SORNet: Spatial Object-Centric Representations for Sequential Manipulation

Wentao Yuan, Chris Paxton, Karthik Desingh, and Dieter Fox

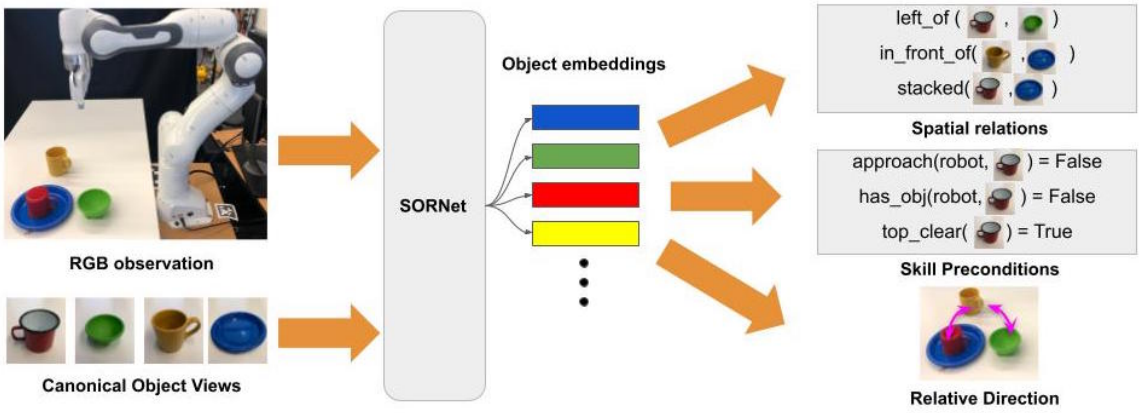

Fig. 1: We propose SORNet (Spatial Object-Centric Representation Network) that learns object embeddings useful for spatial reasoning tasks such as predicting spatial relations, classifying skill preconditions, and regressing relative direction between objects. SORNet embeddings can be used for a variety of tasks, including classifying skill preconditions and regressing object directions, and can be learned for a wide variety of objects.

I. Introduction

To complete multi-step robot tasks, robots must be able to understand qualities of objects and the relationship between them. Take, for example, deconstructing and rebuilding a tower: the robot has to be able to tell whether each block is accessible, determine how to make it so if need be, and then must understand the consequences of actions as each block is placed and moved.

In this sort of task, it is common to use a state estimator followed by a task and motion planner or other model-based system [1], [2]. A variety of powerful approaches exist for explicitly estimating object poses, e.g. [3]. However, it is challenging to generalize these approaches to an arbitrary collection of objects, and manipulation scenarios often include contact and occlusion in which model-free methods tend to fail [4], [5]. Fortunately, knowing exact poses of objects is not necessary for manipulation.

In this work, we propose a neural network that extracts object-centric embeddings from raw RGB images conditioned on object queries, which we call **SORNet**, or **S**patial **O**bject-Centric **R**epresentation **Net**work. The design of SORNet allows it to generalize to novel objects without retraining or finetuning. The object-centric embeddings produced by **SORNet** can be combined with readout networks to inform a task and motion planner with implicit object states relevant to goal-directed sequential manipulation tasks, e.g. logical preconditions for primitive skills or continuous 3D directions from the end effector to objects in the scene.

To summarize, our contribution are: (1) a method for extracting object-centric embeddings from RGB images that generalizes zero-shot to different number and type of objects; (2) a framework for learning object embeddings that capture continuous spatial relations with only logical supervision; (3) a dataset containing sequences of RGB observations labeled with spatial predicates during various tabletop rearrangement manipulation tasks.

Our approach allows for zero-shot generalization to new objects and goals. We evaluate the object-centric embeddings produced by **SORNet** on (1) classification of predicate preconditions; and (2) prediction of relative 3D direction between entities. Models were tested on held-out objects and colors that did not appear in training data. In both tasks, **SORNet** obtains significant improvements over the baseline methods. In this short paper we present the extensions of the **SORNet** on objects beyond blocks along with comparisons to additional baselines; for more results see the full paper [6].

II. Methods

Our object embedding network (**SORNet**) (Fig. 2) takes an RGB image and an arbitrary number of canonical object views and outputs an embedding vector corresponding to each input object patch. The architecture of the network is based on the Visual Transformer (ViT) [7]. The input image is broken into a list of fixed-sized patches, which we call *context patches*. The context patches are concatenated with the canonical object views to form a patch sequence. Each patch is first flattened and then linearly projected into a token vector, then positional embedding is added to the sequence of tokens. Following [7], we use a set of learnable vectors with the same dimension as the token vectors as positional embeddings. The positional-embedded tokens are then passed through a transformer encoder, which includes multiple layers of multi-head self-attention. The transformer encoder outputs a sequence of embedding vectors. We discard the embedding for context patches and keep those for the canonical object views.

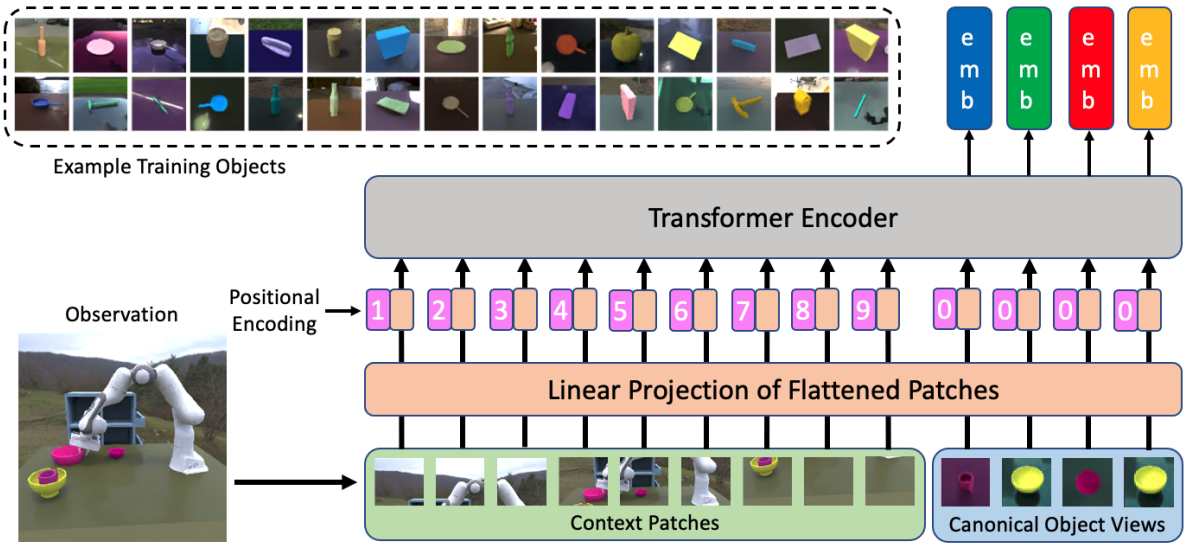

Fig. 2: **SORNet** architecture. Input to the network is an RGB image and canonical views of the objects of interest. The RGB image is broken into context patches which have the same size as the canonical views. These patches are flattened and added with positional encoding and passed through a multi-layer multi-head transformer [7]. The embeddings corresponding to the canonical views are used for the downstream tasks. The top left inset shows examples of canonical object views used during training.

We apply the same positional embedding to the canonical object views to make the output embeddings permutation equivariant. We also mask out the attention among canonical object views and the attention from context patches to canonical object views to ensure the model uses information from the context patches to make predictions. In this way, we can pass in an arbitrary number of canonical object views in arbitrary order without changing model parameters during inference.

Intuitively, the canonical object views can be viewed as queries where the context patches serve as keys to extract the spatial relations' values. Note that the canonical object views are *not* crops from the input image, but arbitrary views of the objects that may not match the objects' appearance in the scene. Our model learns to identify objects even under drastic change in lighting, pose and occlusion. Fig. 2 shows some examples of canonical object views used in our experiments.

In the experiments, we also test on multiple-views. In these experiments, context patches from different views are concatenated to form a single patch sequence.

A. Readout Networks

The readout networks (Fig. 3) are responsible for predicting a list of relations using object embeddings. The relations can be logical statements, e.g., whether the blue block is stacked onto the green block, or continuous quantities, e.g. which direction should the end effector move to reach the red block. The readout networks consist of a collection of 2-layer MLPs, one for each type of relations. Here we focus on unary and binary relations. Unary relations involve a single object or an object and the environment, which could be the robot or a region on the table. Binary relations involve two objects and, optionally, the environment. In principle, our framework is extensible to relations involving more than two objects, but we leave that for future work.

The readout network for unary relations takes the list of object embeddings and outputs relations pertaining to the object that the embedding is conditioned on. Taking the `top_is_clear` classifier for an example, if the input embedding is conditioned on the blue block, the network will output whether there is any object on top of the blue block. If the input embedding is conditioned on the red block, the network will output whether there is any object on top of the red block.

The readout network for binary relations takes a list of binary object embeddings created by concatenating pairs of object embeddings and outputs relations corresponding to a pair of objects, e.g., whether the blue block is on top of the red block. Thus, with N object embeddings, there will be $N(N-1)$ binary object embeddings and $N(N-1)$ output relations.

Parameters of the readout network are independent of the number of objects. The number of output relations dynamically changes with the number of input object embeddings. For example, when are 7 unary relations and 2 binary relations, with 4 objects, the network generates $7 \times 4 + 2 \times 4 \times (4-1) = 52$ outputs; with 5 objects, the network generates $7 \times 5 + 2 \times 5 \times (5-1) = 75$ outputs. In this way, our overall model generalizes zero-shot to scenes with an arbitrary number of objects.

III. Data Generation

We created a simulated tabletop environment where a Franka Panda robot manipulates a set of randomly colored ShapeNet objects [8], specifically from the ACRONYM subset [9]. We also tested on the "Leonardo" blocks dataset described in our full paper [6]. We sampled high-level actions and used a simple task and motion planner [2] to generate trajectories. As we know the ground truth poses of the objects in the simulator, we computed ground-truth logical

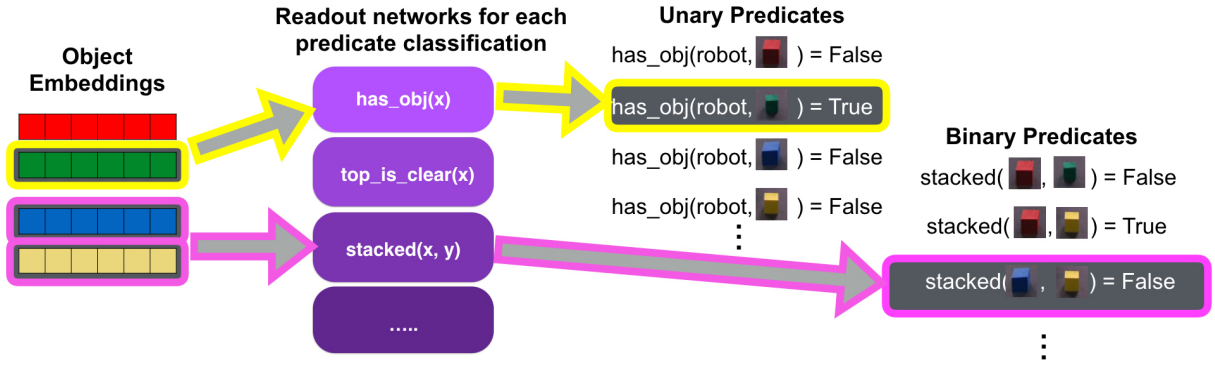

Fig. 3: Architecture of the readout networks, which uses the object embeddings from **SORNet** to predict spatial relations, such as logical statements that can serve as skill preconditions or continuous 3D directions. The readout network is flexible to accommodate any number of input object embeddings without changing its parameters.

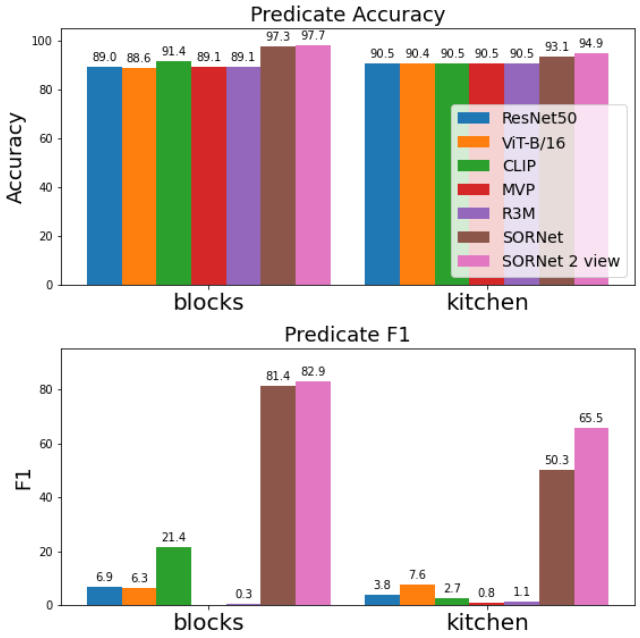

Fig. 4: Predicate classification results on blocks and kitchen data. SORNet clearly outperforms other pre-training methods on reasoning about spatial relations in scenarios with complex object interactions.

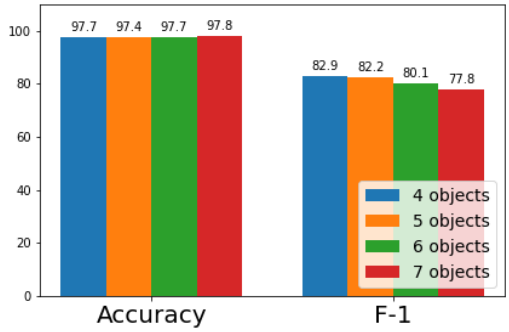

Fig. 5: SORNet generalizes to scenes with different number of objects with almost no performance drop.

predicates at every step of the planning process. We used NViSII [10] to render the RGB and depth images used in training. Domain randomization including random lighting, background and perturbations to the camera position was applied while rendering.

During training, we used the XKCD colors[1], but held out every color containing the words (red, green, blue, yellow). We also held out the mug and bowl classes to only appear in test data. For 2-view experiments, we fixed virtual cameras at 2 different locations as shown in Fig 6.

IV. Results

Predicate classification. We benchmark our model against 3 state-of-the-art pretraining methods, CLIP [11] MVP [12] and R3M [13] on the task of predicting spatial relations among objects in the form of logical predicates. The models are trained/finetuned on our large-scale manipulation dataset to predict 6 types of relations (on surface, has obj, in approach region, top is clear, stacked and aligned with) and tested on images where the objects are completely unseen during training. For our model, we only need to provide a single canonical patch per test object without any predicate labels. For the baselines, we provide 100 sequences with ground truth predicates for the test objects because these models do not work zero-shot on unseen objects.

We report the average accuracy and F-1 score on both the blocks and the kitchen data in Fig. 4. For reference, we also show results for ResNet50 (backbone used by R3M) and ViT-B/16 (backbone used by CLIP and MVP) trained from scratch. We can see that even after fine-tuning on scenes with test objects, the baselines significantly underperform zero-shot SORNet. This demonstrates the generalizability of our model obtained via conditioning on canonical object views.

Further, we tested our model, which was only trained on 4-object scenes, to scenes with 5 to 7 objects. We can see from Fig. 5 that SORNet generalizes to scenes with different number of objects with almost no performance drop.

Open-loop planning. Please refer to our supplementary video for the demo, where we incorporate SORNet as a part of an open-loop planning pipeline in a real-world manipulation scenario.

[1] https://xkcd.com/color/rgb/

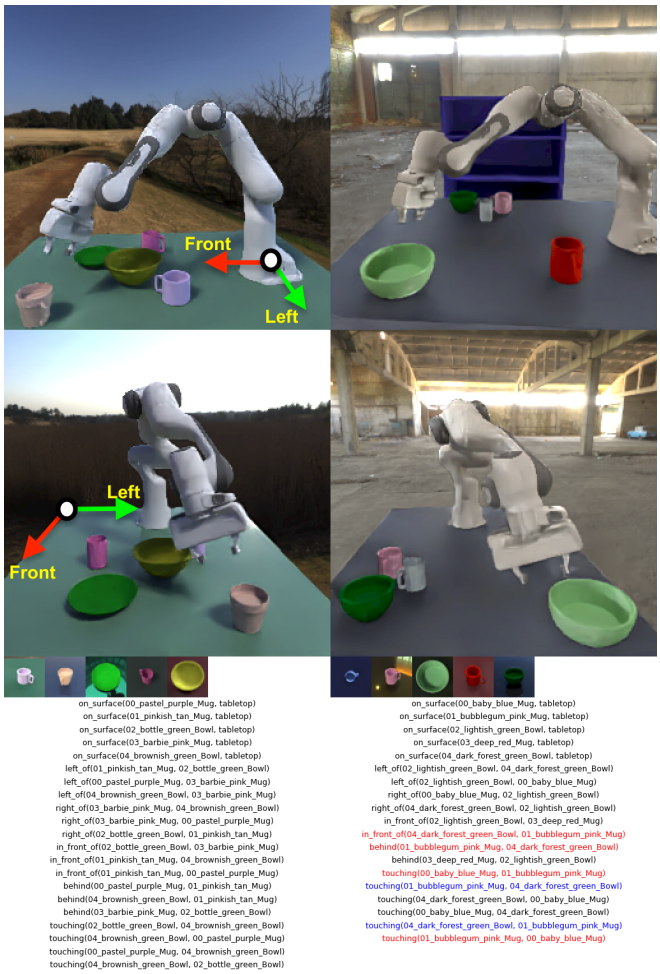

Fig. 6: Qualitative predicate classification results on the kitchen data. Black means true positive, blue means false positive and red means false negative. True negatives are not shown due to limited space. Reference directions (front and left) are shown in the left scenario.

Direction prediction. Finally, we performed an experiment to show that SORNet embeddings capture continuous spatial information. Specifically, we trained a regressor on top of frozen SORNet embeddings to predict the continuous direction between two objects (Obj-Obj) or the direction the end effector should move to reach a certain object (EE-Obj), using an L2 loss. The results are shown in Fig 7. This demonstrate SORNet's representation transfers much better than baseline methods to predict precise spatial quantities, which is crucial in robot manipulation tasks.

V. Conclusion

We proposed **SORNet** (**S**patial **O**bject-Centric **R**epresentation **Net**work) that learns object-centric representations from RGB images. We show that the object embeddings produced by **SORNet** capture spatial relations which can be used in a downstream tasks such as spatial relation classification, skill precondition classification and relative direction regression. Our method works on scenes with an arbitrary number of unseen objects in a zero-shot fashion. With real-world robot experiments, we demonstrate how **SORNet** can be used in manipulation of novel objects.

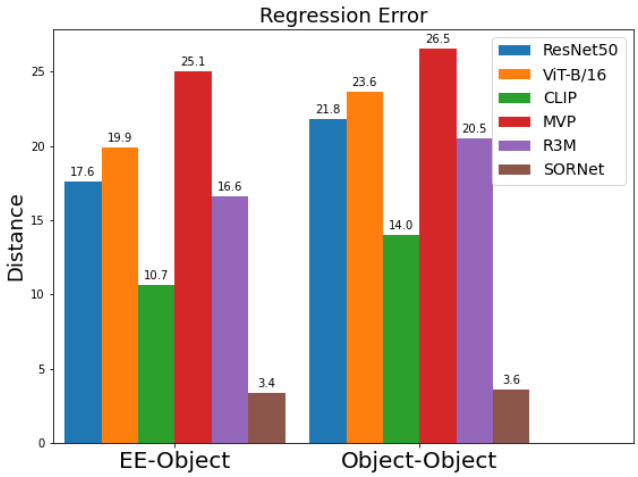

Fig. 7: Euclidean error on regression of continuous 3D unit vector between entities in the scene.

References

[1] M. Fox and D. Long, "Pddl2. 1: An extension to pddl for expressing temporal planning domains," *Journal of artificial intelligence research*, vol. 20, pp. 61–124, 2003.

[2] C. Paxton, N. Ratliff, C. Eppner, and D. Fox, "Representing robot task plans as robust logical-dynamical systems," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 5588–5595.

[3] X. Deng, A. Mousavian, Y. Xiang, F. Xia, T. Bretl, and D. Fox, "Poserbpf: A rao–blackwellized particle filter for 6-d object pose tracking," *IEEE Transactions on Robotics*, 2021.

[4] C. Xie, Y. Xiang, A. Mousavian, and D. Fox, "The best of both modes: Separately leveraging rgb and depth for unseen object instance segmentation," in *Conference on robot learning*. PMLR, 2020, pp. 1369–1378.

[5] Y. Xiang, C. Xie, A. Mousavian, and D. Fox, "Learning rgb-d feature embeddings for unseen object instance segmentation," *arXiv preprint arXiv:2007.15157*, 2020.

[6] W. Yuan, C. Paxton, K. Desingh, and D. Fox, "Sornet: Spatial object-centric representations for sequential manipulation," in *Conference on Robot Learning*. PMLR, 2022, pp. 148–157.

[7] 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.

[8] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[9] C. Eppner, A. Mousavian, and D. Fox, "Acronym: A large-scale grasp dataset based on simulation," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 6222–6227.

[10] N. Morrical, J. Tremblay, S. Birchfield, and I. Wald, "ViSII: Virtual scene imaging interface," 2020, https://github.com/owl-project/ViSII/.

[11] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," *arXiv preprint arXiv:2103.00020*, 2021.

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

[13] 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.