM as the "what" representation in POCR, LIV performs best. Training from scratch (SAM-Scratch) performs significantly worse than all other "what" encoders, which validates the advantage of using pre-trained foundation models over in-domain training. Among the pre-trained "what" encoders, LIV yields a small but consistent advantage over R3M and ImageNet. We retain LIV as the "what" encoder of choice for all following experiments.

D. How Does POCR Compare to SOTA Representations?

To our knowledge, POCR representations are the first generic pre-trained object-centric representation for robotics. We now evaluate them against state-of-the-art pre-trained representations and prior object-centric methods.

Baselines. Our most relevant baselines are prior pre-trained methods, especially those for robotics. We again compare to **LIV**, **R3M**, and **ImageNet**. We also compare to **VIMA** [60], a state of the art object-centric baseline which parses images into object bounding boxes and learns transformer-based object representations from scratch. Instead

"What" Representations	Pick up Cup	Rubbish in Bin
SAM-LIV	**79.3 ± 0.7**	**35.3 ± 1.8**
SAM-R3M	78.0 ± 1.3	33.3 ± 1.3
SAM-ImageNet	70.7 ± 0.7	30.0 ± 2.0
SAM-Scratch	32.7 ± 2.9	12.0 ± 2.0
SAM-None	55.7 ± 1.8	24.6 ± 1.3

of following VIMA's setup to train an object detector with tens of thousands of in-domain mask annotations, we supply VIMA with ground-truth masks generated by RLBench. In spirit, **VIMA** is similar to **SAM-Scratch** in Table I. Finally, we again consider training a CNN network from scratch (**Scratch**) as the visual descriptor of the image observation.

Simulation Results. Table II shows the results. Each flat scene-level pre-trained representation (LIV, R3M, ImageNet) trails behind its POCR counterpart (SAM-LIV, SAM-R3M, SAM-ImageNet), reported in Table I, showing the versatility of the POCR framework. Naturally, as the best POCR, SAM-LIV far outperforms these flat encoders. Next, in our experiments, the SOTA object-centric approach VIMA incurred high compute and time costs for training, so we only trained it on 2 tasks, but the conclusions are clear: VIMA, which only pre-trains its object detectors and trains "what" representations from scratch performs poorly in this limited-demonstration setting. POCR with SAM for "where", and LIV for "what" thus outperforms the best current representations for multi-object manipulation. In Supplementary Materials II-C, we show that prior methods also fall far short of POCR when used as representations in BC-RNN [76], a popular approach for recurrent imitation policy learning. In Supplementary Materials II-F, we ablate the number of demonstrations and show that POCR continues to improve with more data whereas the baselines tend to plateau.

E. Real-World Experiments

Environment. To test our algorithm on real-world robotic manipulation tasks, we design an environment that consists of a counter-top kitchen setup, in which a Franka robot is tasked with placing various fruits, apple, eggplant, and 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 countertop scene, bringing the total number of objects to 10. We use a single 3rd-person monocular RGB camera for policy learning (see Figure 3(h) for the camera view). 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.

Policy Learning. As in our simulation experiments, we run behavior cloning with keyframe action representation. 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 the best performance for all methods; for POCR, 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. See Supplementary Materials I-B for more experimental details.

Real-World Results. For this real robot experiment, we evaluate the best-performing variant of POCR in simulation, SAM-LIV. See Figure 6 for results and Figure 4 for rollout visualizations. POCR (SAM-LIV) once again easily outperforms LIV. On the `apple` task, it achieves more than double the success rate. When trained without augmentation, LIV overfits the demonstrations and fails completely (0% success rate). Even in this very difficult setting, POCR (SAM-LIV) still achieves non-trivial performance. These results highlight the fragility of flat scene-level representations, even when they have been trained on large, diverse human videos. 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 demonstrations 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 have not been given the right input observations – the very issue that can be mitigated with our chaining approach that augments "what" foundation models by explicitly providing the "where" from an off-the-shelf segmentation model. See the project site for videos of POCR (SAM-LIV)'s real-world policy rollouts.

F. Systematic Generalization Experiments

Methods, Training & Evaluation. Prior works [34]–[38] have established systematic generalization, the model's ability to generalize to unseen scenarios different but semantically similar to the training data, as an advantage of OCEs. To validate the ability of POCR to achieve systematic generalization, we perform experiments in the real-world and simulated task setting shown in Figure 5. In Figure 5(a), the robot must pick up the apple and put it in the green pot, but there is one other "distractor" fruit, the green pear, in the scene. In Figure 5(a), the robot needs to perform the same task as above, but now in the presence of a new background in the form of a blue cloth. And finally, in Figure 5(c), the robot must pick up the red cup, but we introduced two unseen distractors, a carrot and a banana. We evaluated the same policies as in the earlier sections in these systematic generalization environments. All new distractor objects were also initialized to random positions.

Results. In every real-world and simulated evaluation setting, POCR (SAM-LIV) policies are largely unaffected, but the baseline LIV performs significantly worse. In the real-world setting with the new distractor fruit, the success rate of POCR (SAM-LIV) only drops from 70% to 60%, whereas LIV drops from 40% to 20%. These trends are even more pronounced in the new background setting (also in real-world): again, POCR (SAM-LIV) is only marginally

| Representations	Multi-Object with Distractors		Multi-Object with Geometric Reasoning			Articulated Objects	
	Pick up Cup	Rubbish in Bin	Stack Wine	Phone on Base	Water Plants	Close Box	Close Laptop
POCR (SAM-LIV)	**79.3 ± 0.7**	**35.3 ± 1.8**	**54.0 ± 2.3**	**34.7 ± 3.3**	**16.3 ± 0.9**	**96.7 ± 0.7**	32.7 ± 1.3
LIV	49.3 ± 1.9	21.9 ± 0.8	24.0 ± 5.0	18.7 ± 2.4	4.0 ± 1.2	86.0 ± 5.3	**35.3 ± 1.8**
R3M	52.3 ± 1.5	20.7 ± 1.8	40.0 ± 2.0	27.3 ± 2.4	8.0 ± 2.0	85.3 ± 1.3	31.3 ± 1.3
ImageNet	14.0 ± 1.2	6.7 ± 0.7	11.3 ± 2.4	7.3 ± 1.3	7.3 ± 0.7	58.7 ± 1.3	30.0 ± 2.0
VIMA (GT masks)	31.0 ± 6.7	5.7 ± 4.0	N/A	N/A	N/A	N/A	N/A
Scratch	15.9 ± 1.0	2.7 ± 0.4	1.3 ± 0.7	3.3 ± 0.7	2.0 ± 0.0	86.7 ± 1.8	30.0 ± 2.3
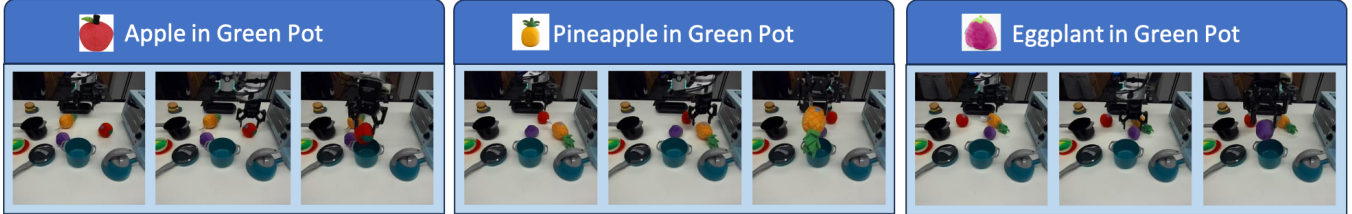
Fig. 4. Real-World Policy Rollouts.
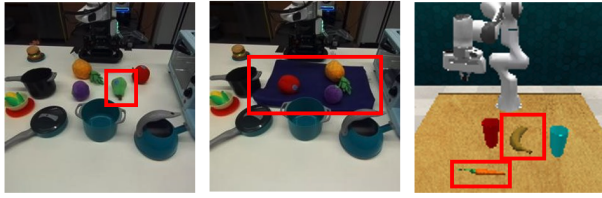
(a) New pear   (b) New blue cloth   (c) New food objects

Fig. 5. Systematic Generalization Evaluation Environment. Figure 5(a): green pear is the new distractor fruit; Figure 5(b): blue cloth serves as new background; Figure 5(c) Carrot and banana are new distractors.
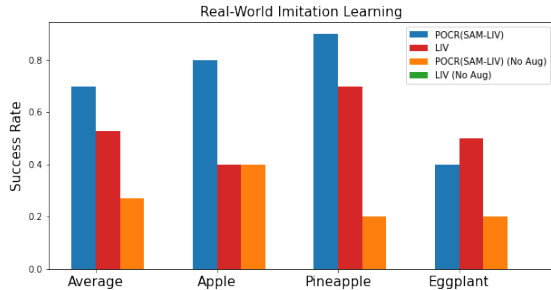

Fig. 6. Real-World Imitation Learning Results.

affected, dropping from 70% to 50%, while LIV, unable to adapt to this out-of-distribution scenario, drops to 0%. In our experiments, the LIV policy, unable to handle the out-of-distribution visual observations, consistently performs meaningless and dangerous actions never seen in training, hyper-extending the robot arm, and nearly causing damage to the experimental setup. POCR (SAM-LIV) degrades much more gracefully. Likewise, in the simulated settings with two new distractor objects, POCR (SAM-LIV) performance decreases only marginally from 79.3 ± 0.7 to 76.0 ± 3.1. In comparison, the image LIV policy dropped in performance from 49.4 ± 1.9 to 27.0 ± 0.6. These results illustrate the natural advantages of POCR for systematic generalization.

VI. LIMITATION AND FUTURE WORKS

POCR policy inference speed is slow due to using the large, slow SAM model. In settings that require high-frequency policy actions, faster segmentation methods that trade off speed with accuracy may be necessary for practical utility. Our procedure for localizing and assigning objects to slots (Section III-A) has a set of parameters. In our experiments, we use a unified set of parameters that we believe are generally applicable. But in an arbitrary new environment, additional tuning may be required. Furthermore, we follow prior works on pre-trained robotics representations in evaluating POCR in tasks that largely have the same set of relevant objects in all train and test episodes. Testing for generalization over different manipulated objects, alongside our systematic generalization experiments above, may provide more thorough insights into the quality of our method and baselines.

VII. CONCLUSION

We have presented a simple yet effective recipe for generating reusable object-centric representations for visual robotic manipulation from plugging and playing "what" and "where" foundation models. Our framework uses off-the-shelf instance segmentation to produce high-quality, self-consistent masks over time and uses off-the-shelf visual representations to acquire fine-grained visual descriptors for policy learning. Instantiated using state-of-art visual foundation models, POCR substantially outperforms the state-of-art representations in both simulation and the real world without any in-domain object-centric representation learning. Since POCR can flexibly plug and play with any "what" and "where" visual foundation models, it has the potential to distill knowledge from more diverse datasets through better pre-trained models in the future.

REFERENCES

[1] T. Brown, B. Mann, N. Ryder, *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[3] A. Radford, J. W. Kim, C. Hallacy, *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, PMLR, 2021, pp. 8748–8763.

[4] R. Girdhar, A. El-Nouby, Z. Liu, *et al.*, "Imagebind: One embedding space to bind them all," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 15 180–15 190.

[5] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 000–16 009.

[6] T. Xiao, I. Radosavovic, T. Darrell, and J. Malik, "Masked visual pre-training for motor control," *arXiv preprint arXiv:2203.06173*, 2022.

[7] S. Nair, E. Mitchell, K. Chen, S. Savarese, C. Finn, *et al.*, "Learning language-conditioned robot behavior from offline data and crowd-sourced annotation," in *Conference on Robot Learning*, PMLR, 2022, pp. 1303–1315.

[8] Y. J. Ma, S. Sodhani, D. Jayaraman, O. Bastani, V. Kumar, and A. Zhang, "Vip: Towards universal visual reward and representation via value-implicit pre-training," *arXiv preprint arXiv:2210.00030*, 2022.

[9] Y. J. Ma, W. Liang, V. Som, *et al.*, "Liv: Language-image representations and rewards for robotic control," *arXiv preprint arXiv:2306.00958*, 2023.

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, Ieee, 2009, pp. 248–255.

[11] K. Grauman, A. Westbury, E. Byrne, *et al.*, "Ego4d: Around the world in 3,000 hours of egocentric video," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 995–19 012.

[12] M. A. Goodale and A. D. Milner, "Separate visual pathways for perception and action," *Trends in neurosciences*, vol. 15, no. 1, pp. 20–25, 1992.

[13] L. G. Ungerleider and J. V. Haxby, "'what'and 'where'in the human brain," *Current opinion in neurobiology*, vol. 4, no. 2, pp. 157–165, 1994.

[14] E. H. de Haan and A. Cowey, "On the usefulness of 'what'and 'where'pathways in vision," *Trends in cognitive sciences*, vol. 15, no. 10, pp. 460–466, 2011.

[15] F. Locatello, D. Weissenborn, T. Unterthiner, *et al.*, "Object-centric learning with slot attention," *ArXiv*, vol. abs/2006.15055, 2020.

[16] C. P. Burgess, L. Matthey, N. Watters, *et al.*, "Monet: Unsupervised scene decomposition and representation," *ArXiv*, vol. abs/1901.11390, 2019.

[17] M. Engelcke, O. P. Jones, and I. Posner, "Genesis-v2: Inferring unordered object representations without iterative refinement," in *Neural Information Processing Systems*, 2021.

[18] Z. Lin, Y.-F. Wu, S. V. Peri, *et al.*, "Space: Unsupervised object-oriented scene representation via spatial attention and decomposition," *ArXiv*, vol. abs/2001.02407, 2020.

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

[20] T. Kipf, E. van der Pol, and M. Welling, "Contrastive learning of structured world models," *ArXiv*, vol. abs/1911.12247, 2019.

[21] C. Devin, P. Abbeel, T. Darrell, and S. Levine, "Deep object-centric representations for generalizable robot learning," *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7111–7118, 2017.

[22] D. Wang, C. Devin, Q.-Z. Cai, F. Yu, and T. Darrell, "Deep object-centric policies for autonomous driving," *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8853–8859, 2018.

[23] S. Kumar, J. Zamora, N. Hansen, R. Jangir, and X. Wang, "Graph inverse reinforcement learning from diverse videos," in *Conference on Robot Learning*, 2022.

[24] Y. Zhu, A. Joshi, P. Stone, and Y. Zhu, "Viola: Imitation learning for vision-based manipulation with object proposal priors," *ArXiv*, vol. abs/2210.11339, 2022.

[25] M. Sieb, X. Zhou, A. Huang, O. Kroemer, and K. Fragkiadaki, "Graph-structured visual imitation," in *Conference on Robot Learning*, 2019.

[26] T. D. Kulkarni, A. Gupta, C. Ionescu, *et al.*, "Unsupervised learning of object keypoints for perception and control," *ArXiv*, vol. abs/1906.11883, 2019.

[27] M. Minderer, C. Sun, R. Villegas, F. Cole, K. P. Murphy, and H. Lee, "Unsupervised learning of object structure and dynamics from videos," *ArXiv*, vol. abs/1906.07889, 2019.

[28] A. Kirillov, E. Mintun, N. Ravi, *et al.*, "Segment anything," *ArXiv*, vol. abs/2304.02643, 2023.

[29] X. Zou, J. Yang, H. Zhang, *et al.*, "Segment everything everywhere all at once," *ArXiv*, vol. abs/2304.06718, 2023.

[30] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta, "R3m: A universal visual representation for robot manipulation," *arXiv preprint arXiv:2203.12601*, 2022.

[31] I. Radosavovic, T. Xiao, S. James, P. Abbeel, J. Malik, and T. Darrell, "Real-world robot learning with masked visual pre-training," in *Conference on Robot Learning*, PMLR, 2023, pp. 416–426.

[32] A. Majumdar, K. Yadav, S. Arnaud, *et al.*, "Where are we in the search for an artificial visual cortex for embodied intelligence?" In *Workshop on Reincarnating Reinforcement Learning at ICLR 2023*, 2023.

[33] N. Heravi, A. Wahid, C. Lynch, *et al.*, "Visuomotor control in multi-object scenes using object-aware representations," *arXiv preprint arXiv:2205.06333*, 2022.

[34] S. van Steenkiste, K. Greff, and J. Schmidhuber, "A perspective on objects and systematic generalization in model-based rl," *arXiv preprint arXiv:1906.01035*, 2019.

[35] K. Greff, S. van Steenkiste, and J. Schmidhuber, "On the binding problem in artificial neural networks," Dec. 2020. arXiv: 2012.05208 [cs.NE].

[36] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman, "Building machines that learn and think like people," en, *Behav. Brain Sci.*, pp. 1–101, Nov. 2016.

[37] A. Dittadi, S. S. Papa, M. De Vita, B. Schölkopf, O. Winther, and F. Locatello, "Generalization and robustness implications in object-centric learning," in *Proceedings of the 39th International Conference on Machine Learning*, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds., ser. Proceedings of Machine Learning Research, vol. 162, PMLR, 2022, pp. 5221–5285. [Online]. Available: https://proceedings.mlr.press/v162/dittadi22a.html.

[38] J. Yoon, Y.-F. Wu, H. Bae, and S. Ahn, "An investigation into pre-training object-centric representations for reinforcement learning," *arXiv preprint arXiv:2302.04419*, 2023.

[39] T. Kipf, G. F. Elsayed, A. Mahendran, *et al.*, "Conditional object-centric learning from video," *ArXiv*, vol. abs/2111.12594, 2021.

[40] S. Haldar, J. Pari, A. K. Rai, and L. Pinto, "Teach a robot to fish: Versatile imitation from one minute of demonstrations," *ArXiv*, vol. abs/2303.01497, 2023.

[41] S. Young, J. Pari, P. Abbeel, and L. Pinto, "Playful interactions for representation learning," *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 992–999, 2021.

[42] T. Xiao, H. Chan, P. Sermanet, *et al.*, "Robotic skill acquisition via instruction augmentation with vision-language models," *arXiv preprint arXiv:2211.11736*, 2022.

[43] M. Engelcke, O. P. Jones, and I. Posner, "Reconstruction bottlenecks in Object-Centric generative models," Jul. 2020. arXiv: 2007.06245 [cs.LG].

[44] S. Papa, O. Winther, and A. Dittadi, "Inductive biases for object-centric representations in the presence of complex textures," in *UAI 2022 Workshop on Causal Representation Learning*, 2022.

[45] Y. Yang and B. Yang, "Promising or elusive? unsupervised object segmentation from real-world single images," *NeurIPS*, 2022.

[46] S. Amir, Y. Gandelsman, S. Bagon, and T. Dekel, "Deep vit features as dense visual descriptors," *ArXiv*, vol. abs/2112.05814, 2021.

[47] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics (NRL)*, vol. 52, 1955.

[48] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola, "Deep sets," *Advances in neural information processing systems*, vol. 30, 2017.

[49] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[50] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[51] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[52] A. Zhou, V. Kumar, C. Finn, and A. Rajeswaran, "Policy architectures for compositional generalization in control," *arXiv preprint arXiv:2203.05960*, 2022.

[53] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, pp. 91–110, 2004.

[54] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, Ieee, vol. 1, 2005, pp. 886–893.

[55] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield, "Deep object pose estimation for semantic robotic grasping of household objects," in *Conference on Robot Learning*, 2018.

[56] Y. Ye, D. Gandhi, A. K. Gupta, and S. Tulsiani, "Object-centric forward modeling for model predictive control," *ArXiv*, vol. abs/1910.03568, 2019.

[57] T. Migimatsu and J. Bohg, "Object-centric task and motion planning in dynamic environments," *IEEE Robotics and Automation Letters*, vol. 5, pp. 844–851, 2019.

[58] Y. Zhu, J. Tremblay, S. Birchfield, and Y. Zhu, "Hierarchical planning for long-horizon manipulation with geometric and symbolic scene graphs," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 6541–6548.

[59] J. Mao, T. Lozano-Pérez, J. B. Tenenbaum, and L. P. Kaelbling, "Pdsketch: Integrated planning domain programming and learning," *arXiv preprint arXiv:2303.05501*, 2023.

[60] Y. Jiang, A. Gupta, Z. Zhang, *et al.*, "Vima: General robot manipulation with multimodal prompts," *arXiv preprint arXiv:2210.03094*, 2022.

[61] T. Silver, A. Athalye, J. B. Tenenbaum, T. Lozano-Perez, and L. P. Kaelbling, "Learning neuro-symbolic skills for bilevel planning," *arXiv preprint arXiv:2206.10680*, 2022.

[62] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask r-cnn," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, pp. 386–397, 2017.

[63] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 1137–1149, 2015.

[64] S. Ferraro, P. Mazzaglia, T. Verbelen, and B. Dhoedt, "Focus: Object-centric world models for robotics manipulation," *arXiv preprint arXiv:2307.02427*, 2023.

[65] J. Yang, W. Tan, C. Jin, *et al.*, "Pave the way to grasp anything: Transferring foundation models for universal pick-place robots," *arXiv preprint arXiv:2306.05716*, 2023.

[66] Y. Zhu, Z. Jiang, P. Stone, and Y. Zhu, "Learning generalizable manipulation policies with object-centric 3d representations," in *7th Annual Conference on Robot Learning*, 2023.

[67] S. Garg, K. Rana, M. Hosseinzadeh, *et al.*, "Robohop: Segment-based topological map representation for open-world visual navigation," in *CoRL 2023 Workshop on Learning Effective Abstractions for Planning (LEAP)*, 2023.

[68] J. Qian, A. Panagopoulos, and D. Jayaraman, "Recasting generic pretrained vision transformers as object-centric scene encoders for manipulation policies," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[69] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison, "Rlbench: The robot learning benchmark & learning environment," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3019–3026, 2020.

[70] S. James and A. J. Davison, "Q-attention: Enabling efficient learning for vision-based robotic manipulation," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1612–1619, 2022.

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

[72] L. J. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification*, vol. 2, pp. 193–218, 1985.

[73] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[74] X. Wang, A. Shrivastava, and A. Gupta, "A-fast-rcnn: Hard positive generation via adversary for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2606–2615.

[75] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[76] A. Mandlekar, D. Xu, J. Wong, *et al.*, "What matters in learning from offline human demonstrations for robot manipulation," *arXiv preprint arXiv:2108.03298*, 2021.