VIOLA: Object-Centric Imitation Learning for Vision-Based Robot Manipulation

Anonymous Author(s)
Affiliation
Address
email

Abstract: We introduce VIOLA, an object-centric imitation learning approach to learning closed-loop visuomotor policies for robot manipulation. Our approach constructs object-centric representations based on general object proposals from a pre-trained vision model. It uses a transformer-based policy to reason over these representations and attends to the task-relevant visual factors for action prediction. Such object-based structural priors improve deep imitation learning algorithm's robustness against object variations and environmental perturbations. We quantitatively evaluate VIOLA in simulation and on real robots. VIOLA outperforms the state-of-the-art imitation learning methods by 45.8% in success rates. It has also been deployed successfully on a physical robot to solve challenging long-horizon tasks, such as dining table arrangements and coffee making. More videos and model details can be found in supplementary materials and the anonymous project website: https://sites.google.com/view/viola-manipulation.

Keywords: Imitation Learning, Manipulation, Object-Centric Representations

1 Introduction

Vision-based manipulation is a critical ability for autonomous robots to interact with everyday environments. It requires the robots to understand the unstructured world through visual perception to determine intelligent behaviors. In recent years, deep imitation learning (IL) [1–4] has emerged as a powerful approach to training visuomotor policies on diverse offline data, particularly human demonstrations. Its success of it stems from the effectiveness of training over-parameterized neural networks end-to-end with supervised learning objectives. These models excel at mapping raw visual observations to motor actions without manual engineering. While deep IL methods often distinguish themselves from reinforcement learning counterparts in their scalability to long-horizon tasks, a burgeoning body of recent work pointed out that IL methods lack robustness to covariate shifts and environmental perturbations [5–8]. End-to-end visuomotor policies are likely to falsely associate actions with task-irrelevant visual factors, leading to poor generalization in new situations.

In this work, we endow imitation learning algorithms with awareness about objects and their interactions to improve their efficacy and robustness in vision-based manipulation tasks. As cognitive science studies suggest, explaining a visual scene as objects and their interactions facilitates humans to learn fast and make accurate predictions [9–11]. Inspired by these findings, we hypothesize that decomposing a visual scene into factorized representations of objects in the scenes would enable robots to reason about the manipulation workspace in a modular fashion and improve their generalization ability. To this end, we develop an *object-centric imitation learning* approach, which infuses structural object-based priors into the model architecture of visuomotor policies. Training policies with these priors would make it easier for the model to focus on task-relevant visual cues while discarding spurious dependencies.

The first and foremost challenge of such an object-centric approach is to determine what constitutes an *object* and how objects are represented. The definitions of objects are often fluid and task-dependent for manipulation tasks. This work studies the notions of objects operationally and

Submitted to the 6th Conference on Robot Learning (CoRL 2022). Do not distribute.

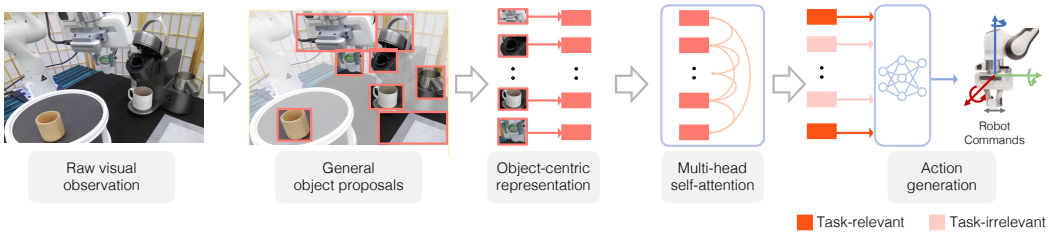


Figure 1: VIOLA first obtains a set of general object proposals from raw visual observations. It extracts object features from the proposals to build the object-centric representation. The transformer-based policy uses multi-head self-attention to reason over the representation and identify task-relevant regions for action generation.

considers them as disentangled visual concepts that inform the robot's decision-making. We are motivated by the recent advances in visual recognition, in particular, image models for generating object proposals[12, 13], *i.e.*, localized bounding boxes on 2D images. These object proposals capture general priors of "objectness" across appearance variations and object categories. They have served as intermediate representations for downstream vision tasks, such as object detection and instance segmentation [14–16]. Similarly, we use object proposals from pre-trained vision models as a starting point to build our own object-centric representations.

We introduce VIOLA (**V**isuomotor **I**mitation via **O**bject-centric **L**e**A**rning), an object-centric imitation learning algorithm to train closed-loop visuomotor policies for robot manipulation. The high-level overview of the method is illustrated in Figure 1. VIOLA first uses a pre-trained Region Proposal Network (RPN) [13] to get a set of general object proposals from raw visual observations. We extract features from each proposal region to build the factorized object-centric representations of the visual scene. These object-centric representations are converted into a set of discrete tokens and processed by a Transformer encoder [17], which is able to learn to focus on task-relevant regions while ignoring the irrelevant visual factors for decision-making through the multi-head self-attention mechanism when trained on supervised imitation learning objectives.

We compare VIOLA against state-of-the-art deep imitation learning methods for vision-based manipulation in simulation and on a real robot. We use simulation to systematically evaluate the policies' performances and generalization abilities in the canonical setting (*i.e.*, testing in the training distribution) and under three challenging variations, including object initial placements, presence of distracting objects, and camera pose perturbations. Our quantitative evaluations show that VIOLA outperforms the most competitive baseline by 45.8% in success rate. When visual variations such as jittered camera views are introduced, VIOLA maintains its robust behaviors of precise grasping and manipulation, while end-to-end learning methods would fail to reach the target objects. VIOLA also produces visuomotor policies to solve three challenging real-world tasks with a small set of 50 demonstrations, including a multi-stage coffee-making task where VIOLA achieves 60% success rate while baseline methods completely fail.

Our contributions with VIOLA are three-fold: 1) We learn object-centric representations based on general object proposals and design a transformer-based policy that determines task-relevant proposals to generate the robot's actions; 2) We show that VIOLA outperforms state-of-the-art baselines in simulation and validate the effectiveness of our model designs through ablative studies, and 3) We show that VIOLA learns policies on a real robot to complete challenging tasks.

2 Related Work

Imitation Learning (IL) for Manipulation. IL has been an established paradigm for acquiring manipulation policies for decades. It can be roughly categorized into non-parametric and parametric approaches. Non-parametric approaches, such as DMP and PrMP, can effectively acquire manipulation behaviors through a small number of demonstrations [18–21]. However, they typically focus on open-loop trajectory generation and fall short in handling high-dimensional observations. Parametric approaches, especially neural networks, have shown promise in vision-based manipulation. Nevertheless, these approaches are susceptible to distributional shifts and observation noises [1–

4, 22]. VIOLA builds upon recent work on end-to-end IL and introduces object-centric priors into the neural network policies, thus improving their robustness towards visual variations.

Visual Representations for Visuomotor Policies. Various types of intermediate visual representations have been explored for visuomotor policy learning. Object bounding boxes have been commonly used as intermediate visual representations [23–25]. However, they require fine-tuned or category-specific detectors and cannot easily generalize to tasks with previously unknown objects. Recently, deep learning methods enable IL to train policies end-to-end on raw observations [2, 3]. These methods are prone to covariate shift and causal confusion [5], resulting in poor generalization performances. Similar to our work, a large body of literature has looked into incorporating additional inductive biases into end-to-end policies. Notable ones include spatial attention [26–28] and affordances [29, 30]. However, these representations are purposefully designed for specific motion primitives, such as pick-and-place, limiting their abilities to generate diverse manipulation behaviors.

Object-Centric Representation. Object-centric representations have been widely used in visual understanding and robotics tasks, where researchers seek to reason about visual scenes in a modular way based on the objects presented. In robotics, poses [31–33] or bounding boxes [23–25] are commonly used as object-level abstractions. These representations often require prior knowledge about object instance/category and do not capture fine-grained details, falling short in applying to new tasks without previous unknown objects. Unsupervised object discovery methods [34, 35] learn object representation without manual supervision. However, they fall short in handling complex scenes [34, 35], hindering the applicability to realistic manipulation tasks. Recent work from the vision community has made great progress in generating object proposals for various downstream tasks, such as object detection [14–16] and visual-language reasoning [36, 37]. Motivated by the effectiveness of region proposal networks (RPNs) on out-of-distribution images [16], we use object proposals to scaffold our object-centric representations for robot manipulation tasks.

3 Approach

We introduce VIOLA, an object-centric imitation learning approach to vision-based manipulation. The core idea is to factorize raw visual observations into object-centric representations, on top of which the policy generates closed-loop control actions. Figure 2 illustrates the pipeline. We first formulate the problem of visuomotor policy learning and describe two key components of VIOLA: 1) how we build the object-centric representations based on general object proposals, and 2) how we use a transformer-based architecture to learn policy over the object-centric representations.

3.1 Problem Formulation

We formulate a robot manipulation task as a discrete-time Markov Decision Process, which is defined as a 5-tuple: $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, R, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P}(\cdot|s, a)$ is the stochastic transition probability, $R(s, a, s')$ is the reward function, and $\gamma \in [0, 1)$ is the discount factor. In our context, $\mathcal{S}$ is the space of the robot's raw sensory data, including RGB images and proprioception, $\mathcal{A}$ is the space of the robot's motor commands, and $\pi : \mathcal{S} \to \mathcal{A}$ is a closed-loop sensorimotor policy that we deploy on the robot to perform a task. Our goal is to learn the policy π from a set of human demonstrations that maximizes the expected return $\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1})]$.

We aim to design an object-centric representation that factorizes the visual observations of an unstructured scene into features of individual entities. For vision-based manipulation, we assume no access to the ground-truth states of object. Rather, we use the top K object proposals from a pre-trained Region Proposal Network (RPN) [13] to represent the set of object-related regions of an image. These proposals are grounded on image regions and optimized for covering the bounding boxes of potential objects. We treat these K proposals as the K (approximate) objects and extract visual and positional features to build *region features* from each proposal. For manipulation tasks, reasoning about object interactions is also important to decide actions, To provide contextual information for such relational reasoning, we design three *context features*: a global context feature from the workspace image to encode the current stage of the task; an eye-in-hand visual feature from the eye-in-hand camera to mitigate occlusion and partial observability of objects; and a proprioceptive feature from the robot's states. We call the set of region features and context features at time step t as the

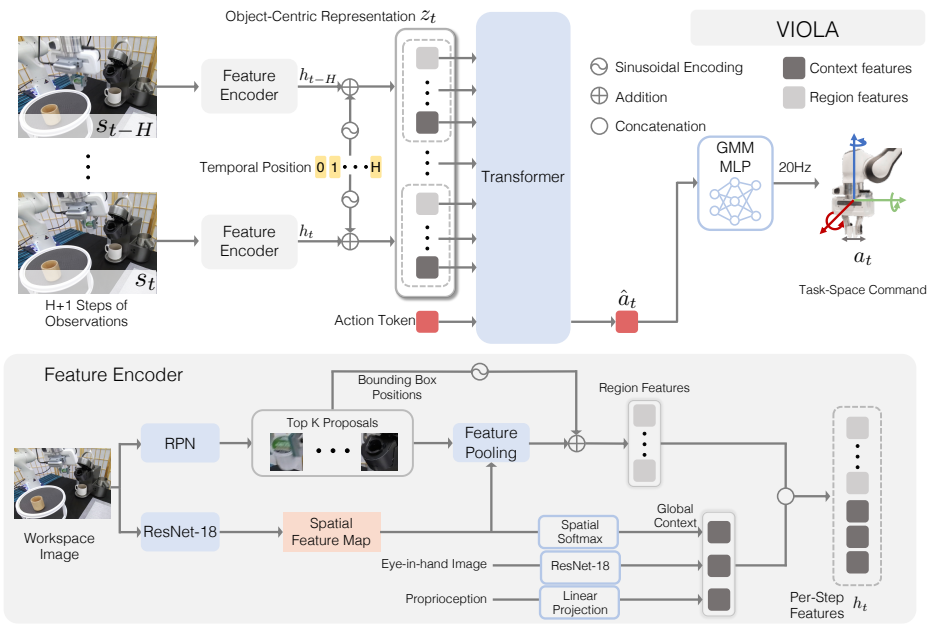


Figure 2: Model architecture of VIOLA. At time t, it computes the per-step features h_t using the top K object proposals. Then it constructs the object-centric representation z_t by composing per-step features from the last $H+1$ time steps with their temporal positional encodings. Transformer-based policy reasons over z_t to output a latent vector of the action token, $\hat{a}_t$, which is passed though a multi-layer perceptron (MLP) to generate actions.

per-step feature h_t. Finally, our object-centric representation z_t is a composition of per-step features from the last $H+1$ time steps with their temporal positional encoding, as we describe in Sec. 3.2.

3.2 Object-Centric Representation

This section describes how to build the object-centric representation z_t. VIOLA first obtains general object proposals using a pre-trained RPN. VIOLA computes region features from proposals and obtains a per-step feature h_t using the region features and the three context features which we describe next. VIOLA builds z_t through the temporal composition of h_t from a history of observations.

General Object Proposals. At each time step t, we generate object proposals using a pre-trained RPN on the workspace image. We select the top K proposals with the highest confidence scores, which indicate regions that contain objects with the highest likelihood. The intuition of using a pre-trained RPN is that it captures good priors of "objectness" in RGB images through supervision of natural image datasets. Recent work shows that an RPN trained on an 80-class dataset can generalize to a 2000-class dataset [16]. Our preliminary studies suggested that such a generalization ability also holds for localizing objects on raw images in our simulation and real-world tasks regardless of the domain gaps, and we use the pre-trained RPN from Zhou et al. [16] to localize regions with objects.

Region Features. For our policy to reason over objects and their spatial relations, we need to identify what objects each region contains and where these regions are from the top K proposals. To encode this information, we design a *visual feature* and a *positional feature* for each region. To extract the visual feature from a region, we learn a spatial feature map by encoding the workspace image with the ResNet-18 module [38] and extract the visual feature using ROI Align [12]. We use a learned spatial feature map for visual features, rather than from a pre-trained detection model because we share the same objective of localizing objects as object detectors, but different goals for the downstream tasks — pre-trained visual models are optimized for visual recognition tasks but we need actionable visual features that are informative for continuous control. For positional features, we encode the coordinates of bounding box corners on images using sinusoidal positional encoding [17] (more details in Appendix A). We flatten each visual feature and add it to the positional feature of the same region to obtain a region feature.

Context Features. We extract the region features for the policies to reason over individual objects. However, they are insufficient for decision-making in vision-based manipulation tasks, so we intro-

duce three context features to assist decision-making. As regions only encode local information, we add a *global context feature* to encode the current task stage from observation, which is computed from the spatial feature map of the workspace image using Spatial Softmax. During manipulation, the robot's gripper often occludes objects in the workspace view, so we add an *eye-in-hand feature* from eye-in-hand images to represent the occluded objects. We also encode the robot's measurements of its joints and the gripper into *proprioceptive features* for the policy to generate precise actions based on the robot's states. We aggregate these context features and region features at each time step t into a set, which we refer to as per-step feature h_t.

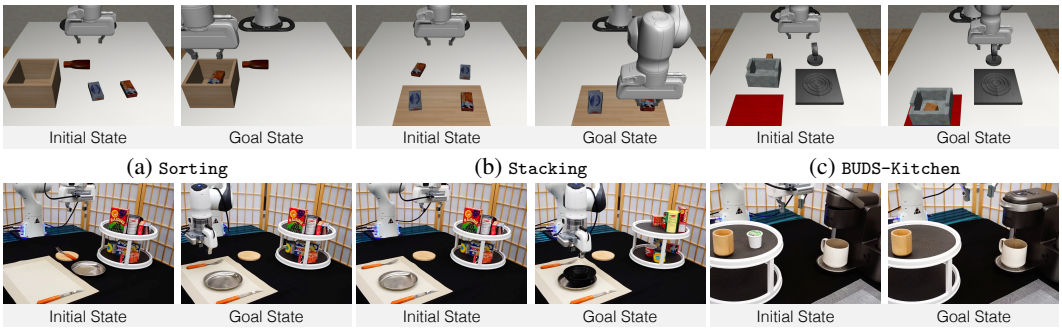
Temporal Composition. We build object-centric representation z_t through temporal composition of h_t that captures temporal dependencies and dynamic changes of object states. Building policies over a sequence of past observations rather than the most recent observation has been shown effective by prior work [1, 22]. In our work, temporal composition also increases the recall rate of object proposals, making the policy more robust in the case of detection failures. More specifically, z_t is built from a set of per-step features $\{h_{t-i}\}_{i=0}^{H}$ from the last $H+1$ time steps. To encode their temporal ordering, we add sinusoidal position encoding of the temporal positions $\{PE_i\}_{i=0}^{H}$ to the per-step features, resulting in our object-centric representations $z_t = \{h_{t-i} \oplus PE_i\}_{i=0}^{H}$. Our ablative studies (see Sec. 4) indicate the importance of temporal positional encoding that retains with the temporal ordering of features, especially when using our transformer-based policy.

3.3 Transformer-based Policy

We desire a policy that focuses on task-relevant region features in z_t to generate actions. Regions that associate with task-relevant objects facilitate the accurate prediction of actions while regions with task-irrelevant objects are likely to confound the policies. We seek to use a transformer [17] as the policy backbone, which allows policies to reason over objects and their relations with its self-attention mechanism. The core of a transformer is an encoder layer which consists of a multi-head self-attention block (MHSA), a layer-normalization [39] function, and a fully-connected neural network (FFN). A transformer encoder layer takes as input a sequence of n latent vectors $(x_1, \ldots, x_n)$ (also called *tokens*) and outputs a sequence of n latent vectors $(y_1, \ldots, y_n)$. A multi-head self-attention (MHSA) block consists of multiple self-attention blocks in parallel, which computes attention over all the tokens and compute a weighted sum of token values. We refer readers to Vaswani et al. for more details [17]. In short, each transformer encoder outputs $Y = \text{FFN}(\text{LayerNorm}(\text{MSHA}(X)))$, where each row of Y is an output latent vectors y_i that corresponds to x_i. Our transformer-based policy is a stack of multiple transformer encoder layers, which allows for a higher degree of expressiveness over input tokens compared to a single layer.

For our policy, we tokenize our object-centric representation z_t, treating each region and context feature vector as an input token. To make action generation attend more to task-relevant region features than task-irrelevant ones, we append a learnable token, action token, to the input sequence of transformer. The action token design resembles the specific class tokens in natural language understanding tasks [40] or visual recognition tasks [41], which are used for outputting latent vectors for downstream tasks. Similarly, we can get the output latent vector $\hat{a}_t$ of the action token, which learn to attend to task-relevant regions through action supervision. In the end, we pass $\hat{a}_t$ through a two-layered fully-connected network, followed by a GMM (Gaussian Mixture Model) output head, which has been shown effective to capture the diverse multimodal behaviors in demonstration data [42, 22].

Implementation Details. Here we describe several key implementation details. Full implementation details are in Appendix A. We set $K = 20$ in simulation which offers a nice trade-off between object recall rates and computation efficiency (see Appendix B.1). In the real world, we set $K = 15$ which are found to attain a high recall rate on real images. For temporal composition, we use $H = 9$ to be consistent with the setup in BC-RNN [22]. The action token is a learnable parameter randomly initialized from normal distribution. We use data augmentation durining training to encourage the generalization ability of policies. We adopt random erasing [43] as well as standard pixel shifting [4, 22, 41] to prevent policies from overfitting to specific region features.



(a) `Sorting` (b) `Stacking` (c) `BUDS-Kitchen`

(d) `Dining-PlateFork` (e) `Dining-Bowl` (f) `Make-Coffee`

Figure 3: Visualization of initial and goal configurations for the simulation tasks and real-world tasks.

Tasks	Variants	BC [46]	OREO [5]	BC-RNN [22]	VIOLA-Patch	VIOLA
`Sorting`	`Canonical`	25.1 ± 1.6	38.7 ± 0.5	62.8 ± 0.9	71.2 ± 1.0	**87.6** ± 1.1
	`Placement`	1.9 ± 0.7	8.3±1.7	11.7 ± 1.0	48.5 ± 2.2	**68.3** ± 1.5
	`Distractor`	14.5 ± 1.9	26.0± 11.4	46.7 ± 6.5	58.6 ± 4.2	**74.4** ± 5.7
	`Camera-Jitter`	6.1 ± 0.3	16.3± 3.3	9.6 ± 0.4	34.6 ± 1.4	**50.7** ± 0.6
`Stacking`	`Canonical`	14.9 ± 1.1	13.3 ± 2.0	27.8 ± 0.8	71.2 ± 1.0	**71.3** ± 1.0
	`Placement`	1.6 ± 0.3	0.3± 0.5	0.0 ± 0.0	28.4 ± 1.9	**46.7** ± 0.2
	`Distractor`	5.6 ± 1.4	5.6 ± 4.5	14.4 ± 3.2	**41.4** ± 5.1	38.6 ± 2.8
	`Camera-Jitter`	2.1 ± 0.3	0.6 ± 0.9	1.0 ± 0.0	17.6 ± 1.0	**29.3** ± 1.7
`BUDS-Kitchen`	`Canonical`	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	5.8 ± 0.6	**84.2** ± 1.3
	`Placement`	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	3.1 ± 0.6	**58.4** ± 1.1
	`Distractor`	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	10.7 ± 0.7	**73.2** ± 6.2
	`Camera-Jitter`	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	2.6 ± 0.3	**41.2** ± 1.7

Table 1: Success rates (%) in simulation setups over 100 initializations with repeated runs of 3 random seeds.

4 Experiments

We design our experiments to answer the following questions: 1) How well does VIOLA perform against state-of-the-art end-to-end imitation learning algorithms? 2) How does it take advantage of object-centric representations? 3) What design choices are essential for good performance? and 4) Is VIOLA practical for real-world deployment?

4.1 Experimental Setup

We conduct quantitative evaluations in simulation and real-world tasks to validate our approach. The simulation tasks are designed with the robosuite [44] framework and used for quantitative comparisons between VIOLA and baselines. We also validate our design choices through ablative studies. We design three simulation tasks, `Sorting`, `Stacking`, `BUDS-Kitchen`, and three real-world tasks, `Dining-PlateFork`, `Dining-Bowl`, `Make-Coffee` that cover a rich spectrum of manipulation behaviors that combine prehensile and non-prehensile motions. We visualize their initial and goal configurations in Figure 3. We design these tasks to understand if the use of object priors benefits policy learning along two axes of task characteristics: large placement variations of objects and multi-stage long-horizon execution. For the first axis, we design `Sorting` and `Stacking` to have large initial ranges of object placements. For the second axis, we use `BUDS-Kitchen`, a multi-stage long-horizon manipulation task from prior work [4]. The real-world tasks are designed to resemble real-world everyday tasks: `Dining-PlateFork` and `Dining-Bowl` for dining table arrangements, `Make-Coffee` for espresso coffee making. Full details of tasks and data collection are included in Sec. C.

Evaluation setup To systematically evaluate the efficacy and robustness of the learned policies, we design the following testing setups in simulation: 1) `Canonical`: all the objects and sensor configurations follow the same distribution as seen in demonstrations; 2) Three testing variants, namely `Placement`, `Distractor`, and `Camera-Jitter`. The design of these variants is inspired by the MAGICAL benchmark [45] for evaluating systematic generalization of imitation learning policies. The design principle of testing variants is to keep the task semantics the same as in the canonical setup, and we identify the three challenging axes of variations for learning-based manipulation, namely object initial placements (`Placement`), distractor objects presented in the scene (`Distractor`), and camera pose jitters (`Camera-Jitter`).

Models	Canonical	Placement	Distractor	Camera-Jitter
Base	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
+ Temporal Observation	$\uparrow 69.5 \pm 2.5$	$\uparrow 25.1 \pm \uparrow 2.2$	$\uparrow 37.7 \pm 3.4$	$\uparrow 39.9 \pm 1.5$
+ Temporal Positional Encoding	$\uparrow 74.0 \pm 0.8$	$\uparrow 48.3 \pm 1.9$	$\uparrow 48.3 \pm 3.3$	$\uparrow 50.9 \pm 1.5$
+ Region Visual Features	72.8 ± 0.9	$\downarrow 37.7 \pm 1.2$	$\uparrow 54.6 \pm 4.6$	49.2 ± 1.6
+ Region Positional Features	$\uparrow 80.2 \pm 2.9$	38.6 ± 0.3	$\uparrow 62.0 \pm 5.7$	$\downarrow 46.5 \pm 1.9$
+ Random Erasing (=VIOLA)	$\uparrow 87.6 \pm 1.1$	$\uparrow 68.3 \pm 1.5$	$\uparrow 74.4 \pm 5.7$	$\uparrow 50.7 \pm 0.6$

Table 2: The effect of VIOLA model designs on success rate (%) in `Sorting` task. Changes of success rates are annotated with ↑ / ↓ if larger than 2%.

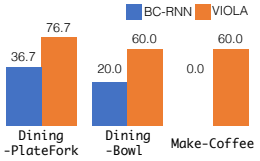


Figure 4: Success rates (%) in real robot tasks.

4.2 Experiment Results

We answer question (1) by quantitatively comparing VIOLA with the following baselines: **BC**: behavioral cloning that conditions on current observations; **OREO** [5]: behavioral cloning method that learns object-aware discrete codes using VQ-VAE [47] and learns policies thereon; **BC-RNN** [22]: a state-of-the-art method that uses a temporal sequence of past observations with recurrent neural networks. **VIOLA-Patch**: a variant of VIOLA, where we use a regular grid of image patches as inputs to the policy rather than proposals, similar to ViT [48]. To make fair evaluation, we randomly generate 100 initial configurations and repeat runs with three different random seeds. We evaluate all policies on the same set of pre-generated initial configurations.

The result in Table 1 shows that VIOLA outperforms the most competitive baseline BC-RNN by 50.8% success rate in `Canonical` and 44.1% in three testing variants. VIOLA's strong performance suggests the advantage of object-centric representations for visuomotor imitation. OREO's results suggest that learning object-aware discrete codes via unsupervised learning does not consistently improve performance over simple BC for all tasks. The comparisons between BC, OREO, and the other methods using temporal windows show the importance of temporal modeling for these algorithms to achieve the high performances and robustness in complex vision-based manipulation domains.

Table 1 shows that VIOLA-Patch has comparable performance to VIOLA in some evaluation setups. It suggests that the transformer backbone exhibits capability to attend to patch regions that cover task-relevant visual cues, even though the patch division is agnostic to objects. Nonetheless, VIOLA still performs better than VIOLA-Patch, especially in the long-horizon task `BUDS-Kitchen`, suggesting regions with complete coverage of objects is critical for the success of VIOLA, highlighting the importance of the object priors.

Ablative Studies To answer question (2) and (3), we use ablative studies to validate our model's design and show how it takes advantage of the object-centric priors. The results in Table 2 quantify the effects of the transformer backbone, the use of temporal windows, the use of object proposal regions, and our data augmentation technique. We start with our base model, a transformer model that only takes the current observations as input. Second, we add the temporal window of observations similar to BC-RNN. Results show that the use of temporal observation is key to unleashing the power of the transformer architecture, making the performance of this ablated version comparable to BC-RNN. Nonetheless, this model does not encode temporal ordering as the transformer model is invariant to input permutation. In the next row, we add temporal positional encoding [17] to the input sequence, which produces a method that outperforms the top baselines.

In the next two rows, we procedurally add in visual and positional features of regions to prove that our model does exploit the object priors. Results show that visual features alone only improve our model's robustness to `Distractor`. When using both visual and positional features, the model performs better in `Canonical` and `Distractor`, but worse in `Placement` and `Camera-Jitter`. We hypothesize that this ablated version overfits to locations of proposal boxes in demonstrations, therefore it generalizes worse in `Placement` and `Camera-Jitter` where the distribution of object locations on 2D images shift from demonstrations. To mitigate such overfitting, we use Random Erasing [43] to achieve highest success rates in all evaluation setup.

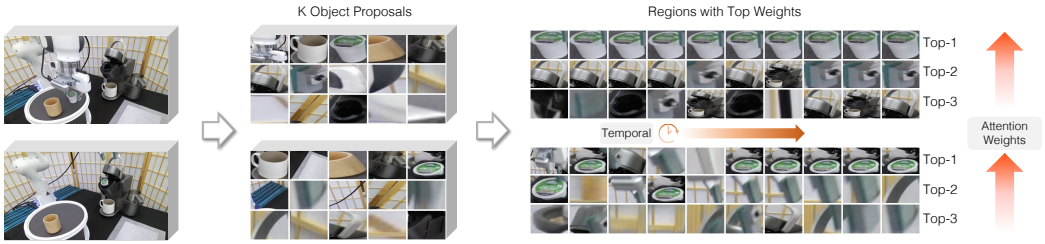


Figure 5: Visualization of transformer attention over regions. When the robot is to grasp k-cup, VIOLA's top attention is over k-cup across observations while it takes robot fingers and coffee machine into account to help spatial reasoning. When the robot is to close the coffee machine, VIOLA's top-1 attention is all over k-cups, robot gripper, and the coffee machine, and it reasons over these regions to generate actions.

Real Robot Evaluation We compare VIOLA against the SOTA baseline, BC-RNN, on all three real-world tasks. We evaluate in 10 different initial configurations and repeat 3 times in the same configuration for each policy evaluation. To mitigate potential human bias introduced by setting up objects before each trial, we employ the A/B testing paradigm for evaluation, that is, we reset the environment while being agnostic to the next policy to execute, and the whole evaluation process for one initialization is repeated until all evaluation trials are completed.

The quantitative evaluation in Figure 4 shows that VIOLA outperforms BC-RNN by 46.7% success rate on average. Qualitatively, we observe that VIOLA can robustly grasp k-cups or open the coffee machine in the `Make-Coffee` task, while BC-RNN tends to reach to wrong positions where grasps are missed, or to release the gripper mistakenly. In the supplementary videos, we include the evaluation rollouts of VIOLA for all the tasks. We also qualitatively visualize the attention of VIOLA in Figure 5 by showing the top 3 regions with highest attention weights at each timestep of temporal observation. The figure shows that when the robot is to grasp k-cup, VIOLA's top attention is over the k-cup while it also takes robot fingers and coffee machine into account to facilitate spatial reasoning. When the robot is to close the coffee machine, VIOLA's top-1 attention spreads over the k-cup, robot gripper, and the coffee machine, and the policy generates actions for closing through spatial reasoning over these regions.

5 Conclusion

We present VIOLA, an object-centric imitation learning approach to learning closed-loop visuomotor policies for robot manipulation. Our approach uses general object proposals to build object-centric representation. This representation encodes the visual and positional features from the proposal regions and context features of global scene information and robot states. We design a transformer-based policy to identify task-specific relevant regions for action generation. The results show the superior performances of VIOLA compared to state-of-the-art baselines in simulation and the real world. We also validate our model designs through ablative studies, showing how each component of the model impacts the policy performance.

Limitations While VIOLA has achieved great success in learning robust visuomotor policies, it suffers from common limitations of learning from offline demonstration datasets. One future direction is to use this object-centric representation in an online manner so that it can improve over roll-out experience over time. Also, we use a pre-trained RPN without adaptation, it may fail to generalize to manipulation scenes with aggressive distributional shifts from natural images on which the RPN is trained. A possible improvement is to fine-tune the RPN while training the policy. Another limitation of our approach is that we only consider monocular RGB images in the input data, which lacks the 3D information of objects. This formulation prevents VIOLA from excluding the background visual elements in the input, which would limit the generalization ability of VIOLA in variants of large visual changes, such as table texture change. For future work, we would like to extend VIOLA to using depth images so that distracting background elements such as tables can be dientangled from the representation.

References

[1] A. Mandlekar, F. Ramos, B. Boots, S. Savarese, L. Fei-Fei, A. Garg, and D. Fox. Iris: Implicit reinforcement without interaction at scale for learning control from offline robot manipulation data. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4414–4420. IEEE, 2020.

[2] A. Mandlekar, D. Xu, R. Martín-Martín, S. Savarese, and L. Fei-Fei. Learning to generalize across long-horizon tasks from human demonstrations. *arXiv preprint arXiv:2003.06085*, 2020.

[3] T. Zhang, Z. McCarthy, O. Jow, D. Lee, X. Chen, K. Goldberg, and P. Abbeel. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5628–5635. IEEE, 2018.

[4] Y. Zhu, P. Stone, and Y. Zhu. Bottom-up skill discovery from unsegmented demonstrations for long-horizon robot manipulation. *IEEE Robotics and Automation Letters*, 2022.

[5] J. Park, Y. Seo, C. Liu, L. Zhao, T. Qin, J. Shin, and T.-Y. Liu. Object-aware regularization for addressing causal confusion in imitation learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[6] P. De Haan, D. Jayaraman, and S. Levine. Causal confusion in imitation learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[7] C. Wen, J. Lin, T. Darrell, D. Jayaraman, and Y. Gao. Fighting copycat agents in behavioral cloning from observation histories. *Advances in Neural Information Processing Systems*, 33: 2564–2575, 2020.

[8] F. Codevilla, E. Santana, A. M. López, and A. Gaidon. Exploring the limitations of behavior cloning for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9329–9338, 2019.

[9] K. Greff, S. Van Steenkiste, and J. Schmidhuber. On the binding problem in artificial neural networks. *arXiv preprint arXiv:2012.05208*, 2020.

[10] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman. Building machines that learn and think like people. *Behavioral and brain sciences*, 40, 2017.

[11] E. S. Spelke and K. D. Kinzler. Core knowledge. *Developmental science*, 10(1):89–96, 2007.

[12] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[13] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[14] Z. Cai and N. Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018.

[15] X. Zhou, D. Wang, and P. Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.

[16] X. Zhou, R. Girdhar, A. Joulin, P. Krähenbühl, and I. Misra. Detecting twenty-thousand classes using image-level supervision. *arXiv preprint arXiv:2201.02605*, 2022.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[18] S. Schaal. Dynamic movement primitives-a framework for motor control in humans and humanoid robotics. In *Adaptive motion of animals and machines*, pages 261–280. Springer, 2006.

[19] J. Kober and J. Peters. Learning motor primitives for robotics. In *2009 IEEE International Conference on Robotics and Automation*, pages 2112–2118. IEEE, 2009.

[20] A. Paraschos, C. Daniel, J. R. Peters, and G. Neumann. Probabilistic movement primitives. *Advances in neural information processing systems*, 26, 2013.

[21] A. Paraschos, C. Daniel, J. Peters, and G. Neumann. Using probabilistic movement primitives in robotics. *Autonomous Robots*, 42(3):529–551, 2018.

[22] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. *arXiv preprint arXiv:2108.03298*, 2021.

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

[24] M. Sieb, Z. Xian, A. Huang, O. Kroemer, and K. Fragkiadaki. Graph-structured visual imitation. In *Conference on Robot Learning*, pages 979–989. PMLR, 2020.

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

[26] A. Zeng, P. Florence, J. Tompson, S. Welker, J. Chien, M. Attarian, T. Armstrong, I. Krasin, D. Duong, V. Sindhwani, et al. Transporter networks: Rearranging the visual world for robotic manipulation. *arXiv preprint arXiv:2010.14406*, 2020.

[27] D. Seita, P. Florence, J. Tompson, E. Coumans, V. Sindhwani, K. Goldberg, and A. Zeng. Learning to rearrange deformable cables, fabrics, and bags with goal-conditioned transporter networks. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4568–4575. IEEE, 2021.

[28] M. Shridhar, L. Manuelli, and D. Fox. Cliport: What and where pathways for robotic manipulation. In *Conference on Robot Learning*, pages 894–906. PMLR, 2022.

[29] M. Breyer, J. J. Chung, L. Ott, R. Siegwart, and J. Nieto. Volumetric grasping network: Real-time 6 dof grasp detection in clutter. *arXiv preprint arXiv:2101.01132*, 2021.

[30] Z. Jiang, Y. Zhu, M. Svetlik, K. Fang, and Y. Zhu. Synergies between affordance and geometry: 6-dof grasp detection via implicit representations. *arXiv preprint arXiv:2104.01542*, 2021.

[31] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield. Deep object pose estimation for semantic robotic grasping of household objects. *arXiv preprint arXiv:1809.10790*, 2018.

[32] S. Tyree, J. Tremblay, T. To, J. Cheng, T. Mosier, J. Smith, and S. Birchfield. 6-dof pose estimation of household objects for robotic manipulation: An accessible dataset and benchmark. *arXiv preprint arXiv:2203.05701*, 2022.

[33] T. Migimatsu and J. Bohg. Object-centric task and motion planning in dynamic environments. *IEEE Robotics and Automation Letters*, 5(2):844–851, 2020.

[34] C. P. Burgess, L. Matthey, N. Watters, R. Kabra, I. Higgins, M. Botvinick, and A. Lerchner. Monet: Unsupervised scene decomposition and representation. *arXiv preprint arXiv:1901.11390*, 2019.

[35] F. Locatello, D. Weissenborn, T. Unterthiner, A. Mahendran, G. Heigold, J. Uszkoreit, A. Dosovitskiy, and T. Kipf. Object-centric learning with slot attention. *Advances in Neural Information Processing Systems*, 33:11525–11538, 2020.

[36] W. Su, X. Zhu, Y. Cao, B. Li, L. Lu, F. Wei, and J. Dai. Vl-bert: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*, 2019.

[37] Y.-C. Chen, L. Li, L. Yu, A. El Kholy, F. Ahmed, Z. Gan, Y. Cheng, and J. Liu. Uniter: Universal image-text representation learning. In *European conference on computer vision*, pages 104–120. Springer, 2020.

[38] 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*, pages 770–778, 2016.

[39] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[40] 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.

[41] I. Kostrikov, D. Yarats, and R. Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *arXiv preprint arXiv:2004.13649*, 2020.

[42] Z. Wang, A. Novikov, K. Zolna, J. S. Merel, J. T. Springenberg, S. E. Reed, B. Shahriari, N. Siegel, C. Gulcehre, N. Heess, et al. Critic regularized regression. *Advances in Neural Information Processing Systems*, 33:7768–7778, 2020.

[43] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13001–13008, 2020.

[44] Y. Zhu, J. Wong, A. Mandlekar, and R. Martín-Martín. robosuite: A modular simulation framework and benchmark for robot learning. *arXiv preprint arXiv:2009.12293*, 2020.

[45] S. Toyer, R. Shah, A. Critch, and S. Russell. The magical benchmark for robust imitation. *Advances in Neural Information Processing Systems*, 33:18284–18295, 2020.

[46] 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.

[47] A. Van Den Oord, O. Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[48] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[49] O. Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 3(1):43–53, 1987.

[50] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li. Bag of tricks for image classification with convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 558–567, 2019.

[51] R. Jangir, N. Hansen, S. Ghosal, M. Jain, and X. Wang. Look closer: Bridging egocentric and third-person views with transformers for robotic manipulation. *IEEE Robotics and Automation Letters*, 2022.