

# SlotGNN: Unsupervised Discovery of Multi-Object Representations and Visual Dynamics

Alireza Rezazadeh<sup>1</sup>, Athreyi Badithela<sup>2</sup>, Karthik Desingh<sup>2\*</sup>, Changhyun Choi<sup>1\*</sup>

**Abstract**—Learning multi-object dynamics from visual data using unsupervised techniques is challenging due to the need for robust, object representations that can be learned through robot interactions. This paper presents a novel framework with two new architectures: *SlotTransport* for discovering object representations from RGB images and *SlotGNN* for predicting their collective dynamics from RGB images and robot interactions. Our *SlotTransport* architecture is based on slot attention for unsupervised object discovery and uses a feature transport mechanism to maintain temporal alignment in object-centric representations. This enables the discovery of slots that consistently reflect the composition of multi-object scenes. These slots robustly bind to distinct objects, even under heavy occlusion or absence. Our *SlotGNN*, a novel unsupervised graph-based dynamics model, predicts the future state of multi-object scenes. *SlotGNN* learns a graph representation of the scene using the discovered slots from *SlotTransport* and performs relational and spatial reasoning to predict the future appearance of each slot conditioned on robot actions. We demonstrate the effectiveness of *SlotTransport* in learning object-centric features that accurately encode both visual and positional information. Further, we highlight the accuracy of *SlotGNN* in downstream robotic tasks, including challenging multi-object rearrangement and long-horizon prediction. Finally, our unsupervised approach proves effective in the real world. With only minimal additional data, our framework robustly predicts slots and their corresponding dynamics in real-world control tasks. Our project webpage: [bit.ly/slotgnn](http://bit.ly/slotgnn).

## I. INTRODUCTION

Studies suggest that the human visual system identifies conceptually distinct visual features, indexes their locations [1], and utilizes this information as the foundation for higher-level cognitive processes, such as comprehending and interacting effectively with the world [2]. A similar principle guides many robotic systems for goal-directed motor planning. In multi-object manipulation, early approaches aimed to directly project the image observation into a unified lower-dimensional space to infer the dynamics [3], [4]. However, such strategies do not reflect the inherent structure of a multi-object system and lack object-level predictions. This limitation not only impedes the model’s ability to learn object interactions but also results in inaccurate dynamics predictions. Addressing this limitation, recent methods build dynamics models by decomposing the

<sup>1</sup> Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455 USA (rezaz003@umn.edu, cchoi@umn.edu).

<sup>2</sup> Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455 USA (kdesingh@umn.edu, badith004@umn.edu).

\* Equal supervision.

<sup>†</sup>This project is partially funded by the UROP Program at the University of Minnesota and the MnDRIVE UMII (University of Minnesota Informatics Institute) Seed Award. This project was also supported in part by the Sony Research Award Program and NSF Award 2143730.

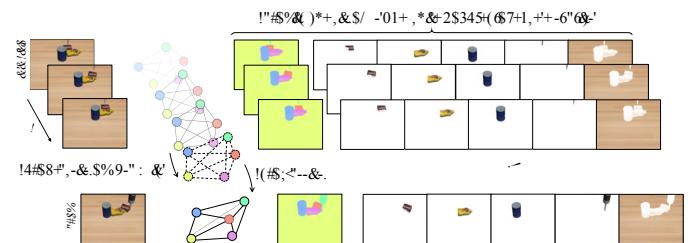


Fig. 1. Overview of our unsupervised framework. (a) We introduce **SlotTransport** to identify temporally-aligned, object-centric slots, that each consistently represents a unique visual element. (b) We introduce **SlotGNN**, a graph-based model that learns scene dynamics from slots and predicts future states based on the robot’s action. (c) Our unsupervised approach facilitates planning to transition from an initial state to a goal image without requiring extensive ground-truth supervision.

observation into object-specific lower-dimensional latents and subsequently learning dynamics within these “object-centric” representations [5], [6], [7], [8], [7]. For multi-object systems, recent studies emphasize the effectiveness of learning object-centric representations to enhance the accuracy and sample efficiency of dynamic models [5], [6]. This category of models follows a natural formulation by first learning to represent a scene as a set of object-centric features and then learning the dynamics among them.

In robotics, unsupervised learning of object dynamics is a key challenge particularly given its significance in model-based action planning for real-world applications. Nevertheless, the majority of existing methods of learning multi-object dynamics heavily rely on ground-truth information, including object pose [9], [10], [6] and segmentation masks [6], [5]. This substantially restricts the applicability of such solutions in real-world settings where comprehensive ground-truth information is often unavailable. To address this challenge, our work focuses on discovering unsupervised object representations in multi-object scenarios and harnessing these representations to understand their dynamics. Our primary contributions include: (1) We introduce **SlotTransport** for unsupervised object discovery, a novel architecture that refines object-centric representation learning through slot attention [11]. Utilizing a feature transport mechanism, *SlotTransport* ensures temporal alignment of the object-centric representations. The discovered slots capture scene composition, each depicting a visual entity in a multi-object scene, such as objects, the background, and the robot. Notably, each slot maintains a consistent association with a distinct object, even when it’s occluded or absent. (2) We propose **SlotGNN**, a novel unsupervised graph-based model for predicting multi-object scene dynamics from object-centric representations. *SlotGNN* uses slots identified by *Slot-*

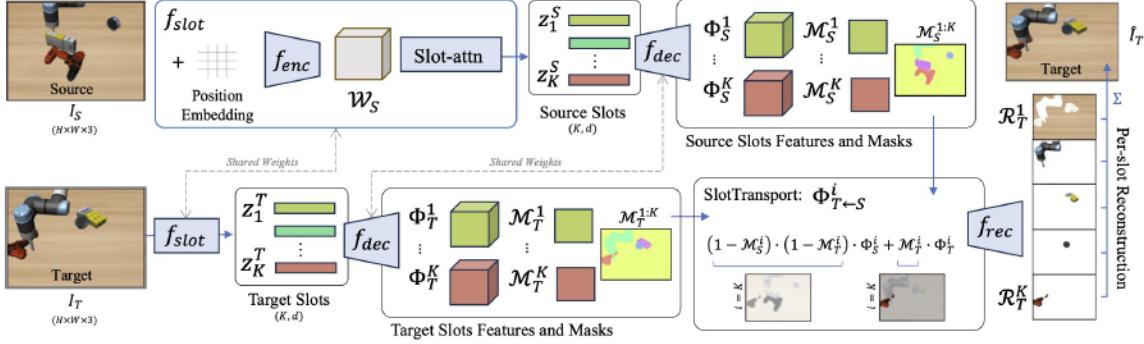


Fig. 2. **SlotTransport**: Unsupervised Multi-Object Discovery. From an RGB image,  $f_{slot}$  identifies object-centric slots  $z_{1:K}$ . Through slot attention [11], slots bind to visual features, and  $f_{dec}$  produces feature maps  $\Phi^i$  and masks  $\mathcal{M}^i$ . Temporal alignment is ensured by transporting slot features between source and target images.  $f_{rec}$  reconstructs each object slot  $\mathcal{R}_T^i$ , which together compose the target image. The model is trained using only reconstruction error. During inference, objects are reconstructed from a single image using learned slots.

Transport to synthesize the scene’s future appearance based on the robot’s actions. With the temporal alignment from SlotTransport, the scene transforms into a graph where each node consistently represents a slot, and edges capture the slot interactions. SlotGNN performs relational reasoning on the graph and learns to project the future appearance of each slot. (3) We examine the dynamics learned with SlotGNN in challenging downstream robotic tasks. We employ SlotGNN for challenging goal-directed multi-object rearrangement using pushing actions and long-horizon dynamics prediction.

(4) Demonstrating the real-world applications of our unsupervised approach, we successfully transfer SlotTransport and SlotGNN, initially trained in simulation, to the real robot by collecting a minimal dataset of just 20 real robot demonstrations (%5 of the amount of simulated training data).

Our results demonstrate the robustness of our unsupervised framework, particularly in downstream robotic applications and real-world scenarios. Our approach consistently predicts accurate multi-object representations and their corresponding dynamics. Throughout this paper, we will use the terms ‘slots’ and ‘object-centric representations’ interchangeably.

## II. RELATED WORK

**Learning Multi-object Dynamics Model:** Early models for graph-based dynamics, such as Interaction Networks (IN) [9], [10] and follow-up adaptations [12], [13], [14], represent a multi-object system with a graph where each node is an object ground-truth state (e.g., position, velocity, mass, friction). These models rely on explicit state information. However, for real-world robotic scenarios, obtaining ground-truth state data is infeasible. Recent methods have focused on learning object representations. Each object is mapped to a lower-dimensional, object-centric representation. The representations are typically a combination of explicit ground-truth states, like position, bounding box, and mask, combined with implicit visual features [15], [7], [16], [6], [5]. However, the primary assumption of the ground-truth state supervision limits their application in the real world. In contrast, our work introduces an unsupervised framework for learning multi-object scene

dynamics based on discovering unsupervised slots. This eliminates the need for explicit ground-truth state supervision.

**Unsupervised Object-centric Representation:** Our work builds on learning object-centric representations using slot attention [11]. Slot attention interfaces with visual outputs to generate a set of slots. For robotics applications, ensuring temporal consistency in these slots is vital for accurate scene dynamics understanding. This consistency is essential for formulating a planning objective or training loss. Thus, we explicitly incorporate a feature transport mechanism in our SlotTransport to maintain consistency across image pairs from different observation timesteps inspired by [17]. In recent work on unsupervised multi-object dynamics, keypoint extraction has been explored [18]. However, keypoint-based dynamics face challenges with occlusion, a common issue in robotics scenarios. Our SlotTransport addresses this problem, reliably associating each slot with specific objects even under heavy occlusions.

## III. METHODS

Our framework has two main components:

(1) **SlotTransport**: An unsupervised multi-object discovery model that efficiently extracts robust and temporally consistent object-centric representation slots from multi-object scenes.

(2) **SlotGNN**: Building on top of the slot discovery, this unsupervised graph-based model learns the dynamics of the object-centric representations. Importantly, SlotGNN is conditioned on the robot’s action which enables applications such as model-based action planning.

### A. SlotTransport: Unsupervised Multi-Object Discovery

The SlotTransport’s role is to map the image to underlying object-centric representations. A detailed architecture of SlotTransport is shown in Fig. 2. We build on the slot attention [11] to extract slots from image frames while ensuring temporal alignment of the slots. Given an RGB image  $I$ , a convolutional encoder  $f_{enc}$  augmented with positional embeddings, maps the image to an intermediate representation of  $\mathcal{W} \in \mathbb{R}^{H \times W \times 3}$ .

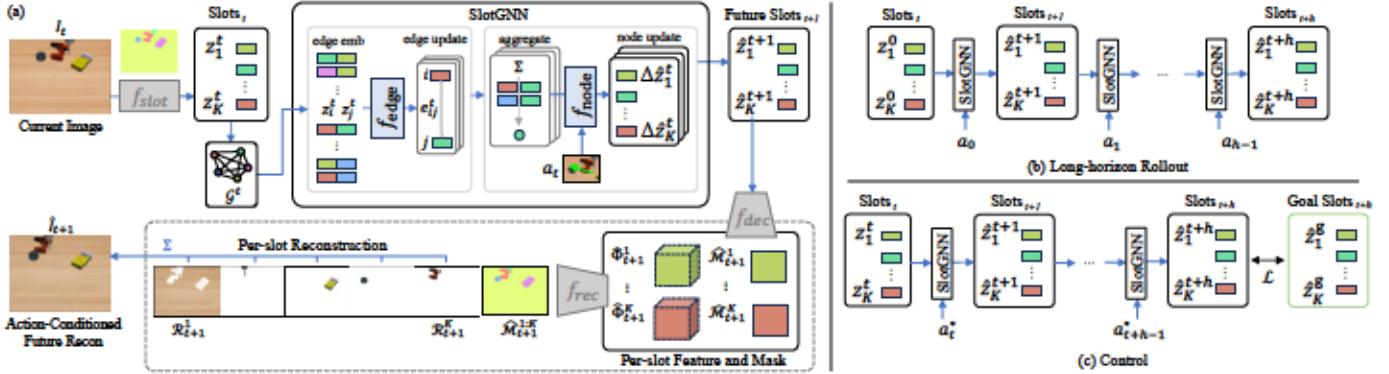


Fig. 3. SlotGNN: Unsupervised Multi-Object Dynamics. (a) SlotGNN predicts slot changes  $\Delta z_i^t$  after applying a pushing action  $a_t$ . Using SlotTransport’s slots ( $f_{slot}$ ), a graph  $\mathcal{G}_t$  is formed with slots as nodes and slot interactions as edges. Edges and nodes are updated via  $f_{edge}$  and  $f_{node}$ , resulting in next timestep slots  $\hat{z}_i^{t+1}$ . The next image  $I_{t+1}$  is then reconstructed. (b) With a sequence of robot actions, SlotGNN projects future multi-object dynamics and synthesizes future scenes. (c) SlotGNN also facilitates goal-directed planning to optimize actions towards a desired goal.

Using the slot attention [11], slots  $z_{1:K} \in \mathbb{R}^d$  are derived that uniquely represent distinct portions of  $\mathcal{W}$ .

We recognize that our ultimate goal of learning dynamics in an object-centric latent space requires temporal consistency of the slots. To explicitly enforce this, we introduce the transport mechanism in SlotTransport to establish temporal alignment in slots. Inspired by [17], this mechanism transports slot features between a pair of source and target images ( $I_S, I_T$ ) sampled from a given scene.

For each image, slots are extracted as  $(z_{1:K}^S, z_{1:K}^T)$ . First, using a convolutional decoder  $f_{dec}$ , we decode each slot into a feature map  $(\Phi_S^{1:K}, \Phi_T^{1:K}) \in \mathbb{R}^{h \times w \times m}$  and an alpha mask  $(\mathcal{M}_S^{1:K}, \mathcal{M}_T^{1:K}) \in \mathbb{R}^{h \times w}$ . These alpha masks serve as mixture weights to inpaint each slot’s feature map from the target image onto the source image (see Fig. 2). We produce a transported feature map  $\Phi_{T \leftarrow S}$  by nullifying the source feature map outside the slot’s predicted mask for both the target and source  $(1 - \mathcal{M}_T^t) \cdot (1 - \mathcal{M}_S^t) \cdot \Phi_S^t$ , followed by overlaying the masked target feature map  $\mathcal{M}_T^t \cdot \Phi_T^t$ . Finally, a convolutional reconstruction module  $f_{rec}$  reconstructs each slot as an RGB image  $\hat{R}_T^t \in \mathbb{R}^{h \times w \times 3}$  based on the transported feature map. The reconstructed slots together reconstruct the target image  $\hat{I}_T$ .

The transport mechanism is SlotTransport enforces temporal alignment between image pairs during training. Notably, the learned slots through SlotTransport consistently register to a unique object even under heavy occlusion or absence of an object. During inference, SlotTransport can discover and reconstruct slots from a single image by directly reconstructing the extracted per-slot features and masks. SlotTransport ensures that each slot’s feature map aligns well with its mask for learning consistent object representation across time. Importantly, this temporal alignment is achieved without adding additional learnable parameters; the same  $f_{slot}$  and  $f_{dec}$  are used when processing both source and target images.

### B. SlotGNN: Unsupervised Multi-Object Dynamics

The main purpose of SlotGNN is to learn the dynamics and model interactions between the visual elements in a multi-

object scene, such as the robot, objects, and the background. It does so using a graph-based representation, where each object-centric slot corresponds to a node in the graph. Crucially, SlotGNN enables learning unsupervised multi-object dynamics, eliminating the need for supervised trajectory labels that require access to the system’s ground-truth state. This feature becomes essential in real-world scenarios where obtaining accurate ground-truth data is challenging or impractical. Refer to the detailed architecture illustrated in Fig. 3-a.

Given an image  $I_t$  with its associated slots  $z_{1:K}^t$  discovered through SlotTransport, we construct a fully connected graph  $\mathcal{G}_t = (\mathcal{V}, \mathcal{E})$ . Each node  $v_i \in \mathcal{V}$  in the graph represents a slot, and each edge  $e_{ij} \in \mathcal{E}$  represents the interaction between the pair of slots  $z_i, z_j$ . For each node, we associate an embedding  $n_i$  which is initialized with the slot representations  $z_i$ . The edge embeddings, representing interactions, are initialized based on augmenting the connected slots representations. To process the information in the graph representation, SlotGNN employs a message-passing neural network architecture [19], [10] to update node and edge embeddings. Incoming information from neighboring nodes is aggregated to update each node’s state, capturing the dynamics and interactions in the scene.

The message-passing operation in the graph consists of two primary steps (see 3). First, the edge embeddings,  $e_{ij}$ , are updated based on their connecting node embeddings:  $e'_{ij} \leftarrow f_{edge}(e_{ij}, n_i, n_j)$ . Secondly, the node embeddings are updated using the updated edge embeddings associated with them and the robot action:  $n'_k \leftarrow f_{node}(n_k, \sum_{i \in \mathcal{N}(k)} e'_{ik}, a_t)$ . Here,  $f_{edge}$  and  $f_{node}$  are multi-layer perception update functions for edges and nodes, respectively.  $\mathcal{N}(k)$  denotes the neighbors of node  $k$ , which in the context of a fully connected graph is all other nodes. To condition on external action, the robot action  $a_t \in \mathbb{R}^4$ , characterized as a point-to-point push vector in image coordinates, is integrated as an input to  $f_{node}$  in SlotGNN. This ensures the learned dynamics are conditioned on the robot’s action and can be used for planning in the downstream robotics control task.

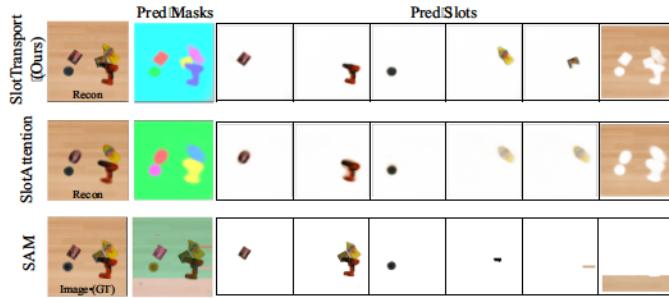


Fig. 4. Visualizations of per-slot masks and reconstructions. SlotTransport exhibits superior performance in accuracy and consistency of object-centric representation, even under occlusion, compared to the SlotAttention baseline [11]. We also showcase predicted segments from SAM [20].

After message-passing, the updated node embeddings are used to predict the evolution of slots in the next timestep conditioned on the action . This allows for the rollout of the dynamics into the future and enables synthesizing the future appearance of the scene.

### C. Training SlotTransport and SlotGNN

SlotTransport is trained using only the image reconstruction error for supervision. The image reconstruction loss  $\mathcal{L}_{\text{Rec}}$ , is defined using a pixel-wise Mean Squared Error (MSE) between the target and reconstructed images. In Fig. 2, modules with learnable parameters are distinctly highlighted in blue. Once SlotTransport is trained, it supervises the training of SlotGNN to learn visual dynamics.

We use a slot prediction MSE loss to train SlotGNN,  $\mathcal{L}_{\text{Pred}}$ . This loss reduces the distance between the slots directly predicted from the next timestep image using SlotTransport and slots from the single-step dynamics with SlotGNN, as visualized in Fig. 3-a (modules with learnable parameters are highlighted in blue). Importantly, employing a per-slot prediction loss requires temporal alignment that is ensured through SlotTransport. Furthermore, we use the single-step slot dynamics to reconstruct the image and also minimize the image reconstruction MSE loss  $\mathcal{L}_{\text{Rec}}$ .

### D. Long-Horizon Multi-Object Dynamics Rollout

Given a sequence of robot pushing actions  $\mathcal{A}$ , a learned SlotGNN rolls out the dynamics of a multi-object scene based only on the initial image frame  $I_0$ . As shown in Fig. 3-b, the model’s single-step predictions are cascaded to generate slots over extended future horizons. This capability for accurate dynamics rollout is possible due to the temporal alignment achieved with SlotTransport. Furthermore, for any given future timestep,  $I_t$  can synthesize an image of the scene.

### E. Goal-Directed Planning

Learning the scene dynamics facilitates goal-directed sequential action planning for multi-object manipulation. As shown in Fig. 3-c, with SlotGNN, we optimize robot actions to align a scene’s state with a target goal image. This is pivotal when the robot interacts with several objects to reach a

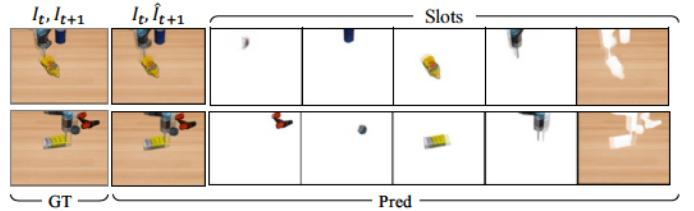


Fig. 5. Examples of single-step dynamics prediction using SlotGNN. Given the current scene image and the robot’s pushing action, our model precisely predicts the future state of each slot and synthesizes the future scene image.

desired state. Given a scene image, we sample possible action sequences over a planning horizon and forecast the slot representation by rolling the dynamics. Using Model-predictive control (MPC) [21], the optimal action sequence is chosen by minimizing the slot loss  $\mathcal{L}_{\text{Slot}}$ , which quantifies the variance between predicted slots and the goal image slots.

## IV. EXPERIMENTS

We structure our experiments around: (1) How accurately and consistently do slots extracted by SlotTransport represent each visual element in the scene? (2) How effective is SlotGNN in predicting multi-object scene dynamics? (3) How well does our framework apply to downstream robotic tasks?

### A. Data

**Simulation:** Using Mujoco [22], [23], we simulate a multi-object tabletop scene with YCB objects [24] and a UR5e robot with a cylindrical end-effector. The robot performs planar pushing action, captured by an RGB camera. The data is formatted as image-action tuples containing pre- and post-action images, and action vectors in the image coordinates. We generate episodes steps of random pushes for a given subset of objects. SlotTransport is trained by randomly sampling target and source images across all episodes. We then use the learned SlotTransport to discover slots and train SlotGNN on the image-action tuples. Evaluations on SlotTransport are done with five objects using images from a top-view camera. Experiments involving single-step dynamics, long-horizon predictions, and object rearrangements are on scenes with three objects with an angled camera.

**Real-world:** We use a UR5e robot with a custom-printed cylindrical end-effector and an RGB camera. We collect data episodes steps of random pushes for subsets of 3 real YCB objects. Models trained in the simulation for the same object subset are retrained on this real-world data.

### B. Baselines

We evaluate our approach against various methods:

**Object Discovery:** Our SlotTransport, is compared with the original slot attention approach [11]. We follow the implementation of this baseline by excluding the transport mechanism introduced in our SlotTransport during training. Furthermore, we compare our approach with the off-the-shelf SAM [20].

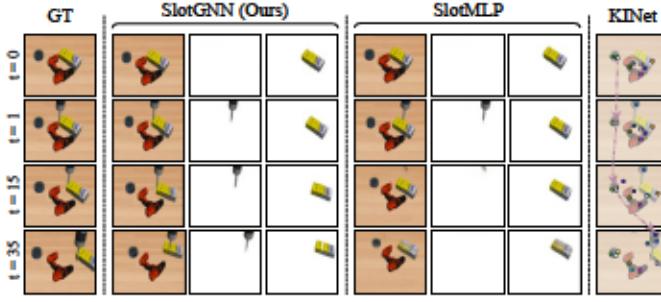


Fig. 6. Long-horizon slot dynamics prediction: SlotGNN has more stability compared to SlotMLP. We also show keypoints detected with KINet [18].

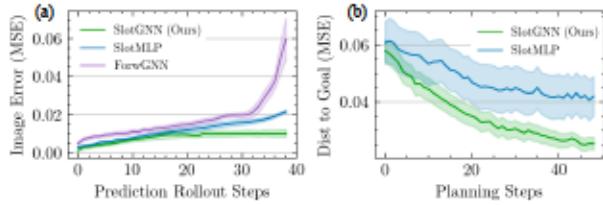


Fig. 7. (a) Long-horizon dynamics rollout error. SlotGNN exhibits robust dynamics predictions as the timestep increases. (b) Planning results: Comparing the distance to the goal image between SlotGNN and SlotMLP.

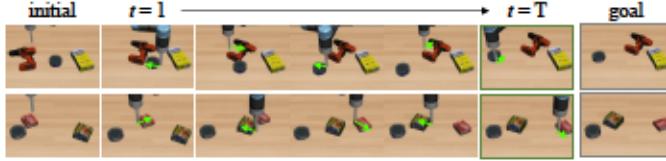


Fig. 8. Qualitative results on control. Each row shows the action sequence (highlighted in green) optimized to maximize scene similarity to goal image.

**Multi-Object Dynamics:** For action-conditioned graph-based dynamics, we consider ForwGNN [6], which uses supervision of ground-truth object masks. The scene graph’s nodes are embedded with the ground-truth object positions and masks to directly reconstruct the future image. We also compare with KINet [18], an unsupervised model that determines dynamics by identifying a set of keypoints from the scene image. Lastly, we compare with SlotMLP variant. While it utilizes slots from SlotTransport, it models dynamics with MLPs rather than the graph-based approach of SlotGNN.

**Evaluation Metrics:** We compute pixel-wise mean squared error (MSE) and Learned Perceptual Image Patch Similarity (LPIPS) [25] to measure the accuracy of the slots in reconstructing the scene composition. Additionally, to quantify the quality and consistency of the slot masks, we compute the mean Intersection over Union (IoU) for slot masks produced by SlotTransport, SlotAttention, and SAM, comparing them against the ground-truth masks from simulation.

## V. RESULTS

### A. Object Discovery Performance

Figure 4 showcases the slot masks and slot reconstructions. In a scene with five objects, SlotTransport qualitatively outper-

TABLE I

Object discovery performance measured as visual quality (MSE and LPIPS) and mask consistency (mIoU) (MSE and LPIPS values are scaled  $\times 10^{-2}$ ).

Method	MSE ↓	LPIPS ↓	mIoU (%) ↑
SlotTransport (Ours)	<b><math>0.17 \pm 0.06</math></b>	<b><math>1.29 \pm 0.46</math></b>	<b><math>93.4 \pm 0.7</math></b>
SlotAttention [11]	$0.43 \pm 0.17$	$4.79 \pm 0.32$	$50.5 \pm 16.2$
SAM [20]	N.A.	N.A.	$86.9 \pm 8.5$

TABLE II

Single-step dynamics prediction accuracy measured as visual quality (MSE) and mask consistency (mIoU).

Method	Supervision	MSE ↓	mIoU (%) ↑
SlotGNN (Ours)	Img	<b><math>0.14 \pm 0.05</math></b>	<b><math>86.9 \pm 2.9</math></b>
SlotMLP	Img	$0.32 \pm 0.09$	$72.6 \pm 1.1$
KINet [18]	Img	$1.86 \pm 0.09$	N.A.
ForwGNN [6]	GT State	$0.50 \pm 0.14$	N.A.

forms the SlotAttention baseline [11]. SlotTransport accurately identifies all distinct visual elements, and predicts an accurate mask for each—even under heavy occlusion. However, as seen in Fig. 4, the SlotAttention baseline overlooks the spam object occluded by the power drill. Moreover, SlotTransport delineates clear boundaries for each slot and accurately reconstructs their appearance. In contrast, the SlotAttention baseline presents indistinct, blurred object masks and reconstructions. We further show that relying on off-the-shelf segmentation methods, such as SAM [20], is not optimal for learning object representations. This is primarily due to SAM’s tendency to over-segment textured objects (e.g., backgrounds) and under-segment cluttered objects.

Table I summarizes the quantitative evaluation of both the visual quality of reconstructed slots and the precision of slot masks. SlotTransport distinctly outperforms the SlotAttention baseline by achieving significantly better visual fidelity, measured in MSE and LPIPS. Furthermore, object masks produced by SlotTransport demonstrate superior alignment with ground-truth masks derived from simulated data. In contrast, SlotAttention often struggles to align slots accurately to cluttered objects, as shown in Fig 4. This limitation is evident in the lower mIoU for SlotAttention compared to SlotTransport.

### B. Dynamics Prediction Performance

Figure 5 illustrates the single-step dynamics prediction of SlotGNN. By taking as input the current image and the intended robot’s pushing action vector, our model accurately predicts the future scene. It does so by predicting the future state of each slot, based on the learned multi-object dynamics of the scene. The quantitative results presented in Table II highlight the accuracy of SlotGNN in single-step dynamics prediction. In single-step dynamics prediction, SlotGNN outperforms all other baselines, including the SlotMLP variant and the unsupervised keypoint dynamics KINet [18]. It’s worth noting that while ForwGNN does rely on ground-truth state information for supervision, it still falls short in MSE com-

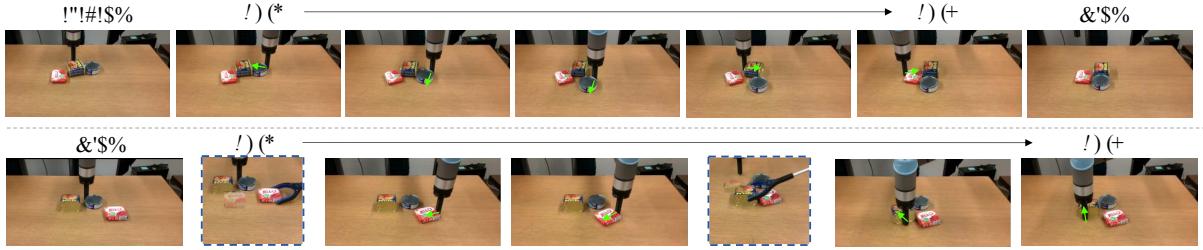


Fig. 9. Real-world control using SlotTransport and SlotGNN: The top row shows objects being rearranged to align with a goal image. In the bottom row, objects are persistently displaced from their goal positions, the robot comes up with a sequence of actions to push the objects back to their desired locations.

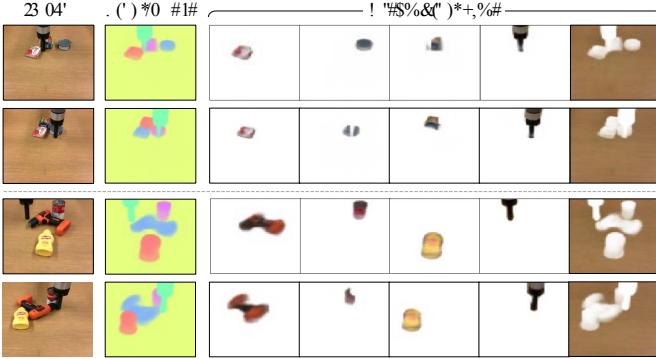


Fig. 10. Real-world object slot discovery with SlotTransport. Our unsupervised framework transfers to real settings and discovers accurate object-centric representations that reflect the positional and visual features of the objects.

pared to SlotGNN, which utilizes image-based supervision. This further highlights the robustness of the detected slots in SlotTransport in representing objects enabling SlotGNN to learn accurate multi-object dynamics.

As illustrated in Fig. 6, SlotGNN excels in predicting stable long-horizon dynamics compared with SlotMLP. Although the scenes reconstructed with SlotGNN may diverge from the ground-truth due to cumulative prediction errors, it yields physically plausible future scenes. In contrast, SlotMLP struggles to retain the coherence slots over time. Given that both SlotGNN and SlotMLP use slots by SlotTransport, the difference in their long-horizon predictions can be attributed to the graph-based model’s enhanced ability to capture multi-object dynamics. In Fig. 6, unsupervised keypoints detected by KINet [18] are also shown. KINet requires stable keypoint-object correspondences to learn multi-object dynamics. This stability is compromised when a robot enters or exits the frame or introduces object occlusions (see the pink keypoint in the last column of Fig. 6). A quantitative summary of the long-horizon rollout outcomes can be found in Fig. 7-a.

### C. Planning with SlotGNN

Fig. 8 shows our method’s application in control tasks. In a challenging object rearrangement scenario, the robot plans an action sequence using SlotTransport and SlotGNN. Through accurate multi-object dynamics projections, the robot effectively aligns objects to a desired configuration using just the

RGB image. The planning performance of slot-based models are compared in Fig. 7 which emphasizes the effectiveness of a graph-based model in learning object-centric dynamics.

### D. Real-World Experiments

Demonstrating the real-world applicability of our unsupervised approach, we successfully transfer SlotTransport and SlotGNN, initially trained in simulation, to the real robot by collecting a minimal dataset of just 20 real robot demonstrations (%5 of the amount of simulated training data). SlotTransport retains its accuracy in the real environment as shown in Fig 10. The slots discovered from the real multi-object scene, clearly distinguish all the scene elements even under occlusion. For the real-world control, we experiment with two tasks as shown in Fig. 9. The first scenario, presented in the top row, involves rearranging objects to achieve a predetermined goal image. The bottom row showcases a more dynamic scenario where objects are continuously displaced from their target positions by a human with a grabber stick. In response, our robot, using SlotTransport and SlotGNN, finds a sequence of actions to restore the objects to their intended locations.

## VI. CONCLUSION

This work addresses the challenges of unsupervised learning for multi-object dynamics through visual observations. We present SlotTransport, a novel approach based on slot attention for unsupervised object discovery, ensuring temporal consistency in object-centric representations. Alongside, we introduce SlotGNN, an unsupervised graph-based dynamics model for predicting the future states of multi-object scenes using the slots. Both methods have proven effective in complex robotic control tasks and long-horizon dynamics prediction. Importantly, we demonstrate that our unsupervised approach, using SlotTransport and SlotGNN, successfully transfers to real-world settings and enables object discovery and dynamic modeling solely from RGB images. For limitations, one key aspect we recognize is that our slot discovery process currently necessitates the pre-determination of the number of slots. In our experiments, we predefined the slot count equal to the anticipated number of elements in the scene. Developing a more adaptive mechanism that automatically determines the required slot count could be a promising future research direction.

## REFERENCES

[1] Z. Pylyshyn, “The role of location indexes in spatial perception: A sketch of the first spatial-index model,” *Cognition*, vol. 32, no. 1, pp. 65–97, 1989.

[2] D. Marr, *Vision: A computational investigation into the human representation and processing of visual information*. MIT press, 2010.

[3] C. Finn and S. Levine, “Deep visual foresight for planning robot motion,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 2786–2793.

[4] P. Agrawal, A. Nair, P. Abbeel, J. Malik, and S. Levine, “Learning to poke by poking: Experiential learning of intuitive physics,” *arXiv preprint arXiv:1606.07419*, 2016.

[5] D. Driess, Z. Huang, Y. Li, R. Tedrake, and M. Toussaint, “Learning multi-object dynamics with compositional neural radiance fields,” in *Conference on Robot Learning*. PMLR, 2023, pp. 1755–1768.

[6] Y. Ye, D. Gandhi, A. Gupta, and S. Tulsiani, “Object-centric forward modeling for model predictive control,” in *Conference on Robot Learning*. PMLR, 2020, pp. 100–109.

[7] H. Qi, X. Wang, D. Pathak, Y. Ma, and J. Malik, “Learning long-term visual dynamics with region proposal interaction networks,” *arXiv preprint arXiv:2008.02265*, 2020.

[8] M. Minderer, C. Sun, R. Villegas, F. Cole, K. Murphy, and H. Lee, “Unsupervised learning of object structure and dynamics from videos,” *arXiv preprint arXiv:1906.07889*, 2019.

[9] P. W. Battaglia, R. Pascanu, M. Lai, D. Rezende, and K. Kavukcuoglu, “Interaction networks for learning about objects, relations and physics,” *arXiv preprint arXiv:1612.00222*, 2016.

[10] A. Sanchez-Gonzalez, N. Heess, J. T. Springenberg, J. Merel, M. Riedmiller, R. Hadsell, and P. Battaglia, “Graph networks as learnable physics engines for inference and control,” in *International Conference on Machine Learning*. PMLR, 2018, pp. 4470–4479.

[11] 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*, vol. 33, pp. 11525–11538, 2020.

[12] T. Kipf, E. Fetaya, K.-C. Wang, M. Welling, and R. Zemel, “Neural relational inference for interacting systems,” in *International Conference on Machine Learning*. PMLR, 2018, pp. 2688–2697.

[13] Y. Li, J. Wu, J.-Y. Zhu, J. B. Tenenbaum, A. Torralba, and R. Tedrake, “Propagation networks for model-based control under partial observation,” in *ICRA*, 2019.

[14] A. Sanchez-Gonzalez, J. Godwin, T. Pfaff, R. Ying, J. Leskovec, and P. Battaglia, “Learning to simulate complex physics with graph networks,” in *International conference on machine learning*. PMLR, 2020, pp. 8459–8468.

[15] 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.

[16] N. Watters, D. Zoran, T. Weber, P. Battaglia, R. Pascanu, and A. Tacchetti, “Visual interaction networks: Learning a physics simulator from video,” *Advances in neural information processing systems*, vol. 30, pp. 4539–4547, 2017.

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

[18] A. Rezazadeh and C. Choi, “Kinet: Unsupervised forward models for robotic pushing manipulation,” *IEEE Robotics and Automation Letters*, 2023.

[19] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, “Neural message passing for quantum chemistry,” in *International conference on machine learning*. PMLR, 2017, pp. 1263–1272.

[20] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, “Segment anything,” *arXiv preprint arXiv:2304.02643*, 2023.

[21] E. Camacho and C. Alba, *Model Predictive Control*, ser. Advanced Textbooks in Control and Signal Processing. Springer London, 2013.

[22] E. Todorov, T. Erez, and Y. Tassa, “Mujoco: A physics engine for model-based control,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 5026–5033.

[23] 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.

[24] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, “The ycb object and model set: Towards common benchmarks for manipulation research,” in *2015 international conference on advanced robotics (ICAR)*. IEEE, 2015, pp. 510–517.

[25] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The unreasonable effectiveness of deep features as a perceptual metric,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595.