Learning Adversarial Unsupervised Representations for Scenes from Undirected Interaction Data

Peter Wagstaff
pwagstaff@gatech.edu

Nadun Ranawaka Arachchige
nadun.ranawaka@gatech.edu

Varshith Sreeramdass
vsreeramdass@gatech.edu

Abstract: Robotic manipulation requires robots to build a concise representation of their environments. Specifically, representing spatial relationships between objects is important for performing certain manipulation tasks. Prior works have attempted to learn spatial relationships in visual input by using human or heuristic labels to provide supervision. In this paper, we propose an unsupervised object-centric representation learning method that attempts to learn object spatial relations from pixels without labels. Our representations are trained using reconstruction and adversarial objectives. These objectives are formulated to explicitly encourage the representations to include more information about spatial relations among objects in a scene while reducing object-specific information. We evaluate our learned scene representations in the context of spatial reasoning tasks common in manipulation. Our evaluation shows that models trained on our scene representations outperform those trained directly on pixels as well as other representations and generalize to objects with unseen shapes and colors.

Keywords: Robot Learning, Object-centric Representation, Unsupervised Learning

1 Introduction

To communicate and execute complex manipulation tasks, robots as well as us humans need to construct a representation of the environment and objects within it which captures and distills meaningful information to facilitate decision-making. This is in contrast to making decisions directly from raw sensory input. These representations should allow easier understanding while preserving vital properties that govern how decisions should be made. These representations can also be called "latent embeddings" since they attempt to capture the "hidden" generative process of the environment. One way of representing an environment is object-centric embeddings, in which the environment is primarily represented by the properties of the objects within it, and the relationships among those objects.

Building object-centric scene representations has shown great promise in a number of robotic applications [1, 2, 3, 4, 5, 6]. This paper's focus is specifically on building representations of spatial relationships between objects (e.g., left, right). We focus on learning the spatial relationships between objects since there is evidence that knowing the exact poses and characteristics of objects is not necessary for robotic manipulation [2]. Furthermore, by focusing on spatial relationships, we are simplifying the learning problem for sequential manipulation and rearrangement. We hypothesize that this will result in better downstream sample efficiency and generalization.

All prior works that capture object relations require human-labeled datasets to learn them. We hypothesize that object relationship representations can be learned without relation labels, just from raw data with object bounding boxes. We propose learning compact representations of objects and their relationships to other objects using reconstruction objectives and adversarial methods.

In addition to common reconstruction based representation learning, we leverage an adversary that attempts to reconstruct individual images from the pair embeddings. Training the embedding networks to maximize reconstruction while simultaneously fooling the adversary enables our pair em-

beddings to be more efficient and drop unnecessary individual object information. This is the main idea behind our proposed method. We implement our method on images generated in a simulator and show that embeddings containing relational information can be learned in an unsupervised manner. We show that learning from these embeddings can allow a robot to learn tasks depending on spatial relationships, such as "pick up the object to the left of the green cube," faster than learning from raw sensory input, all without human labeling.

This key technical insight enables us to use knowledge from existing GAN training methodologies and use them to learn compact pair embeddings. We propose Adversarially Disentangled Variational Autoencoding Representations (ADVAE) and find it to be successful at creating an embedding that maximized spatial relational information necessary to perform a downstream task of selecting objects based on their spatial relationship to a reference object. In summary, our contributions are:

1. A method to adversarially learn object-centric scene representations in an unsupervised manner;
2. Evaluation of our learned representations on spatial reasoning tasks in the context of robotic manipulation;
3. A comparison of our method to others for the same tasks.

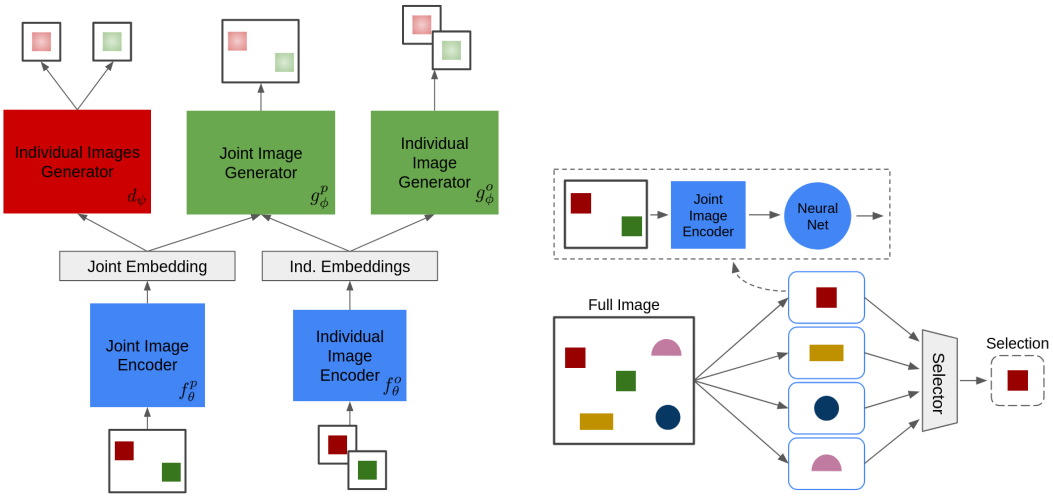

Figure 1: **Left**: Networks involved in our proposed method: **ADVAE**. Blue blocks represent encoding networks while green and red represent generator networks. Red represents the adversarial generator network. The reconstruction loss should decrease for the green networks and increase for the red. **Right**: Selecting object displaying the left spatial relationship using the joint image encoder. The same neural network is used to embed every pair image.

2 Related Works

Unsupervised Object Representations. A number of prior works have explored using unsupervised or self-supervised methods for learning object representations or object-centric scene representations. Grasp2Vec [4] looks at images of a scene before and after grasping an object, and uses the difference between them to learn the object representation in a self-supervised way. However, Grasp2Vec relies on access to a robot either in the real world or in simulation, which can make data collection expensive.

Wu et al. developed a model which encoded possible objects in an RGB image as latent variables, with the object hypotheses being probabilistically refined over time by considering additional image frames [3]. SPACE [7] combines spatial attention and scene-mixture models to decompose a scene

into background and foreground components (objects). MONet [8] trains a recurrent attention network to output pixel-wise masks for the various elements of a scene, which are then modeled by a variational autoencoder. However, none of these works attempt to model spatial relations between objects in an unsupervised way.

Supervised Object Representations Past work has used region proposal networks to detect potential objects, with representations of task-relevant objects being learned through demonstrations [1]. More recently, SORNet [2] trained a model on a dataset of images with labeled object spatial relations, and showed generalization to unseen objects and situations without further training. In contrast, our method does not require demonstrations or a dataset with explicitly labeled relations. Song et al. applied a heuristic method to bounding boxes output from an object detector to classify object spatial relations [9]. However, such heuristics can be brittle and task-specific. We think that our latent embeddings of spatial relations can capture relations beyond what can be described from heuristics.

Visual Representation Learning for Robotic Manipulation Learning the correct visual representations of scenes, tasks, and actions is an important problem in robot manipulation. The correct scene representation can improve sample efficiency of learning, increase the success rate of the downstream task, provide generalization capabilities, or result in a combination of these benefits.

CLIP [10] is often used in robotic manipulation for semantic representations of the various components in a scene using natural language. Transporter Networks is an approach that greatly improved sample efficiency for robotic manipulation by reformulating the problem as trying to predict displacements in visual input [11]. Perceiver-Actor is another successful method that represents a robot workspace as voxels and then generates a latent embedding of the voxels which is used for downstream training of a language-conditioned imitation learning agent [12]. Xiao et al. used a masked autoencoder [13] to learn visual representations of images, which were then used as inputs to reinforcement learning models [14]. R3M is a similar approach that uses pre-training and a suite of learning methods to learn separated and clustered embeddings for manipulation [15].

3 Methodology

3.1 Problem Statement

We consider the problem of learning representations for a scene s containing a variable set of N objects. For objects $i, j \in 1, N$, bounding box images for individual objects o_i, o_j and pairs $p_{i,j}$ of size $(H \times W \times 3)$ are assumed to be available. We wish to learn individual and joint embeddings e_i, $e_{i,j}$ which form a representation of the scene. These representations can be leveraged in a number of downstream tasks such as:

- Conditioning a robot manipulation policy.
- As input to a visual question answering model.
- As input to a task and motion policy model.

3.2 Preliminaries

We will briefly explain vanilla variational auto-encoders (VVAEs) [16] as our method builds various components using them. For capturing latent representations $z \sim p_z(\cdot)$ for $x \in P$, which obey the generative distribution $p(x|z)$, a function p_ϕ modeled by an NN with parameters ϕ is used. To approximate the intractable prior $p(z|x)$, another function q_θ modeled by an NN with parameters θ is used. These two networks are trained by optimizing the ELBO term $\sum_{x \sim P} V(x, x, p_\phi, q_\theta)$ where

$$V(x, y, p_\phi, q_\theta) = \sum_{z \sim q_\theta(\cdot|x)} \left[\log p_\phi(y|z) - \log \frac{q_\theta(z|x)}{p_z(z)} \right]$$

The networks that approximate the posterior (q_ϕ) form the embedding networks and those that model the forward process (p_θ) form the generator networks.

3.3 Object-Centric Embeddings

We assume that the images are generated from a random (inaccessible) process which consists of two steps: (i) embeddings e_i, e_j are generated from an object prior $p^o(\cdot)$ (ii) joint embedding $e_{i,j}$ is generated from a pair prior $p^p(\cdot)$. Joint image $p_{i,j}$ containing o_i and o_j is generated from a conditional distribution $P(\cdot|e_{i,j}, e_i, e_j)$. To capture these embeddings from observed images, we propose an individual object embedding network f_θ^o maps the object image o_i to e_i and a joint object embedding network f_θ^p maps the joint image $p_{i,j}$ to $e_{i,j}$.

3.4 Unsupervised Learning

To capture useful information in the embeddings, we train the two networks in three different ways. The first two serve as baselines and the last serves as the proposed approach.

Autoencoding Representations (VAE) We use two additional generator networks g_ϕ^o and g_ϕ^p that construct the input images from the respective embeddings e_i and $e_{i,j}$. The training loss is similar to the above with two different terms for individual and joint embeddings.

$$L_{VAE}(\theta, \phi) = \sum_{s_k \in D} \sum_{i \in s_k} \left[V(o_i, o_i, g_\phi^o, f_\theta^o) + \sum_{j \in s_k} V(p_{i,j}, p_{i,j}, g_\phi^p, f_\theta^p) \right]$$

Disentangled Autoencoding Representations (DVAE) So as to obtain a compact joint embedding, we wish to encourage $e_{i,j}$ to contain information only pertaining to the relationship between the two objects and no object-specific information. This is done by conditioning the joint generator network g_ϕ^p on the individual object embeddings e_i, e_j in addition to the embedding $e_{i,j}$. The corresponding loss function would look similar to the above loss, except for the reconstructed joint image which would take the form $g_\phi^p(e_{i,j}, e_i, e_j)$. This removes the incentive for the network f_θ^p to embed information pertaining to individual objects and instead focus on the relationship between the two objects.

Adversarially Disentangled Representations (ADVAE) To further encourage the joint embeddings to not have information pertaining to the individual objects, we leverage a flipped adversarial training mechanism [17]. An additional object-s generator network d_ψ is trained to generate the individual object images from the joint embedding $\hat{o}_i, \hat{o}_j = d_\psi(e_{i,j})$ while the joint embedding network f_θ^p is trained to prevent d_ψ from succeeding. ψ is obtained by minimizing L_{ADV} while θ and ϕ are obtained by minimizing L_{GEN}. Gradient steps are taken alternatively until equilibrium is attained.

$$L_{ADV}(\theta, \psi) = \sum_{s_k \in D} \sum_{i,j \in s_k} V(p_{i,j}, (o_i, o_j), d_\psi, f_\theta^p)$$

$$L_{GEN}(\theta, \phi) = L_{VAE}(\theta, \phi) - L_{ADV}(\theta, \psi)$$

The motivation behind disentangling representations is to ensure that spurious correlations between object features and joint embeddings are removed. The network architecture is shown in Figure. 1.

3.5 Implementation

To build our various models, we used existing implementations of VAEs [18] created using PyTorch Lightning based on the original implementation of a VAE [16]. Each VAE was trained for 150 epochs. Training took around 1 hour for each model on a single RTX 3090 GPU.

The encoder portion of our VAE had the following architecture:

- Conv2D(3, 32, 3, 2, 1)
- Conv2D(32, 64, 3, 2, 1)
- Conv2D(64, 128, 3, 2, 1)
- Conv2D(128, 256, 3, 2, 1)
- Conv2D(256, 512, 3, 2, 1)

The parameters for the Conv2D are Conv2D(in_channels, out_channels, kernel_size, stride, padding). Additionally, there were ReLU activations and batch-normalization after each conv layer. The outputs from the conv layers are then passed into a linear layer of size *latent_dim* to get the mean and variance of the latents. The decoder section had the same architecture but in reverse, using transpose convolution layers. Hyperparameters, which were the same for all models, are listed in Table 1. While we were unable to do an extensive hyperparameter search, we did tune the number of latent dims of the VAE to ensure that it was providing enough information for the downstream task. This was done by analyzing the loss and qualitative results on our datasets as well as the existing CelebA dataset which is typically used as a benchmark for VAEs. [19].

Hyperparameter	Value
batch size	4
image size	64x64
VAE latent dims	128
lr	0.0003
kld penalty weight	0.00025
lr_scheduler	ExponentialLR
scheduler gamma	0.95
optimizer	Adam

Table 1: VAE hyperparameters

4 Evaluation

In our evaluation, we empirically confirm the following hypotheses: (1) Classic reconstruction embedders cannot capture relation-specific information efficiently, (2) Training separate embedders for individual and pair images help isolate relation-specific information, and (3) Adversarial training mechanisms enhance relation specific information by destroying individual object information.

4.1 Setup, Data Generation

We evaluate the method in a tabletop manipulation scene with objects of different shapes and colors, placed at various locations. In the interest of simplifying the evaluation, we limit ourselves to pair images where a particular green cube, selected as the reference object, is always present. The embeddings are evaluated by their performance in capturing information in relation to the reference object.

We use code from PandaGym [20] for spawning objects on the table at desired locations and generating images and bounding boxes from the default rendering engine. To gather data, we spawn the green cube in a random location around the center of the table. We then spawn four other objects (red cube, blue cylinder, yellow rectangular prism, pink half-sphere) in relation to the green cube, by selecting a random angle (with world x-axis) and distance from the reference cube. The angle was selected such that no object having an angle within 30 degrees of another was possible. Purely for the purpose of evaluation, ground truth labels that inform spatial relations "to the left of" and "to the top of" were created based on heuristics using the recorded angle information. We collected data for 2000 table arrangements, resulting in 10,000 individual images and 8000 joint images, angles, and labels. We scale all images to 64x64 so as to simplify the network design, which results in some distortions since the aspect ratio of images varies greatly.

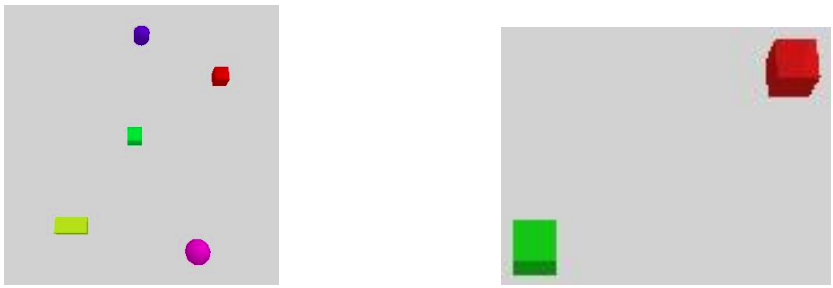

Figure 2: Examples of objects arranged on the table, and a joint image.

4.2 Visualization of embeddings

We train the three embedders (VAE, DVAE, and ADVAE) on a dataset of individual and pair images of the five known objects.

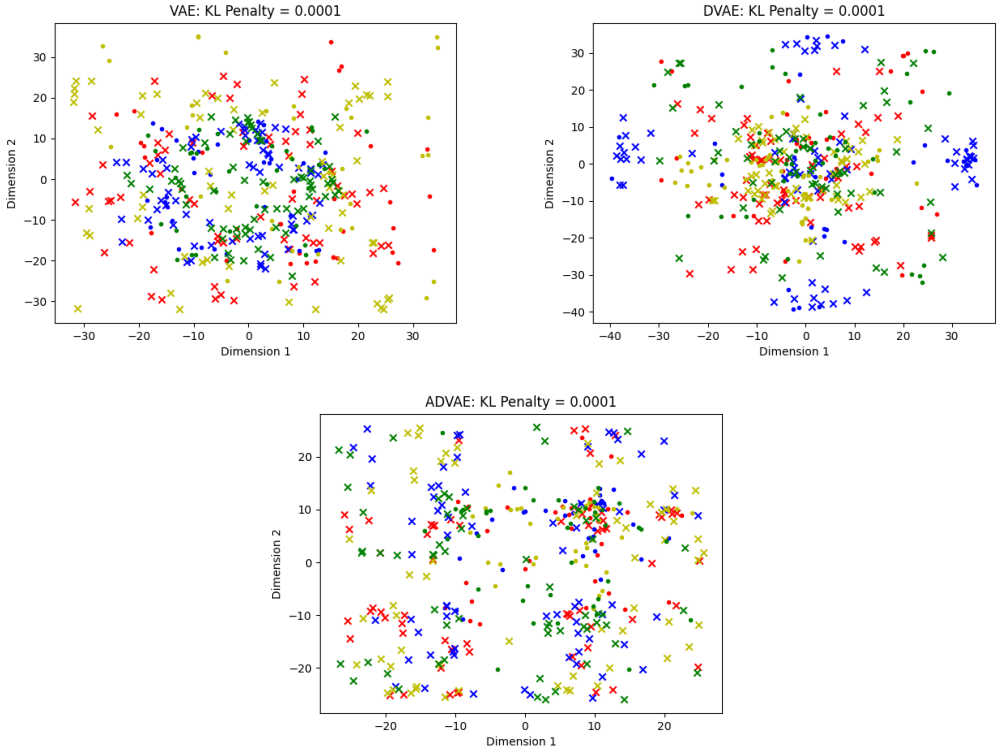

Figure 3: Plot of reduced dimensionality latent representations for joint images (object and reference object). Each mark corresponds to an object. X's are objects with a primarily top or bottom relationship with the reference object and O's are objects with a primarily left or right relationship. Colors denote the object type of the same color (except green which is for the pink half-sphere).

For a set of 100 table arrangements and their 400 resulting joint images, we construct scatter plots of embeddings produced by the three embedders. The dimensionality is reduced to two by T-distributed Stochastic Neighbor Embeddings [21] shown in Figure 3). VAE and DVAE plots show no separation based on spatial relationship as indicated by X's and O's. However, there is clear separation of the objects based on their object type and color, as seen by the clustering of colors. Additionally, in the VAE plot, the blue and green marks - indicating similarly shaped objects (cylinder, half-sphere) - are close to each other, while the yellow and red marks - indicating similar shapes (cube, rectangular prism) - are close. By contrast, in the ADVAE plot, we see no clustering based on object

type. However, we do observe some clustering based on spatial relationship with O's being closer to the center of the image and X's being on the edges. This suggests that the ADVAE successfully removed the non-spatial-relational information which prevents it from clustering by object type but promotes clustering based on spatial relationship.

4.3 Downstream decision making

In this section, we leverage the representations built by our models for downstream decision making. An expert provides demonstrations where they choose an object that satisfies a target relationship with the reference object. Our task is to learn to choose the object that satisfies the demonstrated target relationship in scenarios that are unseen and may have a different number or types of objects.

Training Procedure Using representations of the different object-reference pairs using the embedding networks as features, we train the classification model with ground truth labels to choose the object that most satisfies the target relationship. In our evaluation, we fixed the target relationship to "to the top of". We compare our embeddings against baseline representations: (1) displacements in image space relative to reference object and (2) raw pixels corresponding to the joint images. Our classification model individually takes as input features of the objects and outputs an energy value for each object. These energies are used as logits along with the ground truth label for cross-entropy loss, which is used to train the classification model weights. We choose to process features independently for two reasons: (1) to make our models independent of the number of objects in the scene, and (2) to allow our baseline model which uses raw pixel displacements to serve as a reliable upper bound on the generalization performance.

Conditions for testing generalization We train the classification models in three conditions based on the types of objects in the train/val splits and the test splits as seen in Table 2. Condition 0 evaluates the performance of the models if the train and test distributions include the same set of objects. This represents the easiest condition and is included as a baseline condition. In condition 1, the classification model trains on blue and pink round objects and is tasked to generalize to red and yellow boxy objects. With both unseen shapes and colors, this represents a hard condition. In condition 2, the model trains both boxy and round objects and is tasked to generalize to unseen colors. **Note** that the classification model sees all objects as input in all conditions. However, only the specified objects satisfy the target relationship. We re-iterate that our embedders are trained on images with all the objects.

Condition	Train/Val	Test
0	All	All
1	BC, PC	RB, YB
2	RB, BC	YB, PC

Model / Condition	0	1	2
Displacement	99.5%	99.4%	99.3%
Pixels (ftuned)	94.3%	4.2%	18.4%
VAE (ftuned)	95.3%	0.0%	35.4%
DVAE (ftuned)	96.3%	9.1%	24.6%
ADVAE (ftuned)	**97.3**%	21.7%	40.0%
Pixels (frozen)	94.3%	45.9%	<u>50.2</u>%
VAE (frozen)	91.8%	10.1%	46.3%
DVAE (frozen)	88.1%	44.4%	42.7%
ADVAE (frozen)	92.8%	**66.0**%	**59.3**%

Table 2: **Left**: Table represents the objects considered in the training and test splits. In the two-letter code, the first letter indicates color (R=Red, B=Blue, Y=Yellow, P=Pink), and the second indicates shape (B=Boxy, C=Round (Half-sphere, Cylinder)). 'All' indicates the set of all four objects. **Right**: Table shows the results of the classification model's performance, and accuracy measured on the test splits under the three conditions. The highest accuracies under each condition ignoring those for 'Displacement' are made **bold**.

Our main results are shown in Table 2. Classification which uses image displacements as features performs and generalizes best, as there is no information included that pertains to independent object attributes. This allows for the model to generalize exceedingly well (close to 99%), serving as an

upper limit. 'Pixels' performs best on Condition 0, which requires no generalization. While the performance of models other than VAE is close to each other on Conditions 1 and 2, we hypothesize that VAE performs poorly because of having to embed information about the entire pair image into just one embedding, overfitting to individual object attributes - pixels corresponding to object form majority of the image input. DVAE is better than VAE in capturing features as the performance under Condition 1 indicates. Ultimately, our proposed method ADVAE is competitive in Condition 0 and outperforms the other models in 1 and 2. This shows that ADVAE best isolates relational information in the embeddings providing a compact and useful representation of the scene.

Additionally, we see that allowing for the embedders to be finetuned for the classification task universally hurts performance for Conditions 1 and 2 indicating that models can overfit very easily. Surprisingly, we see that randomly initialized embedders provide a reasonably good representations for the purpose of classification.

4.4 Miscellaneous Experiments

k-NN Classification We evaluated if the embeddings could be used for classification without a neural network classifier. We chose a fit a lightweight k-Nearest Neighbor classifier [22]), trained it on 400 samples of representations of the scene for each type of relation, and then tested the classifier on held-out samples. Note that these results shown in Table 3, are on a different dataset which includes a moving robot arm in the images.

Model type/ k-NN Accuracy	Top-Bottom Relations	Left-Right Relations
Raw Pixels	74.89%	46.48%
VAE	89.3%	94.1%
DVAE	91.0%	94.5%
ADVAE	57.31%	76.11%

Table 3: k-NN Classifier accuracy on latents of spatial relation images

The high accuracy of the classifier is with DVAE representations showing that the proposed embeddings contain sufficient information relevant to spatial relationships among the objects. Furthermore, the classifiers on latent embeddings generally outperform those on images (except for the ADVAE model) which shows that the image embeddings with spatial relation information augmentation are effective representations. To ensure that the classifier was not just overfitting to our dataset, we also tested the classifier on random labels, and the accuracy was not better than random chance.

Predicting pick-place locations We attempted to evaluate using a pick-and-place task in a simulated environment. Given an image of the scene before manipulation and pick and place locations in 2D space, the task was to learn a policy that outputs these locations.

We eventually abandoned this approach as it introduced additional complexity to the evaluation without offering a way to distinguish the various models. This is because predicting pick and place locations would require having global knowledge about the objects (e.g., locations). Including this as input provides a "shortcut" for the policy to regress the pick and place locations, completely ignoring the pair representations. Hence, we instead performed the evaluation presented in the above section where the relevant object is to be chosen. It would be easy enough to translate this to actual pick locations, but as such an evaluation is not indicative of the performance across models, we do not perform this.

5 Conclusion

We present a methodology for obtaining compact representations for scenes with a variable number of objects, which are aware of spatial relations among objects. We do this in an unsupervised manner without the use of human labels by leveraging reconstruction and adversarial learning objectives. We

show that the embeddings of object pairs thus obtained robustly capture spatial relations and offer generalization capabilities as compared to baseline methods.

Limitations Currently, our model captures spatial relations between only a pair of objects. Additionally, it assumes access to an effective object detector to provide the ground truth bounding boxes for the different objects in a scene. The effectiveness of the scene representation deteriorates with the occlusion of the reference or manipulated object. If there are multiple objects which satisfy the same spatial relation, the embedding learning might fail. Our current pipeline requires that the images are all scaled to the same sizes, thought this can be addressed by using appropriate pooling layers in the embedders.

Extensions To establish the robustness of the representations, they should be evaluated on low-level control tasks. Since the scene representation obtained is variable in size, this would require using models that handle variable inputs, such as LSTMs [23] or Transformers [24].

Applying to Real Robots Our method can be readily applied to real-world robot systems. Once the scene representation embedders are trained, they can be used as the vision modules for the robots, and manipulation policies can be trained on the embeddings. This can be done in the context of imitation learning through provided demonstrations or even using reinforcement learning. The embedders should ideally be trained on a dataset of real-world objects in various configurations.

6 Team Contributions

Table 4: Project Contributions.

Team Member	Contribution
Peter	ADVAE idea, environment and dataset generation, scatter plot generation and analysis
Nadun	Dataset processing, training VAEs, analysis of scene embeddings and k-NN
Varshith	Implementation of the three VAE models, implementation, training, evaluation of classification models, formulation of training schemes, pick-place evaluation
All	Writing, literature review

References

[1] 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.

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

[3] Y. Wu, O. P. Jones, M. Engelcke, and I. Posner. Apex: Unsupervised, object-centric scene segmentation and tracking for robot manipulation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3375–3382. IEEE, 2021.

[4] E. Jang, C. Devin, V. Vanhoucke, and S. Levine. Grasp2vec: Learning object representations from self-supervised grasping. *arXiv preprint arXiv:1811.06964*, 2018.

[5] Y. Wang, J. Wang, Y. Li, C. Hu, and Y. Zhu. Learning latent object-centric representations for visual-based robot manipulation. In *2022 International Conference on Advanced Robotics and Mechatronics (ICARM)*, pages 138–143. IEEE, 2022.

[6] Y. Huang, A. Conkey, and T. Hermans. Planning for multi-object manipulation with graph neural network relational classifiers. *arXiv preprint arXiv:2209.11943*, 2022.

[7] Z. Lin, Y.-F. Wu, S. V. Peri, W. Sun, G. Singh, F. Deng, J. Jiang, and S. Ahn. Space: Unsupervised object-oriented scene representation via spatial attention and decomposition. *arXiv preprint arXiv:2001.02407*, 2020.

[8] 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.

[9] X. Song, S. Jiang, B. Wang, C. Chen, and G. Chen. Image representations with spatial object-to-object relations for rgb-d scene recognition. *IEEE Transactions on Image Processing*, 29: 525–537, 2019.

[10] 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. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[11] 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. In *Conference on Robot Learning*, pages 726–747. PMLR, 2021.

[12] M. Shridhar, L. Manuelli, and D. Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. *arXiv preprint arXiv:2209.05451*, 2022.

[13] 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*, pages 16000–16009, 2022.

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

[15] 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.

[16] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[17] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[18] A. Subramanian. Pytorch-vae. https://github.com/AntixK/PyTorch-VAE, 2020.

[19] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[20] Q. Gallouédec, N. Cazin, E. Dellandréa, and L. Chen. panda-gym: Open-Source Goal-Conditioned Environments for Robotic Learning. *4th Robot Learning Workshop: Self-Supervised and Lifelong Learning at NeurIPS*, 2021.

[21] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[22] K. Taunk, S. De, S. Verma, and A. Swetapadma. A brief review of nearest neighbor algorithm for learning and classification. In *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, pages 1255–1260. IEEE, 2019.

[23] A. Graves and A. Graves. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, pages 37–45, 2012.

[24] 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.

[25] P. R. Florence, L. Manuelli, and R. Tedrake. Dense object nets: Learning dense visual object descriptors by and for robotic manipulation. *arXiv preprint arXiv:1806.08756*, 2018.

[26] N. Heravi, A. Wahid, C. Lynch, P. Florence, T. Armstrong, J. Tompson, P. Sermanet, J. Bohg, and D. Dwibedi. Visuomotor control in multi-object scenes using object-aware representations. *arXiv preprint arXiv:2205.06333*, 2022.

[27] T. Migimatsu and J. Bohg. Grounding predicates through actions. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 3498–3504. IEEE, 2022.

[28] R. Gao, D. Jayaraman, and K. Grauman. Object-centric representation learning from unlabeled videos. In *Computer Vision–ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part V 13*, pages 248–263. Springer, 2017.

[29] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[30] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun. Graph neural networks: A review of methods and applications. *AI open*, 1:57–81, 2020.

[31] G. Mena, J. Snoek, S. Linderman, and D. Belanger. Learning latent permutations with gumbel-sinkhorn networks. In *ICLR 2018 Conference Track*, volume 2018. OpenReview, 2018.

[32] K. Greff, R. L. Kaufman, R. Kabra, N. Watters, C. Burgess, D. Zoran, L. Matthey, M. Botvinick, and A. Lerchner. Multi-object representation learning with iterative variational inference. In *International Conference on Machine Learning*, pages 2424–2433. PMLR, 2019.

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

[34] 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.