Visuomotor Control in Multi-Object Scenes Using Object-Aware Representations

Negin Heravi[1]* **Ayzaan Wahid**[2] **Corey Lynch**[2] **Pete Florence**[2] **Travis Armstrong**[2]
Jonathan Tompson[2] **Pierre Sermanet**[2] **Jeannette Bohg**[1] **Debidatta Dwibedi**[2]
[1]Stanford University, [2]Robotics at Google

Abstract: Perceptual understanding of the scene and the relationship between its different components is important for successful completion of robotic tasks. Representation learning has been shown to be a powerful technique for this, but most of the current methodologies learn task specific representations that do not necessarily transfer well to other tasks. Furthermore, representations learned by supervised methods require large labeled datasets that are expensive to collect in the real world. Using self-supervised learning to obtain representations from unlabeled data can mitigate this problem. In this paper, we show the effectiveness of using self-supervised object-aware representation learning techniques for robotic tasks. Our representations are learned by observing the agent freely interacting with different parts of the environment and is queried in two different settings: (i) policy learning and (ii) object location prediction. We show that our model learns control policies in a sample-efficient manner and outperforms state-of-the-art object agnostic techniques as well as methods trained on raw RGB images. Our results show a 20% increase in performance in low data regimes (1000 trajectories) in policy training using implicit behavioral cloning (IBC). Furthermore, our method outperforms the baselines for the task of multi-object localization.

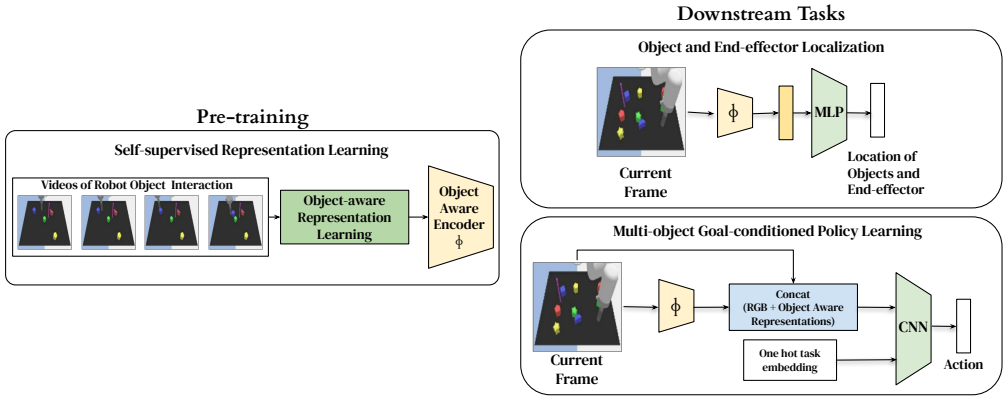

Figure 1: We train a perception encoder using object-aware self-supervised learning. Then, we freeze the encoder weights and demonstrate that it is suitable for downstream tasks of object localization and multi-object goal-conditioned policy learning.

1 Background

Prior work in robotics has shown that performance and sample-efficiency of policy learning improves with self-supervised scene representations. These methods use compact representations in form of a global embedding [1], sparse key-point coordinates [2, 3, 4, 5], or other object embeddings [6] as input to policy learning modules. In this work, we explore a class of self-supervised models called Slot Attention [7] for representation learning in a robotic setup and differ from these prior works by learning the policy directly on dense per-pixel features and object masks produced by the Slot Attention model. This is particularly important in the context of multi-object manipulation as the representations need to encode the location of multiple objects in the scene along with the end-effector. Additionally, learning slot based representations does not require multi-view cameras [1, 4, 5] or canonical images of objects [3, 6].

Slot Attention models use sequential attention based mechanisms to group low level features in a scene where each group falls into a slot bin [7]. This enables them to segment objects in an

*Work done as an intern at Google. Toyota Research Institute ("TRI") provided partial funds to support this work, but this article solely reflects the opinions and conclusions of its authors and not TRI or any other Toyota entity.

CoRL 2022 Workshop on Pre-training Robot Learning, Auckland, New Zealand.

unsupervised manner. Inspired by this architecture, we propose to use these models to learn the extent of multiple objects in a scene and to use their corresponding representations for a variety of downstream robotic tasks. Our hypothesis is that the abstracted information using Slot Attention can improve data sample efficiency and performance in downstream training since it is object-aware making it suitable for extracting information in multi-object scenes. We test this hypothesis in the tasks of object localization and multi-object goal-conditioned policy learning. Our model is trained in multiple stages. First, we train Slot Attention in the object discovery mode in a self-supervised manner. Then, we freeze the weights and use these learned representations to train different small downstream networks for each task. Using this setup, we study the gain in performance by using these representations in different data regimes. The mask and features learned by our model are able to boost performance on both tasks of object localization and behavioral cloning. Particularly in the low data regime, our features result in a 20% improvement in task completion success rate.

In summary, the contributions of our work are: (1) We show that our Slot Attention inspired representations encode location and properties of all objects in the scene while object agnostic self-supervised methods such as MoCo [8] only focus on a few objects. (2) We show that our method needs fewer supervised action labels to learn policies (i.e. it is more sample efficient) and learns policies that have faster training convergence than alternative state-of-the-art methods.

2 Approach

Our framework learns object aware representations from unlabeled videos using Slot Attention [7]. We then freeze the weights of the representation architecture, and use features from this model for downstream robotic tasks of object and end-effector localization (Section 3.1) as well as policy learning (Section 3.2). Figure 1 shows an overview of our approach. As our representation method, we use an image encoder ϕ that takes an RGB image I as input, and outputs an embedded feature representation $\phi(I)$. We train ϕ using a variation of the Slot Attention network [7]. This architecture groups the features of an image into K slots where K is a hyperparameter. The attention [9] mechanism is normalized over the K slots. This makes the slots compete with each other and each specialize in explaining a different component in the image resulting in self-supervised decomposition of low-level image features into abstract groups. We modified the Slot Attention architecture in two ways. First, like [10], we initialize the slots to be learnable fixed vectors instead of samples from a learned Gaussian distribution to mitigate slot swapping. Second, we use convolutions followed by upsampling instead of transposed convolutions to prevent checkerboard artifacts [11].

After training the Slot Attention network, we freeze the weights and use the output of the convolution-based encoder as representation in our policy learning. The pre-trained frozen Slot model used in our experiments were trained for $500k$ steps with a batch size of 8 on images of size 160 by 320 and $K = 16$ slots with random seed initialization unless otherwise noted. For the experiments where the fraction of the data available varies, the slot representations are also only trained on the selected fraction of data. For the localization task, we use the dense masks $(M_1, .., M_K)$ as input to the downstream network. These masks encode the location of objects in pixel space.

Environment We train our models using data of a robot interacting with blocks of different shapes and colors placed on a table in a simulation environment in PyBullet [12]. In our setup, a robot arm is attached to a fixed base such that it can push objects in front of it on a table using a cylindrical end-effector. The robot arm is constrained to move on a 2D plane (similar to [13]). We use this environment for collecting the data for all the simulation experiments in the following sections.

3 Experiments and Discussion

Our goal is to evaluate whether self-supervised object aware representations learned using Slot Attention provide performance gain for robotic tasks. To quantify this, we compare various representation learning techniques for tasks of multi-object localization and goal-conditioned policy learning.

Baseline We train a MoCo[8]-based architecture as well as an autoencoder as baseline. For MoCo, we train an encoder using a contrastive loss in a self-supervised manner similar to [14]. The output of the encoder is spatially averaged to produce an embedding for downstream tasks. We use this method to compare the performance of contrastive losses and reconstruction losses (like that used in Slot Attention) for ours tasks. We use an embedding size of 128, queue size of 16384, softmax temperature of 0.1 and batch size of 16 for this encoder. For our autoencoder baseline, we train an encoder-decoder architecture using the the same loss as our method but with no notion of slots in the architecture. The objective of this baseline comparison is isolating the importance of the Slot Attention module and the reconstruction loss for learning representations.

3.1 Object Localization

Task and Metrics In this experiment, we investigate if our method can encode information about *all* the objects present in a scene. This property is important for learning policies on datasets with

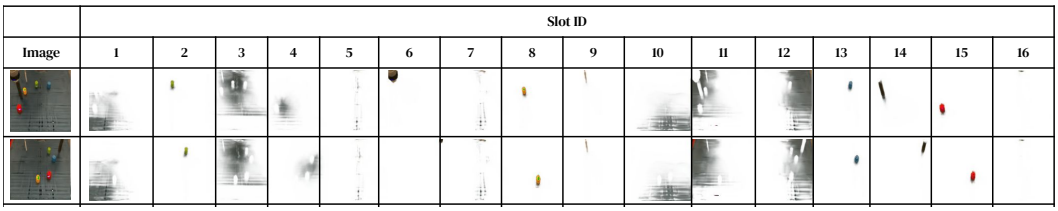

Figure 3: Qualitative example of masks learned by Slot Attention on real world data. Slot Attention is able to discover the pixels corresponding to the blocks (slots 2,8,13,15), the robot arm (slot 6), the end-effector (slot 14), and the pole (slot 9) without any labels.

multiple objects and for tasks that depend on full scene information such as object localization. We evaluate the representations on the object localization task in a simulated environment which provides ground truth object locations. We only use this ground truth informtation during downstream training not for representation learning. We learn representations from a dataset of demonstrations collected in simulation environment described in section 2. Then, we freeze the weights of the representation network and train a two layer MLP with size of 256 to predict the 2D location of the center of each block and the end-effector in robot coordinates. For our method in this task, the input to our downstream MLP is the center of mass of the predicted slot masks. For our MoCo baseline, we use the output of the image encoder as input. For our Autoencoder baseline, we use global average pooling layer on top of the output of the encoder CNN as input.

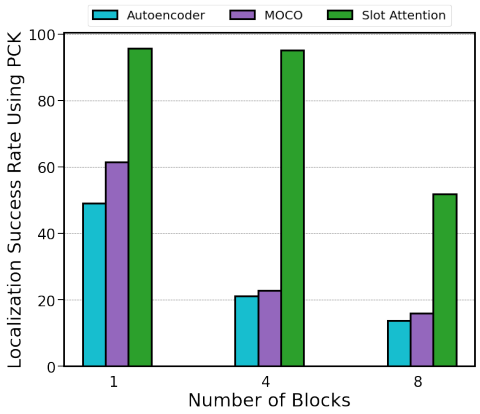

Figure 2: Performance comparison on object localization over number of blocks in the scene using *Probability of Correct Keypoint* as metric. The baselines are able to learn the location of one object in the scene (the end-effector) resulting in a low performance on average that decreases as the number of blocks increases. Slot Attention is able to localize multiple objects but sometimes struggles with objects of same color but different shapes in the 8 block case.

As an interpretable evaluation metric, we use *Probability of Correct Keypoint* [15] which captures the percentage of times an object location is predicted correctly. This metric considers an object correctly localized if the predicted coordinates are within a given threshold of the ground truth. We chose a threshold value of 0.1 of the length of the table.

We test the performance of our method with 1, 4, and 8 blocks. For the Slot Attention models, we use 7, 11, and 11 slots for each case respectively. The number of slot were chosen based on their performance on the reconstruction loss when pretraining the model independent of the downstream task. We train models with a dataset of 160k trajectories with image size of 256×256 for 150k steps each with batch size of 16 on 1 V100 GPU. We use the checkpoint with the lowest loss during representation learning for downstream training.

Results Slot Attention outperformed the baselines in all the object localization experiments specially in multi-object cases as seen in Figure 2. Furthermore, we observed that while MoCo and Autoencoder are able to learn to predict object location more accurately on a dataset with a single object, they struggle to encode object locations in multi object scenes. We also observed that the Slot Attention model finds it difficult to localize blocks of the same color but with fine-grained differences in their shape. In the 8 block case, the Slot Attention model gave poor localization performance for the two yellow and the two green objects that have similar shapes. This is due to Slot being trained using a pixel-wise image reconstruction loss and hence struggling to differentiate subtle differences. Subtle shape differences only contribute to a small number of pixel differences only resulting in slight changes in the loss. However, despite this, Slot Attention still outperforms the other methods for this task in all cases.

3.2 Multi-object Goal-conditioned Policy Learning

Task and Metrics In this experiment, we compare different representation learning techniques by studying their effectiveness as inputs to a policy learning method. We only consider the imitation learning setup and learn a policy from a dataset of demonstrations provided by experts. The task for these experiments is to manipulate one of 8 blocks (i.e. the target block) on the plane to the target location shown by a purple rod. When the block is within 0.05 units of the rod, the episode is considered a success. If the robot fails to move the target block to the rod in 200 steps then the episode is considered a failure. We use Implicit Behavior Cloning [16] to learn policies. During evaluation, we run the policy for 200 different initial configurations and measure the number of times the policy was

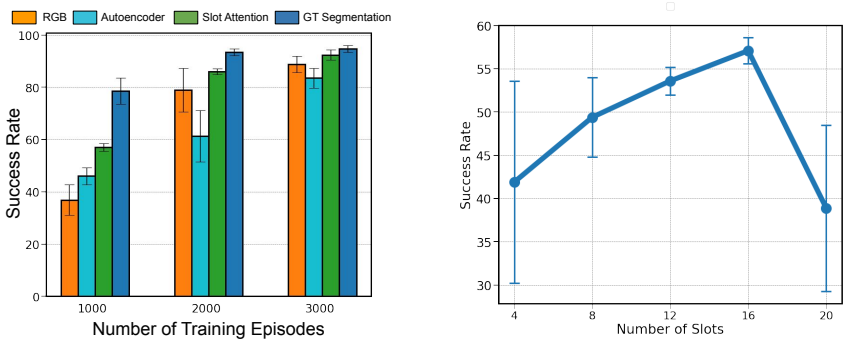

Figure 4: (left) Validation performance comparison on policy learning of different methods over episodes in training data. Slot Attention based representations provide a performance boost in the low data regimes. Slot Attention leads to a 20% performance increase in performance in low data regime (1000 episodes). Using ground truth segmentation masks is shown as an upper bound. Solid lines show the mean across 4 seeds and the shaded area indicates 1 standard deviation from each side. (right) Effect of number of slots on IBC policy performance. Here we are using Slot Attention features in 1000 episodes data regime.

able to successfully move the required block to the target location within the tolerated distance. We run policy training with 4 random seeds and report the mean and standard deviation of the success rates over the 4 runs. To compare different methods, we keep the policy learning method the same while we vary the input representations between RGB, RGB+Ground Truth (GT) Segmentation, Slot Attention, and Autoencoder. For fair comparison between the different techniques, we take the penultimate layer of the CNN encoder (before the spatial average pooling) and resize it to match the input RGB image as input to the policy learning network. We also experimented with using MoCo features as input but were not able to train it to convergence. MoCo is trained to minimize a contrastive loss, and to succeed at minimizing this loss, the final representation does not need to capture information about all the objects in the scene. MoCo's loss can be minimized by only focusing on objects that move more often, like the robot arm. The lack of convergence we observed for MoCo applied to IBC is likely due to MoCo features not reliably localizing the rod which is needed to solve the task. Please refer to [16] for details on the IBC training. In the following experiments, we used a slot model trained with 16 bins since it had the lowest evaluation reconstruction loss during Slot Attention training. All models were trained to convergence.

Results We make the following observations (Figure 4 (left)):

(1) Better perception inputs lead to more sample efficient policies. By using the GT semantic segmentation for all the blocks as input, the policy learning method can learn high-performing policies with over 90 percent success rate with few samples (2000 episodes). However, the policy with RGB needs somewhere between 3000 and 10000 episodes to achieve the same performance.
(2) Slot Attention provides performance boost in low data regimes. We observe that Slot Attention models provide a boost in performance in the success rate of task completion over using raw RGB as input. Slot Attention models are object aware and by using this prior we are able to learn representations from demonstration video datasets that can result in performance improvement without collecting object bounding boxes or segmentation masks from humans. We also note that the performance gain of using Slot Attention over baselines decreases as the number of samples available for learning the policy increases as shown in figure 4 (left).
(3) Slot Attention performs better than Autoencoder. We find that slot has better performance than autoencoder which has the same loss as the Slot Attention model but not the object/slot prior in its architecture. This shows that the prior of objects/slots is important for the performance gains.

Ablation: effect of number of slot bins The number of slots is an important design choice in the Slot Attention architecture. This hyper-parameter can be set by looking at the Slot reconstruction loss during the representation pretraining as well as evaluating on a validation set for downstream tasks. Figure 4 (right) shows the effect of varying the number of slots (K in Section 2).

4 Conclusion

We presented a method to improve performance of multi-object goal-conditioned behavior cloning policies using the Slot Attention architecture. We find that features and masks from this model are especially useful in the low data regime which is especially pertinent to deploying machine learning models on real-world robots. As preliminary evidence for real world applications, we show a qualitative example of the Slot Attention algorithm successfully localizing objects in a real scene trained with no labels using demonstration data in Figure 3. Even though the hole pattern on the table and the lighting complicates the task, Slot Attention is still able discover all the objects.

References

[1] P. Sermanet, C. Lynch, Y. Chebotar, J. Hsu, E. Jang, S. Schaal, and S. Levine. Time-contrastive networks: Self-supervised learning from video. *Proceedings of International Conference in Robotics and Automation (ICRA)*, 2018. URL http://arxiv.org/abs/1704.06888.

[2] C. Finn, X. Y. Tan, Y. Duan, T. Darrell, S. Levine, and P. Abbeel. Deep spatial autoencoders for visuomotor learning. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 512–519. IEEE, 2016.

[3] T. Kulkarni, A. Gupta, C. Ionescu, S. Borgeaud, M. Reynolds, A. Zisserman, and V. Mnih. Unsupervised learning of object keypoints for perception and control. *arXiv preprint arXiv:1906.11883*, 2019.

[4] P. Florence, L. Manuelli, and R. Tedrake. Self-supervised correspondence in visuomotor policy learning. *IEEE Robotics and Automation Letters*, 5(2):492–499, 2019.

[5] L. Manuelli, Y. Li, P. Florence, and R. Tedrake. Keypoints into the future: Self-supervised correspondence in model-based reinforcement learning. *arXiv preprint arXiv:2009.05085*, 2020.

[6] W. Yuan, C. Paxton, K. Desingh, and D. Fox. SORNet: Spatial object-centric representations for sequential manipulation. In *5th Annual Conference on Robot Learning*, 2021. URL https://openreview.net/forum?id=mOLu2rODIJF.

[7] F. Locatello, D. Weissenborn, T. Unterthiner, A. Mahendran, G. Heigold, J. Uszkoreit, A. Dosovitskiy, and T. Kipf. Object-centric learning with slot attention. NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

[8] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[10] C. Yang, H. Lamdouar, E. Lu, A. Zisserman, and W. Xie. Self-supervised video object segmentation by motion grouping. *arXiv preprint arXiv:2104.07658*, 2021.

[11] A. Odena, V. Dumoulin, and C. Olah. Deconvolution and checkerboard artifacts. *Distill*, 1 (10):e3, 2016.

[12] E. Coumans and Y. Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. 2016.

[13] C. Lynch, A. Wahid, J. Tompson, T. Ding, J. Betker, R. Baruch, T. Armstrong, and P. Florence. Interactive language: Talking to robots in real time, 2022. URL https://arxiv.org/abs/2210.06407.

[14] X. Chen, H. Fan, R. Girshick, and K. He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[15] Y. Yang and D. Ramanan. Articulated human detection with flexible mixtures of parts. *IEEE transactions on pattern analysis and machine intelligence*, 35(12):2878–2890, 2012.

[16] P. Florence, C. Lynch, A. Zeng, O. Ramirez, A. Wahid, L. Downs, A. Wong, J. Lee, I. Mordatch, and J. Tompson. Implicit behavioral cloning. *Conference on Robot Learning (CoRL)*, November 2021.