

InvSlotGNN: Unsupervised Discovery of Viewpoint Invariant Multiobject Representations and Visual Dynamics

Alireza Rezazadeh¹, Houjian Yu², Karthik Desingh¹, *Member, IEEE*, and Changhyun Choi¹, *Member, IEEE*

Abstract—Learning multiobject dynamics purely from visual data is challenging due to the need for robust object representations that can be learned through robot interactions. In previous work (Rezazadeh et al., 2023), we introduced two novel architectures: SlotTransport for discovering object-centric representations from singleview RGB images, referred to as slots, and SlotGNN for predicting scene dynamics from singleview RGB images and robot interactions using the discovered slots. This article introduces InvSlotGNN, a novel framework for learning multiview slot discovery and dynamics that are invariant to the camera viewpoint. First, we demonstrate that SlotTransport can be trained on multiview data such that a single model discovers temporally aligned, object-centric representations from a wide range of different camera angles. These slots bind to objects from various viewpoints, even under occlusion or absence. Next, we introduce InvSlotGNN, an extension of SlotGNN, that learns multiobject dynamics invariant to the camera angle and predicts the future state from observations taken by uncalibrated cameras. InvSlotGNN learns a graph representation of the scene using the slots from SlotTransport and performs relational and spatial reasoning to predict the future state of the scene for arbitrary viewpoints, conditioned on robot actions. We demonstrate the effectiveness of SlotTransport in learning multiview object-centric features that accurately encode visual and positional information. Furthermore, we highlight the accuracy of InvSlotGNN in downstream robotic tasks, including long-horizon prediction and multiobject rearrangement. Finally, with minimal real data, our framework robustly predicts slots and their dynamics in real-world multiview scenarios.

Index Terms—Multi-object dynamics, object-centric representation learning, visual perception in robotics.

Received 16 June 2024; revised 19 November 2024; accepted 13 January 2025. Date of publication 18 February 2025; date of current version 13 March 2025. This work was supported in part by the UROP Program at the University of Minnesota and the MnDRIVE UMII (University of Minnesota Informatics Institute) Seed Award and in part by the Sony Research Award Program and NSF under Grant 2143730. This article was recommended for publication by Associate Editor R. Detry and Editor J. Bohg upon evaluation of the reviewers' comments. (Karthik Desingh and Changhyun Choi contributed equally to this work.) (Corresponding author: Alireza Rezazadeh.)

Alireza Rezazadeh, Houjian Yu, and Changhyun Choi are with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: rezaz003@umn.edu; yu000487@umn.edu; cchoi@umn.edu).

Karthik Desingh is with the Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: kdesingh@umn.edu).

Our project page: <https://bit.ly/InvSlotGNN>.

This article has supplementary downloadable material available at <https://doi.org/10.1109/TRO.2025.3543274>, provided by the authors.

Digital Object Identifier 10.1109/TRO.2025.3543274

I. INTRODUCTION

STUDIES suggest that the human visual system identifies conceptually distinct visual features, indexes their locations [2], and utilizes this information as the foundation for higher-level cognitive processes, such as comprehending and interacting effectively with the world [3]. A similar principle guides many robotic systems for goal-directed motor planning. In multiobject manipulation, early approaches aimed to directly project the image observation into a unified lower-dimensional space to infer the dynamics [4], [5]. However, such strategies do not reflect the inherent structure of a multiobject 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 observation into object-specific lower-dimensional latent variables and subsequently learning dynamics within these “object-centric” representations [6], [7], [8], [9]. For multiobject systems, recent studies emphasize the effectiveness of learning object-centric representations to enhance the accuracy and sample efficiency of dynamic models [6], [7]. 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 multiobject dynamics heavily rely on ground-truth information, including object pose [7], [10], [11] and segmentation masks [6], [7]. 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 multiobject scenarios and harnessing these representations to understand their dynamics.

In our previous work [1], we introduced *SlotTransport*, an unsupervised method for discovering object-centric representations from singleview RGB images. These discovered slots capture the scene composition, with each slot representing a visual entity in a multiobject scene. We also proposed *SlotGNN*, an unsupervised graph-based model for predicting multiobject scene dynamics from these object-centric slot representations

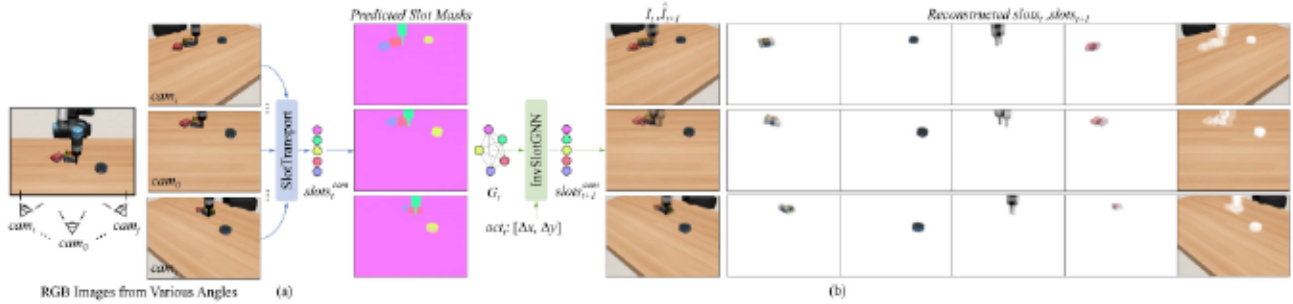


Fig. 1. (a) Using *SlotTransport*, we identify a set of temporally aligned, object-centric slots, each representing a unique visual element from RGB images captured from different camera angles within the scene. (b) We introduce *InvSlotGNN*, a graph-based model that learns and predicts scene dynamics within the slot representation space based on the robot's actions. This learned dynamics model can make accurate predictions for various camera angles.

derived from singleview RGB images. In this model, the scene is transformed into a graph where each node represents a slot, and edges capture the interactions between slots. SlotGNN performs relational reasoning on this graph and learns to predict the future state of each slot. However, one limitation remains unaddressed: how the proposed framework applies to observations from multiple arbitrary camera angles without calibration.

In this evolved work, we present an in-depth study of viewpoint invariant object-centric representation and scene dynamics prediction in robot manipulation and strive to carry on the unsupervised robot learning scheme without the laborious human annotations. With observations from uncalibrated RGB cameras in different angles, our robot system is able to effectively and efficiently learn the temporal consistent object representations and predict the scene changes conditioned on robot motion (see Fig. 1). We propose three core technical contributions as follows.

- 1) *Viewpoint invariant object discovery with SlotTransport*: We first demonstrate that our proposed SlotTransport can be trained to discover temporally aligned, object-centric representations from a wide range of different camera angles within the scene. The learned multiview slots robustly bind to distinct objects, ensuring consistent object representation across various viewpoints.
- 2) *Viewpoint invariant scene dynamics InvSlotGNN*: We design *InvSlotGNN*, a novel framework that addresses the challenge of learning multiview slot dynamics invariant to camera viewpoints.
- 3) *Evaluation and analysis of multiview object representation and dynamics*: We evaluate the multiview object discovery and scene dynamics prediction extensively in simulation and real robot setup, which verifies the model effectiveness.

These extensions significantly advance the original framework by enabling the unsupervised learning of multiview object representations and dynamics. Importantly, *InvSlotGNN* extends SlotGNN by learning a graph representation of the scene that is invariant to the camera angle, allowing for accurate prediction of future states of multiobject scenes from uncalibrated camera observations. This capability is critical for real-world applications, where camera calibration is challenging or viewpoints can vary unpredictably.

II. RELATED WORK

Learning Multiobject Dynamics Model: Early models for graph-based dynamics, such as Interaction Networks (IN) [10], [11] and follow-up adaptations [12], [13], [14], represent a multiobject system with a graph where each node is an object ground-truth state (e.g., position, velocity, mass, and 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 [6], [7], [8], [15], [16], [17], [18]. 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 multiobject 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 [19]. 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 [20]. In recent work on unsupervised multiobject dynamics, keypoint extraction has been explored [21]. 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.

Learning Multiview Representation and Dynamics: Recent advances in neural scene representations have significantly contributed to learning multiview representations and dynamics. For multiview learning, it is more practical to not rely on camera calibration, as obtaining precise pose information can be challenging or infeasible in real-world scenarios. RUST [22] introduces a pose-free approach to novel view synthesis trained on RGB

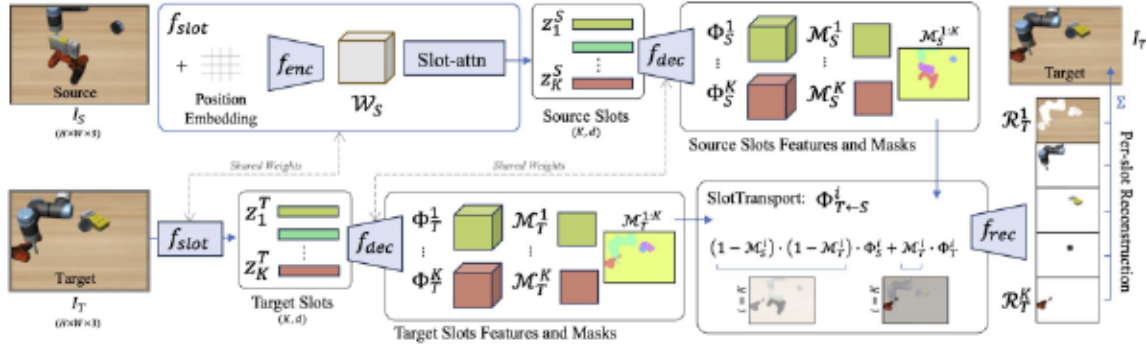


Fig. 2. *SlotTransport*: Unsupervised Multiobject Discovery. From an RGB image, f_{slot} identifies object-centric slots $z_{1:K}$. Through slot attention [19], slots bind to visual features, and f_{dec} produces feature maps Φ^i and masks $\mathcal{M}^i$. Temporal alignment is ensured by transporting 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.

images. Unlike prior methods requiring precise pose information [23], [24], [25], RUST utilizes a pose estimator module that infers a latent pose embedding from the target image. This model successfully matches the performance of methods with perfect pose data. However, RUST is limited to static scenes, which restricts its application in learning dynamics that require action-based future state predictions. More closely related to our approach is DyST [26]. DyST learns latent decompositions of videos from an uncalibrated camera into scene content, per-view scene dynamics, and camera pose for view generation by co-training on synthetic and real-world data. However, DyST lacks object-level predictions and is not action-conditioned, limiting its ability to predict future states resulting from robot actions, making it more suited for video prediction. In contrast, our InvSlotGNN makes future predictions at the object level and is conditioned on the robot's actions, enabling accurate predictions in response to robotic interactions.

III. METHODS

1) *SlotTransport*: An unsupervised multiobject discovery model that efficiently extracts robust and temporally consistent object-centric representation slots from multiobject scenes. We demonstrate that SlotTransport is able to learn object-centric representations from singleview and multiview observations.

2) *SlotGNN*: This unsupervised graph-based model learns the dynamics of the object-centric representations discovered through SlotTransport. Importantly, SlotGNN is conditioned on the robot's action, which enables applications such as model-based action planning.

3) *InvSlotGNN*: To enable learning dynamics from multiview observations, we build on top of SlotGNN. InvSlotGNN handles varying camera viewpoints, allowing the model to learn dynamics that is invariant to the camera angle. InvSlotGNN predicts the future states of multiobject scenes from uncalibrated camera observations. This capability is crucial for applications, where camera calibration is challenging or the viewpoints can vary unpredictably.

A. SlotTransport: Unsupervised Multiobject 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 [19] 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 $\mathcal{W} \in \mathbb{R}^{h \times w \times e}$. Using the slot attention [19], 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 [20], 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 $\Phi_{T \leftarrow S}^i$ by nullifying the source feature map outside the slot's predicted mask for both the target and source $(1 - \mathcal{M}_T^i) \cdot (1 - \mathcal{M}_S^i) \cdot \Phi_S^i$, followed by overlaying the masked target feature map $\mathcal{M}_T^i \cdot \Phi_T^i$. Finally, a convolutional reconstruction module f_{rec} reconstructs each slot as an RGB image $\mathcal{R}_T^i \in \mathbb{R}^{h \times w \times 3}$ based on the transported feature map. The reconstructed slots together reconstructed 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

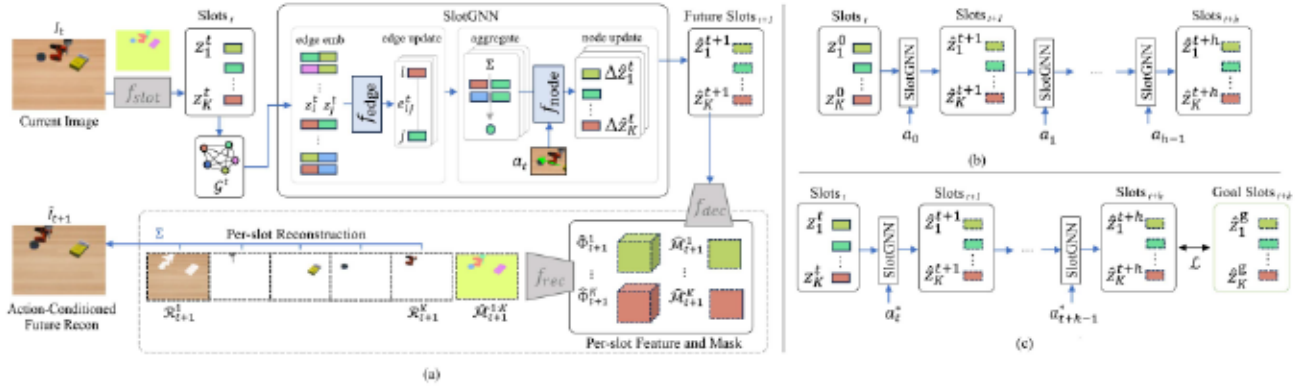


Fig. 3. *SlotGNN: Unsupervised Multiobject Dynamics.* (a) SlotGNN predicts slot changes Δz_i^t after applying a pushing action a_t . Using SlotTransport's slots (f_{slot}), a graph G_t is formed with slots as nodes and slot interactions as edges. Edges and nodes are updated via f_{edge} and f_{node} , resulting in next timestep slots z_i^{t+1} . The next image I_{t+1} is then reconstructed. (b) With a sequence of robot actions, SlotGNN projects future multiobject dynamics and synthesizes future scenes. (c) SlotGNN also facilitates goal-directed planning to optimize actions toward a desired goal.

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 Multiobject Dynamics

The main purpose of SlotGNN is to learn the dynamics and model interactions between the visual elements in a multiobject 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 multiobject 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 $G_t = (\mathcal{V}, \mathcal{E})$. 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 [11], [27] 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 Fig. 3). First, the edge embeddings, e_{ij} , are updated based on their connecting node embeddings: $e'_{ij} \leftarrow f_{\text{edge}}(e_{ij}, n_i, n_j)$. Second, the node embeddings are updated using the updated edge embeddings associated with them and the robot action: $n'_k \leftarrow f_{\text{node}}(n_k, \sum_{i \in \mathcal{N}(k)} e'_{ik}, a_t)$. Here, f_{edge} and f_{node} are multilayer perceptron 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 $a_t = [(u, v)_{\text{start}}, (u, v)_{\text{end}}] \in \mathbb{R}^4$, characterized as a point-to-point push vector projected on 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.

After message-passing, the updated node embeddings are used to predict the evolution of slots Δz_i^{t+1} in the next timestep conditioned on the action a_t . This allows for the rollout of the dynamics into the future and enables synthesizing the future appearance of the scene.

C. Viewpoint Invariance With Uncalibrated Camera

In this section, we explore how object discovery using SlotTransport and dynamics modeling with SlotGNN can be applied to multiview observations. Our goal is to train a single model to extract unsupervised object-centric representations from multiview observations and develop a single dynamics model that is invariant to camera viewpoints. First, we discuss how SlotTransport can be trained on a multiview observation dataset to extract robust object-centric representations from various camera angles within a scene. Next, we extend SlotGNN and propose InvSlotGNN, which learns the dynamics of multiobject scenes from arbitrary camera viewpoints without requiring a calibrated camera.

1) *SlotTransport for Multiview Observations:* SlotTransport learns to bind slots to distinct visual features in an image. When trained on a singleview dataset, the slot that binds to the background (e.g., tabletop) represents a static view of its appearance. To extend this to multiview observations, we leverage the ability of slots to represent various poses of objects and robots. The key intuition behind SlotTransport's capability to handle multiview variations lies in its design: by abstracting visual features into distinct slots, the model can adapt to different perspectives without requiring explicit pose information. Similarly, the background slot can represent the tabletop from different poses or camera viewpoints. Since SlotTransport only takes

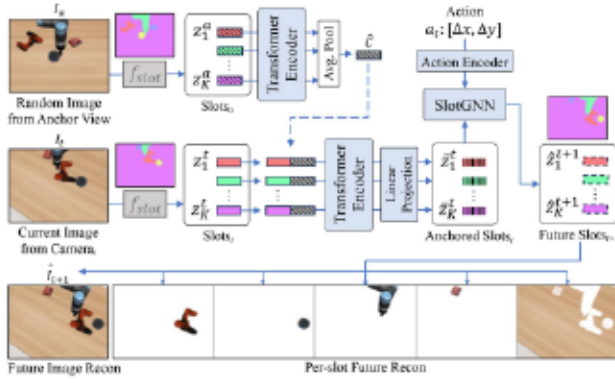


Fig. 4. *InvSlotGNN*: Unsupervised Viewpoint-Invariant Multiobject Dynamics. *InvSlotGNN* predicts future slots $\hat{z}_{1:K}^{t+1}$ after a pushing action a_t is applied, for arbitrary uncalibrated camera angles. From an anchor view image I_a , an anchoring latent $\hat{c}$ is derived and concatenated with the slots detected from the current viewpoint $\hat{z}_{1:K}^t$. A transformer encoder then produces an anchored slot representation $\hat{z}_{1:K}^t$. SlotGNN is subsequently employed to predict the future state of each slot, conditioned on the external action, which can be used to reconstruct the future appearance of the scene $\hat{I}_{t+1}$.

RGB images as input and does not depend on any camera pose, the SlotTransport architecture (see Fig. 2) can process images captured from various camera viewpoints in the environment. Experiments in Section V-A show the capability of SlotTransport in a multiview setting.

2) *InvSlotGNN for Viewpoint-Invariant Dynamics*: To learn dynamics from RGB observations captured by multiview uncalibrated cameras with *InvSlotGNN*, we introduce two major modifications to SlotGNN.

First, to learn viewpoint invariance scene dynamics with uncalibrated cameras, the *InvSlotGNN* is required to comprehend the relationship among different views. We propose anchoring the slot representation obtained from arbitrary camera angles to a fixed camera viewpoint, referred to as the anchor view (see Fig. 4). The idea is to extract slots from observations taken from any viewpoint, alongside a random image from the anchor view, and incorporate both into the dynamics learning process. This approach implicitly accounts for the effect of camera viewpoint on predicting the future state of each slot.

Next, we specify the robot's action directly in world coordinates as planar displacement $a_t = [\Delta x, \Delta y] \in \mathbb{R}^2$ with a length ranging from 1 cm to 3 cm. This differs from SlotGNN's singleview dynamics, where actions were represented as start and end points in image coordinates. This is because learning how robot action from one image coordinate is projected to other uncalibrated camera views while predicting the scene dynamics with constantly changing camera angle is difficult and inefficient (see ablation study in Fig. 11). On the other hand, the world coordinate action representation invariant to viewpoint changes can be learned simultaneously from an anchor view as well as uncalibrated views. Therefore, it achieves better training stability and testing performance. We argue that this delta action representation in world coordinates is crucial for achieving viewpoint invariance in learning dynamics, as representing actions in image coordinates necessitates camera calibration.

More formally, we collect a dataset consisting of image-action tuples (I_t, a_t, I_{t+1}) of different scenes with multiple camera angles including the anchor view, which records scene changes affected by the robot action (more details discussed in Section IV-A). Given the current RGB image observation I_t from an arbitrary uncalibrated camera angle, we extract slots $z_{1:K}^t$ using SlotTransport trained on multiview observations. We also set a fixed anchor view camera angle and randomly select one image of the same scene under the fixed anchor view from the dataset I_a , from which we obtain anchor slots $z_{1:K}^a$. Please note that the selected anchor view image I_a does not require to be synchronized with I_t . We map the anchor slots down to a single anchoring latent representation $\hat{c} \in \mathbb{R}^d$ through a Transformer encoder and applying average pooling. The Transformer encoder is used here to capture relevant information from the anchoring view that can signal the effect of the camera viewpoint. We then augment the anchoring latent with the current observation's slots through concatenation $[z_{1:K}^t, \hat{c}]$. The augmented slot representations contain the current state's information as well as an implicit signal of the camera pose relative to the anchor view.

In order to combine the anchor embedding with the state information, we process the augmented slot representations through a Transformer encoder and project the outputs to obtain anchored slot representations for the current observation $\hat{z}_{1:K}^t$. The Transformer encoder is used again to effectively merge the anchor embedding with the state information, leveraging its capability to handle complex dependencies and global context. The pushing action, provided as displacement a_t , is also passed through an MLP-based action encoder. This set of anchored slots, along with the encoded action, is input to SlotGNN. SlotGNN learns a graph representation of the slots and incorporates the action input in the message passing, as described in Section III-B, to estimate the future state of each slot $\hat{z}_{1:K}^{t+1}$. Through the SlotTransport decoder, we can also obtain a per-slot future image reconstruction and combine them to synthesize a future image of the scene conditioned on the robot's action.

Note that *InvSlotGNN* does not rely on camera pose information directly. It accounts for the effect of camera viewpoint on dynamics by anchoring the slot representations to the anchor view, which implicitly incorporates information regarding the camera pose.

D. Training SlotTransport, SlotGNN, and InvSlotGNN

SlotTransport is trained using only the image reconstruction error for supervision. The image reconstruction loss $\mathcal{L}_{\text{rec}}(I_T, \hat{I}_T)$, 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 or *InvSlotGNN* to learn visual dynamics.

We use a slot prediction MSE loss to train SlotGNN or *InvSlotGNN*, $\mathcal{L}_{\text{slot}}(z_{1:K}^{t+1}, \hat{z}_{1:K}^{t+1})$. This loss reduces the distance between the slots directly predicted from the next timestep image using SlotTransport $z_{1:K}^{t+1}$ and slots from the single-step dynamics prediction $\hat{z}_{1:K}^{t+1}$, as visualized in Figs. 3(a) and 4 (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 $\mathcal{L}_{\text{rec}}(I_{t+1}, \hat{I}_{t+1})$.

E. Long-Horizon Multiobject Dynamics Rollout

Given a sequence of robot pushing actions $a_{0:h}$, a learned SlotGNN or InvSlotGNN rolls out the dynamics of a multiobject 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, f_{rec} can synthesize an image of the scene.

F. Goal-Directed Planning

Learning the scene dynamics facilitates goal-directed sequential action planning for multiobject manipulation. As shown in Fig. 3(c), with SlotGNN or InvSlotGNN, 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 desired state. Given a scene image, we sample possible action sequences over a planning horizon $h \geq 1$ and forecast the slot representation $\hat{z}_{1:K}^{t+h}$ by rolling the dynamics. Using model-predictive control (MPC) [28], the optimal action sequence is chosen by minimizing the slot loss $\mathcal{L}_{\text{slot}}(z_{1:K}^T, \hat{z}_{1:K}^G)$, which quantifies the variance between predicted slots and the goal image slots.

G. Implementation Details

For SlotTransport, input images are 256×256 pixels, and the slot embedding size is set to 256. The visual encoder f_{enc} and decoder f_{dec} are four-layer convolutional networks, each with a 64-dimensional channel size. Batch normalization is applied after each convolutional layer. The reconstruction module f_{rec} is a two-layer convolutional operator. In SlotGNN, the f_{edge} and f_{node} functions are one-layer MLPs with a hidden dimension of 256 and ReLU activations. For InvSlotGNN, both Transformer encoders consist of 3 layers with 8 heads and use a hidden dimension of 256. The action encoder is a one-layer MLP with an output dimension of 256 and ReLU activations.

IV. EXPERIMENTS

For both singleview and multiview observations, we structure our experiments around three primary questions: 1) How accurately and consistently do slots extracted by SlotTransport represent each visual element in the scene? 2) How effective is SlotGNN and InvSlotGNN in predicting multiobject scene dynamics? 3) How well does our framework apply to downstream robotic tasks?

A. Data

Simulation: Using Mujoco [29], [30], we simulate a multiobject tabletop scene with YCB objects [31] and a robot with a cylindrical end-effector. We simulate datasets using UR5e

and Franka Emika Panda robots. The robot performs planar pushing actions, captured by an RGB camera. Please note that such actions result in any combination of rotation/translation of the objects presented in the scene. This dataset further helps us verify whether our InvSlotGNN pipeline has both rotation and translation awareness. The data is formatted as image-action tuples (I_t, a_t, I_{t+1}) , containing pre- and postaction images from different camera views and action vectors in the world coordinates. For singleview experiments, we generate approximately 750 episodes with 20 steps of random pushes for a given subset of objects. SlotTransport is trained by randomly sampling target and source images across all episodes. For multiview experiments, we generate approximately 1400 episodes with 15 interactions each. In each episode, we randomly select 10 camera angles from a set of 250 cameras, pointed toward the center of the tabletop from randomized heights and angles, with a fixed radius of 80 cm from the center of the table. We then use the learned SlotTransport to discover slots and train SlotGNN or InvSlotGNN on the image-action tuples.

Real-World: For singleview experiments, we use a UR5e robot with a custom-printed cylindrical end-effector and an RGB camera. We collect data comprising approximately 20 episodes with 40 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. For multiview experiments, we use Franka Emika Panda and randomly select three different camera angles distributed to collect images from the front, left, and right angles of the tabletop. We collect approximately 5 episodes with 20 steps of random pushes for subsets of 3 real YCB objects for each of the camera angles.

B. Baselines

We evaluate our approach against various methods:

Object Discovery: For singleview experiments, we compare our SlotTransport with the original slot attention approach [19]. For this baseline, we follow the implementation by excluding the transport mechanism introduced in our SlotTransport during training. In addition, we compare our approach with the off-the-shelf SAM [32]. In multiview experiments, we conduct extensive comparisons by training SlotTransport on both singleview and multiview observations. We also test the model's performance on unseen camera angles.

Multi-Object Dynamics: For singleview experiments, in the context of action-conditioned graph-based dynamics, we consider ForwGNN [7], 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 [21], an unsupervised model that determines dynamics by identifying a set of keypoints from the scene image. Last, we compare with a SlotMLP variant, which utilizes slots from SlotTransport but models dynamics with MLPs rather than the graph-based approach of SlotGNN. For multiview experiments, we compare the performance of InvSlotGNN with SlotGNN, training each on both singleview and multiview setups for a thorough comparison.

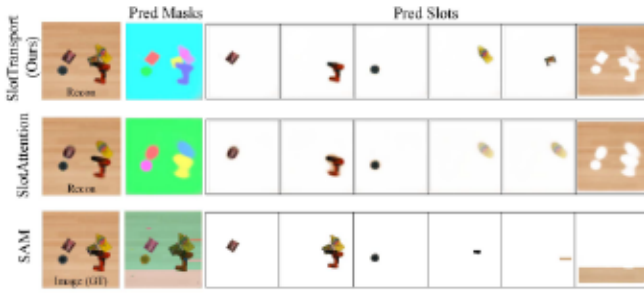


Fig. 5. Visualizations of singleview per-slot masks and reconstructions. SlotTransport exhibits superior performance in accuracy and consistency of object-centric representation, even under occlusion, compared to the SlotAttention baseline [19]. We also showcase predicted segments from SAM [32].

TABLE I
SLOTTRANSPORT SINGLEVIEW OBJECT DISCOVERY PERFORMANCE MEASURED AS VISUAL QUALITY (MSE AND LPIPS $\times 10^{-2}$) AND MASK CONSISTENCY (mIoU)

Method	MSE $\downarrow$		LPIPS $\downarrow$		mIoU (%) $\uparrow$	
SlotTransport (Ours)	0.17	0.06	1.29	0.46	93.4	0.7
SlotAttention [19]	0.43	0.17	4.79	0.32	50.5	16.2
SAM [32]	N.A.	N.A.	N.A.	N.A.	86.9	8.5

Evaluation Metrics: We compute pixel-wise mean squared error (MSE) and Learned Perceptual Image Patch Similarity (LPIPS) [33] 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

Fig. 5 showcases the slot masks and slot reconstructions for singleview observations. In a scene with five objects, SlotTransport qualitatively outperforms the SlotAttention baseline [19]. SlotTransport accurately identifies all distinct visual elements, and predicts an accurate mask for each—even under heavy occlusion. However, as seen in Fig. 5, 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 [32], 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 in singleview observations. 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

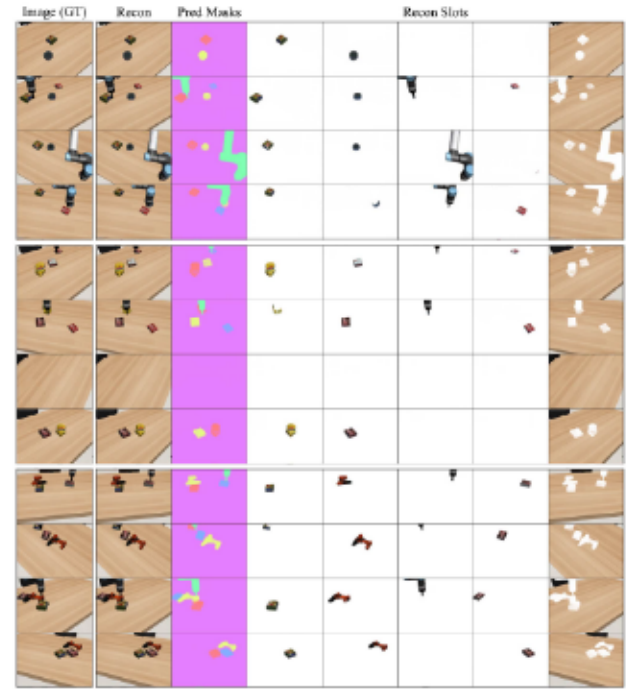


Fig. 6. Visualizations of SlotTransport performance on multiview observations. SlotTransport consistently detects accurate object-centric representations from arbitrary camera angles.

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. 5. This limitation is evident in the lower mIoU for SlotAttention compared to SlotTransport.

In singleview observations, the slot representing the background remains static as the background appearance does not change under a single camera viewpoint. However, similar to how slots that bind to objects and the robot can represent them in various poses, the background slot can also adapt to represent the tabletop’s appearance from different camera viewpoints. As shown in Fig. 6, SlotTransport trained on multiview observations robustly discovers temporally consistent slot representations. Importantly, the corresponding slots capture the viewpoint effect on the appearance of the robot and background. In addition, Fig. 6 demonstrates SlotTransport’s capability to handle high occlusion, object absence, and challenging viewpoints effectively. Therefore, SlotTransport is not limited to singleview observations and can be trained on multiview observations to discover object-centric representations that are intuitively more 3-D aware.

Fig. 7 summarizes a thorough quantitative evaluation of SlotTransport trained and tested on singleview and multiview observations. The image reconstruction quality measures how accurately slots capture the positional and visual features of the elements in the scene. On a given singleview observation test set, SlotTransport trained on multiview observations achieves a significantly lower reconstruction error compared to SlotTransport trained solely on singleview observations.

We also report the reconstruction loss of foreground slots (objects and robot) and the background slot (tabletop). When

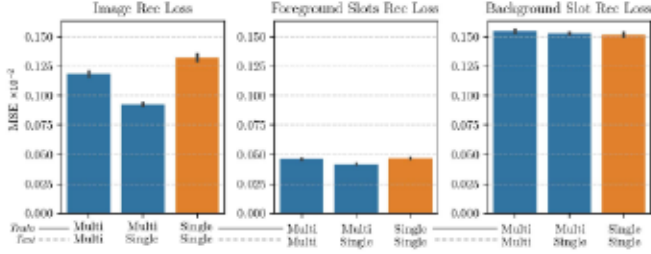


Fig. 7. Ablation study on SlotTransport object discovery performance. The performance is measured as image reconstruction loss (MSE) during training and testing on multiview and singleview observations. Foreground slots refer to those assigned to objects and the robot, while the background slot refers to the tabletop.

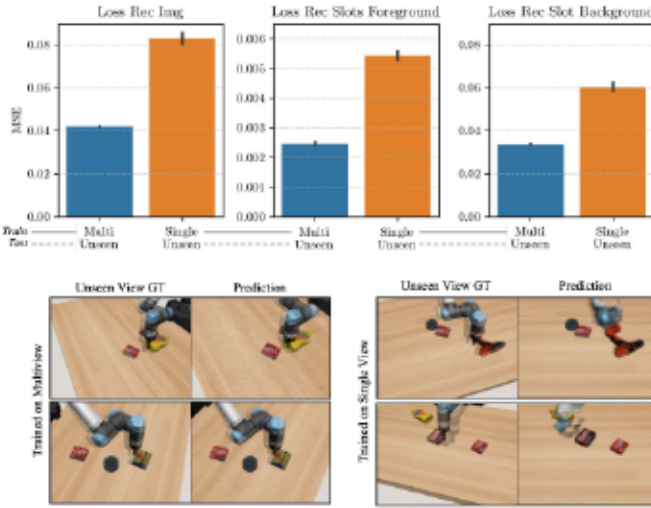


Fig. 8. Comparison of SlotTransport's performance on unseen camera angles. The baseline methods are trained on multiview and singleview observations. Foreground slots refer to those assigned to objects and the robot, while the background slot refers to the tabletop.

tested on the singleview dataset, the accuracy of SlotTransport trained on the multiview dataset is slightly better than that of SlotTransport trained on the singleview dataset. We hypothesize that this is due to training on multiview datasets enhancing the learned representation, as changing the viewpoint helps the model distinguish objects better and learn a more 3D-aware representation. For the background slot, the accuracy of SlotTransport trained on singleview data is slightly better than that of SlotTransport trained on multiview data, although the difference is statistically insignificant. We believe this is because reconstructing the background from multiple angles is a more complex task than reconstructing the background only from a static appearance.

To test whether the slots learned through multiview observations can generalize to unseen angles, we randomly sample out-of-distribution camera poses and evaluate the reconstruction accuracy of these unseen viewpoints after training SlotTransport on singleview and multiview observations. As shown in Fig. 8, SlotTransport trained on the multiview dataset is significantly more accurate in discovering object-centric representations from

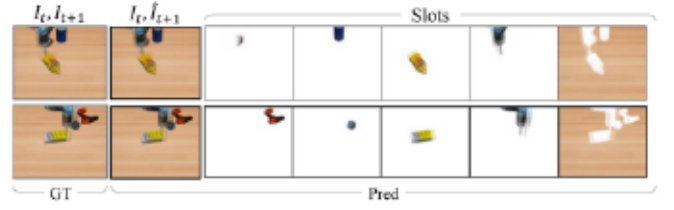


Fig. 9. Examples of SlotGNN single-step dynamics prediction using SlotGNN. Given the current scene image and the robot's pushing action, our model predicts the future state of each slot and synthesizes the future image.

TABLE II
SINGLEVIEW DYNAMICS ACCURACY OF SINGLE-STEP PREDICTIONS MEASURED AS VISUAL QUALITY (MSE $\times 10^{-1}$) AND MASK CONSISTENCY (mIoU)

Method	Supervision	MSE $\downarrow$	mIoU (%) $\uparrow$
SlotGNN (Ours)	Img	0.14 0.05	86.9 2.9
SlotMLP	Img	0.32 0.09	72.6 1.1
KINet [21]	Img	1.86 0.09	N.A.
ForwGNN [7]	GT State	0.50 0.14	N.A.

unseen camera angles. This improvement is evident in both the foreground (objects and robot) and background (tabletop) slots.

We include examples of SlotTransport's performance on unseen angles in Fig. 8, showing the ground-truth image from the unseen viewpoint and the image reconstructed by SlotTransport. When trained on singleview observations, the model fails to accurately reconstruct the robot appearance and the tabletop in the unseen pose, often inpainting the singleview appearance of the tabletop instead. In contrast, when trained on the multiview dataset, the model captures the unseen appearance of the tabletop, although with inaccuracies. More importantly, the foreground slots (objects and robot) are more accurately represented by the model. These results demonstrate that SlotTransport is not only capable of learning robust object-centric representations from multiview observations, but also generalizes these representations better to novel viewpoints.

B. Dynamics Prediction Performance

Fig. 9 illustrates the single-step dynamics prediction of SlotGNN in singleview observations. By taking as input the current image and the intended robot's pushing action vector in image coordinate, our model accurately predicts the future scene. It does so by predicting the future state of each slot, based on the learned multiobject dynamics of the scene. The quantitative results presented in Table II highlight the accuracy of SlotGNN in single-step dynamics prediction in singleview observations. In single-step dynamics prediction, SlotGNN outperforms all other baselines, including the SlotMLP variant and the unsupervised keypoint dynamics KINet [21]. It is worth noting that while ForwGNN does rely on ground-truth state information for supervision, it still falls short in MSE compared 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 multiobject dynamics.



Fig. 10. Examples of single-step dynamics prediction using InvSlotGNN. Given the current scene image from an arbitrary uncalibrated camera angle, our model accurately predicts the future state of each slot based on the robot's pushing action in the world coordinate, and synthesizes the future scene image.

For multiview observations, Fig. 10 illustrates examples of single-step dynamics prediction using InvSlotGNN. The current RGB image and the robot's displacement in world coordinates are inputs to InvSlotGNN, which predicts the future state of the scene from arbitrary camera viewpoints. These examples demonstrate that by anchoring the slot representations to the anchor view, InvSlotGNN effectively accounts for the impact of viewpoint on dynamics prediction. Moreover, InvSlotGNN achieves the translation and rotation awareness for both symmetric and asymmetric objects. In Fig. 10, the poses of the power drill align with the ground-truth in arbitrary camera views. In addition, with its graph-based dynamics, InvSlotGNN accurately captures robot-to-object and object-to-object interactions in this multiobject scene. We also thoroughly quantify and compare the performance of SlotGNN and InvSlotGNN trained on multiview and singleview datasets (summarized in Fig. 11).

First, we justify the additional components introduced in InvSlotGNN. Specifically, we implemented an anchoring scheme where slots from arbitrary viewpoints are anchored to slots from a fixed anchor viewpoint. This approach helps the model implicitly account for the viewpoint effect on dynamics without relying on ground-truth camera pose information. For a fair comparison, we modified SlotGNN to take delta actions in world coordinates (SlotGNN-w), similar to InvSlotGNN. By comparing SlotGNN-w and InvSlotGNN models trained and

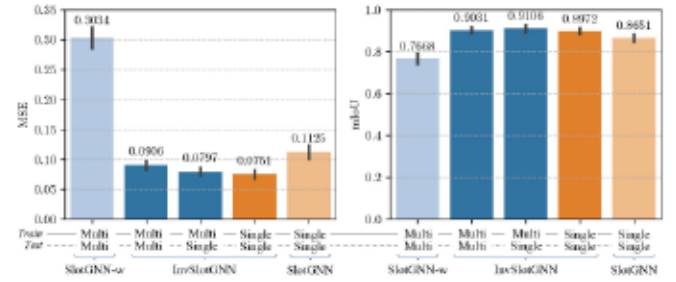


Fig. 11. InvSlotGNN single-step dynamics prediction accuracy: measured as visual quality (MSE $\times 10^{-1}$) and mask consistency (mIoU) during training and testing on multiview and singleview observations.

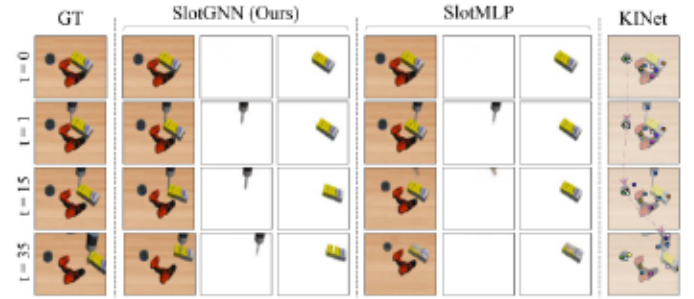


Fig. 12. Long-horizon slot dynamics prediction leverages the graph-based model: SlotGNN has more stability compared to SlotMLP. We also show keypoints detected with KINet [21].

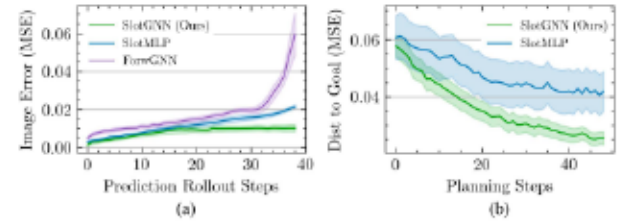


Fig. 13. (a) Long-horizon dynamics rollout error. SlotGNN exhibits robust dynamics predictions as the timestep increases. (b) Planning results: Comparing the distance to the goal image between SlotGNN and SlotMLP.

tested on multiview observations, we found that the addition of the anchoring scheme significantly improved the accuracy of dynamics prediction, both in terms of visual quality (MSE) and mask accuracy (mIoU) (see Fig. 11). This highlights the effectiveness of InvSlotGNN in accurately predicting multiobject scene dynamics from various camera viewpoints, achieving viewpoint-invariant dynamics learning.

Next, we highlight the effectiveness of training on multiview observations by comparing the performance of InvSlotGNN on a given singleview test set after training on both multiview and singleview observations (see Fig. 11). When trained on the multiview dataset, InvSlotGNN shows higher accuracy in predicting the future state of the scene, as measured by mask accuracy (mIoU). However, there is a slight decrease in performance when measured by reconstruction loss (MSE). We argue that this decrease is due to the need to balance performance across a wide range of viewpoints when learning from multiview observations. While this can lead to slightly higher reconstruction loss in pixel

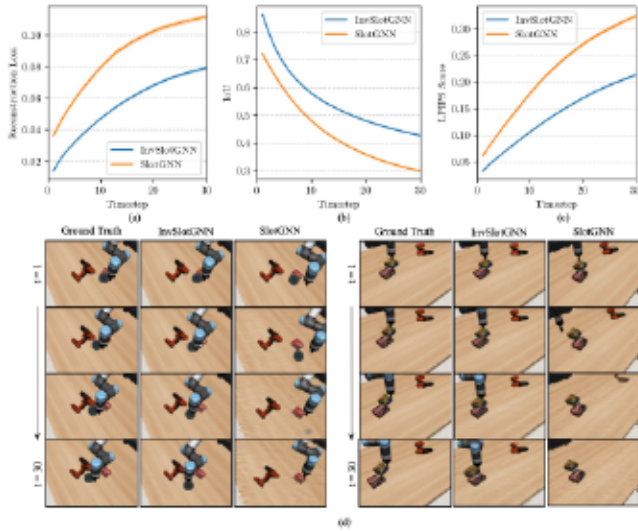


Fig. 14. Long-horizon dynamics prediction leverages the viewpoint-invariant learning in the slot representation space. InvSlotGNN exhibits more stable future rollouts for multiview observations compared to SlotGNN.



Fig. 15. Qualitative results on control. Each row shows the action sequence (highlighted in green) optimized to maximize scene similarity to goal image.

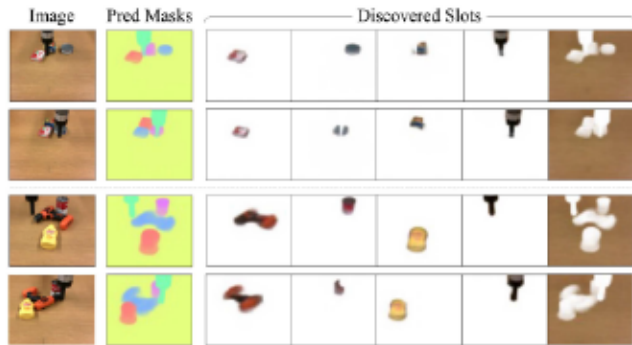


Fig. 16. SlotTransport singleview: Real-world singleview object slot discovery with SlotTransport.

comparisons, the dynamics prediction is indeed more accurate, as indicated by the improved mask accuracy.

Finally, to highlight the effectiveness of representing the action in world coordinates, we compare the performance of InvSlotGNN trained on singleview observations with the robot's action input as delta displacement in world coordinates to SlotGNN trained on singleview observations with the action input as a push vector projected in image coordinates (see Fig. 11). In this comparison, InvSlotGNN is significantly more accurate in both reconstruction quality and mask accuracy. We believe this improvement is due to the precision of representing actions as displacement vectors in world coordinates rather than push

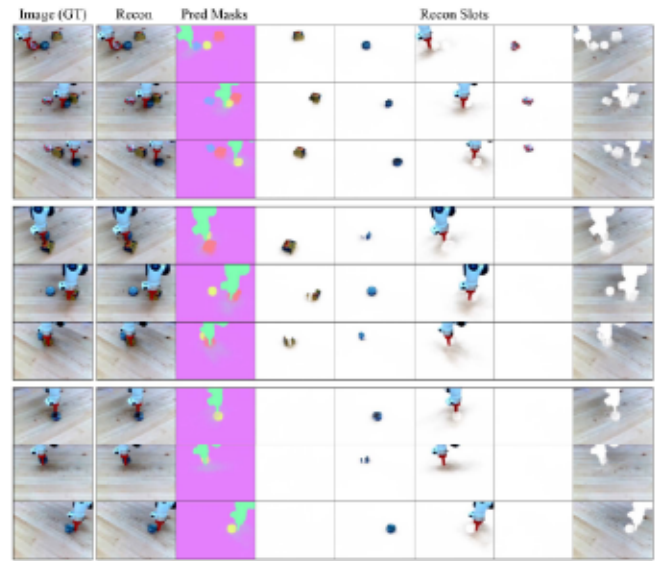


Fig. 17. SlotTransport multiview: Real-world multiview object slot discovery with SlotTransport from various camera angles even under high occlusion or absence of objects.

vectors in pixel coordinates. When actions are projected to pixel coordinates, slight changes in action cannot be accurately reflected due to the granularity of pixels, whereas world coordinate actions provide a more precise representation. In addition, training on delta displacement is more flexible than using absolute start and end points for the pushing action, leading to better performance in dynamic prediction tasks. These findings justify our choice of action representation in learning viewpoint-invariant dynamics.

C. Long-Horizon Dynamics Rollouts

As illustrated in Fig. 12, 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 coherent slots over time. Given that both SlotGNN and SlotMLP use slots from SlotTransport, the difference in their long-horizon predictions can be attributed to the graph-based model's enhanced ability to capture multiobject dynamics. Fig. 12 also shows unsupervised keypoints detected by KINet [21]. KINet requires stable keypoint-object correspondences to learn multiobject dynamics, but 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. 12). A quantitative summary of the long-horizon rollout outcomes can be found in Fig. 13(a).

In addition, we compare InvSlotGNN's long-horizon rollout for multiview observations with SlotGNN. Specifically, we use SlotTransport trained on a multiview dataset to extract slots for varying camera viewpoints and use SlotGNN and InvSlotGNN to roll out the dynamics in the slot representation space. As shown in Fig. 14, InvSlotGNN learns a more accurate and robust dynamics model, retaining visual consistency and dynamics

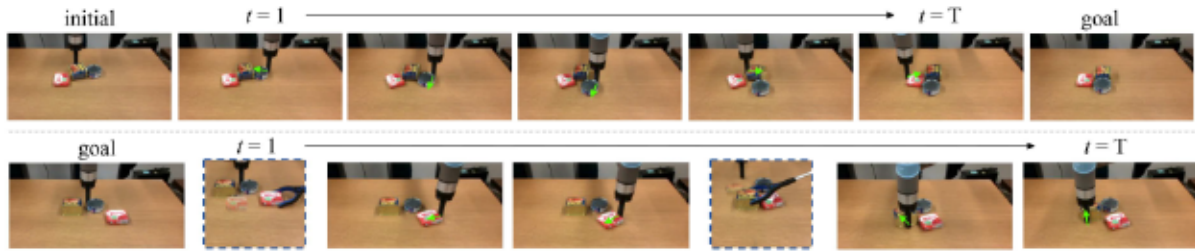


Fig. 18. *Real-world control using SlotTransport and SlotGNN*: The top row shows objects being rearranged to align with a goal image. In the bottom row, objects are persistently displaced from their goal positions, the robot comes up with a sequence of actions to push the objects back to their desired locations.

accuracy after multiple timesteps of rolling out the predictions into the future. These results demonstrate that the graph-based model of SlotGNN provides significant advantages in maintaining coherent multiobject dynamics over extended timeframes. InvSlotGNN further enhances this capability by accurately accounting for viewpoint variations, leading to more reliable and consistent long-horizon predictions in dynamic scenes.

D. Action Planning

Fig. 15 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 multiobject 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. 13(b) which emphasizes the effectiveness of a graph-based model in learning object-centric dynamics.

E. 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). In singleview observations, SlotTransport retains its accuracy in the real environment as shown in Fig. 16. The slots discovered from the real multiobject scene, clearly distinguish all the scene elements even under occlusion. For the real-world control, we experiment with two tasks as shown in Fig. 18. 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.

For multiview observations, SlotTransport consistently discovers object-centric features that bind to distinct objects across the three viewpoints tested in the real setup (see Fig. 17). Importantly, SlotTransport performs well under conditions of high occlusion and object absence in the real multiview setup. Notably, in our Franka Emika Panda robot setup, we observed that the robot's slot also associated the shadow of the end-effector with the robot's representation. This behavior arises because our framework is unsupervised and there is no explicit control

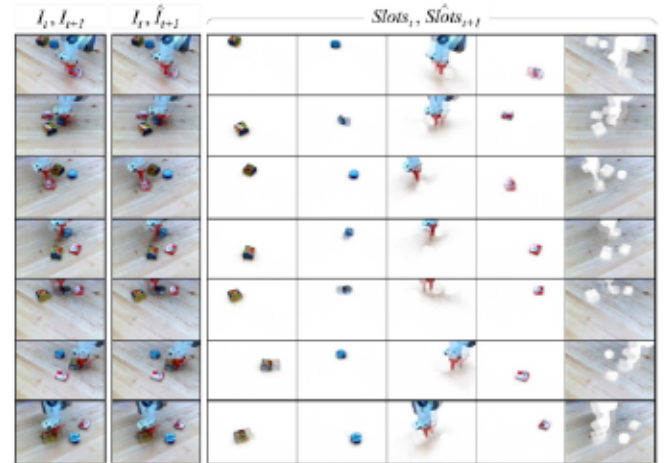


Fig. 19. *Examples of single-step dynamics prediction using InvSlotGNN transferred to real-world robot images*. Starting with the current scene image from an arbitrary uncalibrated camera angle, our model accurately estimates the future state of each slot based on the robot's pushing action in the real-world coordinate system, and then generates the corresponding future scene image.

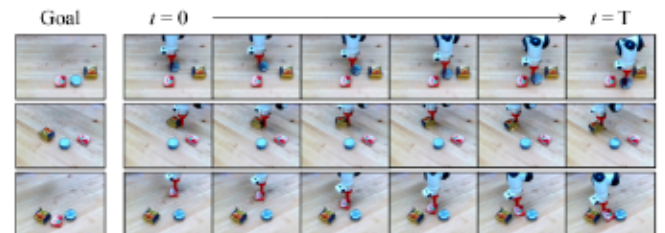


Fig. 20. *Real-world control experiment using SlotTransport and InvSlotGNN*. Observations from an uncalibrated camera with various viewpoint angles are used to optimize a sequence of pushing actions to achieve the goal image.

over what the slots represent. However, it also demonstrates the strength of our approach in learning object-centric representations, as the shadow of the robot correlates with the robot's pose. In addition, the model effectively handles other real-world factors such as lighting conditions, real tabletop textures, and reflections of objects.

We also illustrate single-step dynamics prediction of InvSlotGNN with the real robot. As shown in Fig. 19, our unsupervised model for dynamics, which is invariant to camera viewpoints, transferred well to the real-world setting. Importantly, no camera

calibration was performed to obtain the future dynamics predictions, highlighting the practicality and robustness of our proposed method in real-world settings where camera calibration can be challenging. In addition, we performed action planning for multiview observations (see Fig. 20). Given a goal image, the robot successfully pushed the object back to its goal pose after it was displaced.

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. *SlotTransport* has demonstrated its capability to learn robust and temporally aligned object-centric slots from both singleview and multiview observations, maintaining high accuracy even under occlusions and varying viewpoints.

Alongside *SlotTransport*, we introduced *SlotGNN*, an unsupervised graph-based dynamics model designed to predict the future states of multiobject scenes using the discovered slots. *SlotGNN* leverages the relational reasoning capabilities of graph neural networks to model interactions between objects, providing accurate and dynamic predictions essential for complex robotic control tasks and long-horizon dynamics prediction. By conditioning on the robot's actions, *SlotGNN* enables applications such as model-based action planning, enhancing the utility of the learned dynamics in practical robotic scenarios.

Building on these foundations, we developed *InvSlotGNN* to address the challenges of learning dynamics from multiview observations. *InvSlotGNN* extends the capabilities of *SlotGNN* by incorporating mechanisms to handle varying camera viewpoints, enabling the model to learn dynamics invariant to the camera angle. This advancement is critical for real-world applications where camera calibration is challenging or the viewpoints can vary unpredictably. *InvSlotGNN* successfully combines the strengths of *SlotTransport*'s robust object-centric representation with *SlotGNN*'s dynamic prediction, offering a comprehensive solution for multiview, action-conditioned scene dynamics prediction.

Our extensive experiments demonstrate the effectiveness of *SlotTransport*, *SlotGNN*, and *InvSlotGNN*. Importantly, we show that our unsupervised approach can be transferred from simulation to real-world settings with minimal additional data, enabling object discovery and dynamic modeling solely from RGB images. This capability highlights the practical robustness and applicability of our framework in real-world scenarios.

Despite the significant advancements, we recognize some limitations in our current approach. One key aspect is that our slot discovery process 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. This requirement could potentially limit the adaptability and scalability of our method in more complex and dynamic environments. Developing a more adaptive mechanism that automatically determines the required slot count based on the scene

complexity could be a promising direction for future research. In addition, integrating other sensory modalities, such as depth or tactile information, could further enhance the richness and accuracy of the learned representations.

In conclusion, this work not only provides robust and temporally consistent object-centric representations but also accurately predicts future states of multiobject scenes from both singleview and multiview observations. These contributions enables more advanced and adaptive robotic applications, emphasizing the potential of unsupervised methods in dynamic and real-world environments.

REFERENCES

- [1] A. Rezazadeh, A. Badithela, K. Desingh, and C. Choi, "SlotGNN: Unsupervised discovery of multi-object representations and visual dynamics," in *Proc. 2024 IEEE Int. Conf. Robot. Autom.*, 2024.
- [2] 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.
- [3] D. Marr, *Vision: A Computational Investigation Into the Human Representation and Processing of Visual Information*. Cambridge, MA, USA: MIT Press, 2010.
- [4] C. Finn and S. Levine, "Deep visual foresight for planning robot motion," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2017, pp. 2786–2793.
- [5] P. Agrawal, A. Nair, P. Abbeel, J. Malik, and S. Levine, "Learning to poke by poking: Experiential learning of intuitive physics," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, vol. 29.
- [6] D. Driess, Z. Huang, Y. Li, R. Tedrake, and M. Toussaint, "Learning multi-object dynamics with compositional neural radiance fields," in *Proc. Conf. Robot Learn.*, PMLR, 2023, pp. 1755–1768.
- [7] Y. Ye, D. Gandhi, A. Gupta, and S. Tulsiani, "Object-centric forward modeling for model predictive control," in *Proc. Conf. Robot Learn.*, PMLR, 2020, pp. 100–109.
- [8] H. Qi, X. Wang, D. Pathak, Y. Ma, and J. Malik, "Learning long-term visual dynamics with region proposal interaction networks," 2020, arXiv: 2008.02265.
- [9] M. Minderer, C. Sun, R. Villegas, F. Cole, K. Murphy, and H. Lee, "Unsupervised learning of object structure and dynamics from videos," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, vol. 32.
- [10] P. Battaglia, R. Pascanu, M. Lai, D. Rezende, and K. Kavukcuoglu, "Interaction networks for learning about objects, relations and physics," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, vol. 29.
- [11] A. Sanchez-Gonzalez et al., "Graph networks as learnable physics engines for inference and control," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2018, pp. 4470–4479.
- [12] T. Kipf, E. Fetaya, K.-C. Wang, M. Welling, and R. Zemel, "Neural relational inference for interacting systems," in *Proc. Int. Conf. Mach. Learn.*, 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 *Proc. IEEE Int. Conf. Robot. Automat.*, 2019, pp. 1205–1211.
- [14] A. Sanchez-Gonzalez, J. Godwin, T. Pfaff, R. Ying, J. Leskovec, and P. Battaglia, "Learning to simulate complex physics with graph networks," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2020, pp. 8459–8468.
- [15] W. Yuan, C. Paxton, K. Desingh, and D. Fox, "Sornet: Spatial object-centric representations for sequential manipulation," in *Proc. Conf. Robot Learn.*, 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," *Adv. Neural Inf. Process. Syst.*, vol. 30, pp. 4539–4547, 2017.
- [17] H. Yu and C. Choi, "Self-supervised interactive object segmentation through a singulation-and-grasping approach," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2022, pp. 621–637.
- [18] H. Yu, X. Lou, Y. Yang, and C. Choi, "IOSG: Image-driven object searching and grasping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2023, pp. 3145–3152.
- [19] F. Locatello et al., "Object-centric learning with slot attention," *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 11525–11538, 2020.

- [20] T. D. Kulkarni et al., "Unsupervised learning of object keypoints for perception and control," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, vol. 32.
- [21] A. Rezazadeh and C. Choi, "KINet: Unsupervised forward models for robotic pushing manipulation," *IEEE Robot. Automat. Lett.*, vol. 8, no. 10, pp. 6195–6202, Oct. 2023.
- [22] M. S. Sajjadi et al., "Rust: Latent neural scene representations from unposed imagery," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 17297–17306.
- [23] M. S. Sajjadi et al., "Scene representation transformer: Geometry-free novel view synthesis through set-latent scene representations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 6229–6238.
- [24] V. Sitzmann, S. Rezkikov, B. Freeman, J. Tenenbaum, and F. Durand, "Light field networks: Neural scene representations with single-evaluation rendering," *Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 19313–19325, 2021.
- [25] A. R. Kosiorek et al., "NeRF-VAE: A geometry aware 3D scene generative model," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2021, pp. 5742–5752.
- [26] M. Seitzer, S. Van Steenkiste, T. Kipf, K. Greff, and M. S. Sajjadi, "DYST: Towards dynamic neural scene representations on real-world videos," 2023, arXiv: 2310.06020.
- [27] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2017, pp. 1263–1272.
- [28] E. Camacho and C. Alba, *Model Predictive Control, Ser. Advanced Textbooks in Control and Signal Processing*. London, U.K.: Springer London, 2013.
- [29] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 5026–5033.
- [30] Y. Zhu et al., "robosuite: A modular simulation framework and benchmark for robot learning," 2020, arXiv: 2009.12293.
- [31] 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 *Proc. Int. Conf. Adv. Robot.*, 2015, pp. 510–517.
- [32] A. Kirillov et al., "Segment anything," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023.
- [33] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 586–595.



Alireza Rezazadeh received the B.Sc. degree in mechanical engineering from Sharif University of Technology, Tehran, Iran, in 2016, the M.Sc. degree in mechanical engineering from the University of Illinois Chicago, IL, USA, in 2019, and the M.Sc. and Ph.D. degrees both in electrical and computer engineering from the University of Minnesota, Twin Cities, MN, USA, in 2022 and 2024, respectively.

His Ph.D. research focused on graph-structured representation learning for robotic manipulation.



Houjian Yu received the B.Eng degree in electrical engineering from North China Electric Power University, Beijing, China, in 2018, and the M.S. degree in electrical and computer engineering from University of California San Diego, CA, USA, in 2020. He is currently working toward the Ph.D. degree in computer engineering with the University of Minnesota, MN, Minneapolis, USA.

His research interests are in robotics and machine learning.



Karthik Desingh (Member, IEEE) received the Ph.D. degree in computer science and engineering from the University of Michigan, Ann Arbor, MI, USA, in 2020.

He is currently an Assistant Professor with the Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN, USA and is affiliated with the Minnesota Robotics Institute (MnRI). He directs the Robotics: Perception and Manipulation (RPM) Lab. He completed his Postdoctoral position with the University of Washington (UW).

During his Ph.D., he was closely associated with the Robotics Institute and Michigan AI. He researches the intersection of robotics, computer vision, and machine learning, primarily focusing on providing perceptual capabilities to robots using deep learning and probabilistic techniques to perform grasping, manipulation, and mobile manipulation tasks in unstructured environments.

Dr. Desingh serving as an Associate Editor for conferences and journals such as IROS, ICRA, and RA-L. His work has been awarded the best workshop paper award at RSS 2019 and nominated as a finalist for the best systems paper award at CoRL 2021.



Changhyun Choi (Member, IEEE) received the Ph.D. degree in robotics with Georgia Institute of Technology, Atlanta, GA, USA, in 2014.

He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN, USA. Before joining the UMN, he was a Postdoctoral Associate with the Computer Science & Artificial Intelligence Lab, Massachusetts Institute of Technology. His research interests are in visual perception for robotic manipulation, with a focus on deep learning

for object grasping and assembly manipulation, soft manipulation, object pose estimation, visual tracking, and active perception. Dr. Choi is the recipient of the NSF CAREER Award, Sony Research Award, Russell J. Penrose Excellence in Teaching Award, and ICRA 2022 Outstanding Student Paper Award.