# Latent Space Planning for Unobserved Objects with Environment-Aware Relational Classifiers

Yixuan Huang[1], Jialin Yuan[2], Weiyu Liu[3], Chanho Kim[2], Li Fuxin[2], and Tucker Hermans[1,4]

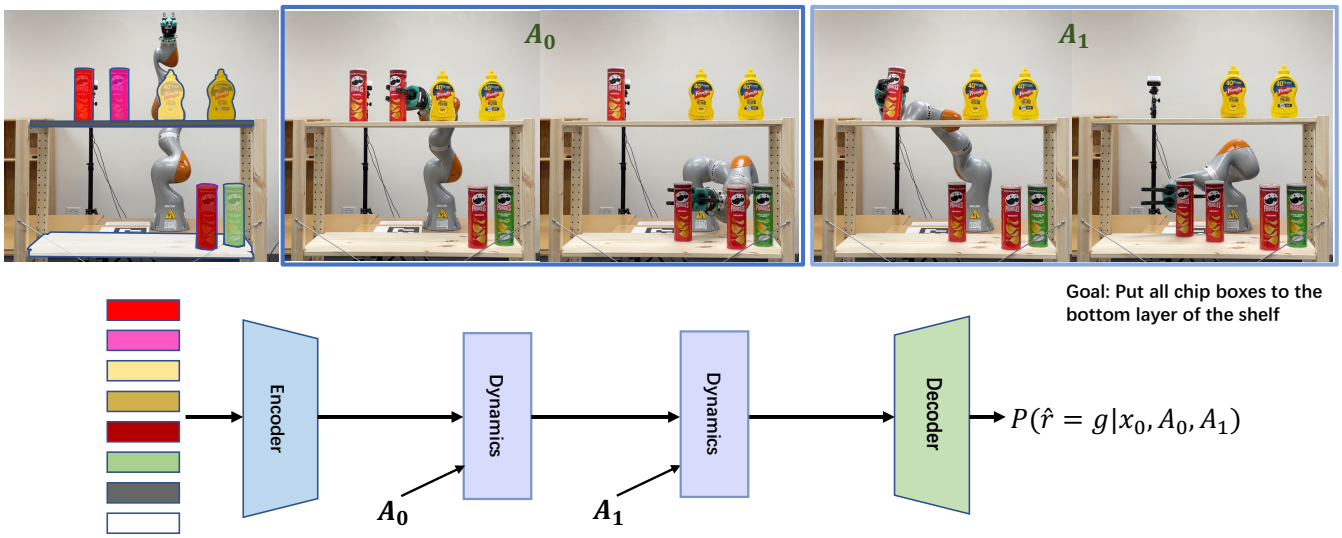


**Fig. 1: Multi-step Planning:** The robot encodes the segmented scene point cloud into a learned latent state space using the encoder. The robot then plans a sequence of actions using a learned latent dynamics function multiple times to predict relations that match the goal relations $\mathbf{g}$. The robot plans to pick-and-place two red chip boxes separately to achieve the goal of organizing all chip boxes to the bottom layer of the shelf.

## I. Introduction

For robots to act as our assistants, caregivers, and coworkers at home, at work, and in the wild, they will need to contend with varying environments. Environmental structure has profound impacts on manipulation. Placing a large tray on a small table is much different than placing a cup onto a high shelf. Furthermore, we want our robots to understand multi-object dynamics and manipulate many objects at once as shown in recent work [1]. However, Huang and colleagues [1] do not consider the environment and object-environment relations during planning. As such the model cannot perform seemingly simple desirable robotic behaviors such as picking an object from one shelf and placing it on another. No work to date explicitly reasons about object-environment relational dynamics when manipulating multiple, novel objects observed from partial-view data. Furthermore, with more complex environments, some objects will more likely become occluded. Thus, we need the robot to have a memory of the hidden objects to be able to reason about them as well.

To address these challenges, we propose a novel neural network model that explicitly represents the current environment and multiple objects. Using the proposed model, we can predict the relations between multiple objects and the environment before and after the robot's actions, which enables the robot to plan to desired goal relations. The model also predicts the pose of the objects and environments and how poses change based on the robot's actions. This allows the robot to reason about the occluded objects with the predicted pose even when they disappear from the current observation. We demonstrate that our model generalizes to novel scenes and objects operating only from partial-view point clouds, which enables direct sim-to-real transfer without any fine-tuning on the real-world data.

Research over the past few years has shown an increasing ability for robots to manipulate and rearrange novel objects in cluttered environments [2–7]. Significantly less attention has been given to reasoning about novel environments and how environmental structures relate and interact with the objects being manipulated. For example, a robot should be able to reason that an object on a high shelf is above an object in a drawer lower in the same cabinet. Giving robots inter-object and object-environment logical relational understanding enables logic-based task planning [8] and allows human operators to provide goals as logical states. Defining these logical relations by hand to operate on sensor data has proved burdensome and inaccurate, motivating the use of learning-based tools to directly predict semantic relations [1,3–5,7,9,10]. A recent paper from Huang et al. [1] proposes the first approach to perform semantic reasoning about multi-object dynamic interactions from partial-view point clouds. This is in contrast to previous works that are

[1]Robotics Center and Kahlert School of Computing, University of Utah, Salt Lake City, UT 84112, USA. [2] Oregon State University. [3] Georgia Tech. [4]NVIDIA; Seattle, WA, USA. yixuan.huang@utah.edu

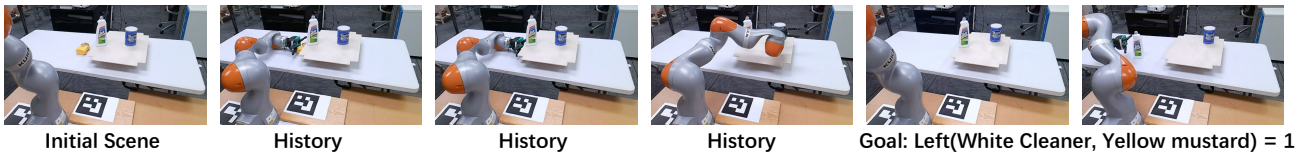


**Fig. 2:** We show one example of how our model can achieve desired goal relations including unobserved objects. After the robot pushes the yellow mustard to the bottom of the shelf and pushes the coffee can, our model can predict the pose of the yellow mustard even if the robot cannot see the yellow mustard. Then the robot can pick and place the white cleaner to achieve the goal relation.

limited either by manipulating only one object at a time [2–7,9] or lack explicit semantic reasoning [11,12].

Prior work [13–15] in computer vision has addressed object occlusion through a memory-based model for visual tracking. However, these memory models primarily serve the purpose of appearance matching and cannot reason about the potential movements of occluded objects in response to a robot's actions. In the context of robotics, Shafiullah et al. [16] proposed an implicit model for semantic navigation, but this model cannot be directly applicable to manipulation tasks.

In this work, we show that planning using our learned network enables a robot to achieve many multi-object rearrangement tasks in different environments and to reason and plan with occluded objects. Our model shows superior performance to an extension of [1]. We show more videos on our website: `https://sites.google.com/view/erdunobserved`.

## II. Approach

We assume the robot perceives the world as a point cloud $Z$ with $N$ associated object and environment segments $O_i \subset Z, i = 1, 2, ..., N$. The robot receives a goal defined as a logical conjunction of $M$ desired object and environment relations, $\mathbf{g} = r_1 \wedge r_2 \wedge ... \wedge r_M, r_j \in \mathcal{R}$. Where $\mathbf{g}$ represents the goal conjunction, $r_j$ denotes each goal relation, and $\mathcal{R}$ represents the set of all possible relations. We provide our robot with a set of $L$ parametric action primitives $\mathcal{A} = \{A_1, \ldots, A_L\}$ where $A_l$ defines a discrete skill, which has associated skill parameters $\theta_l$. For example, a push skill is defined with parameters encoding the push direction and length, or a pick-and-place skill is defined with parameters encoding the grasp and placement poses. We define the robot's planning task as finding a sequence of skills and skill parameters $\tau = (A_0, \ldots, A_{H-1})$ that, when sequentially executed, transform the objects such that they satisfy all relations defined in the logical goal $\mathbf{g}$. We propose a novel environment-aware relational dynamics transformer as eRDTransformer. We visualize our approach in Fig 1. The model contains three main components: an encoder, a decoder, and a latent dynamics model.

**Encoder:** The model takes in a segmented point cloud of the current observation $Z_t$. We pass the point cloud segments into the point cloud encoder [17] to get a feature vector for each segment, $P_i = E_p(O_i) \forall O_i \in Z_t$. We use a learned positional embedding to encode the segment identifier as $I_i = \texttt{Emb}_p(i)$. We then concatenate the feature vector with the positional embedding identifier for each object, $P'_i = P_i \oplus I_i$. To improve the generalization ability, we randomly generate the object IDs during training [18] over a range larger than the highest number of objects expected to be seen at deployment. The network then passes these updated features through an encoder $E$ generating a latent feature $\mathbf{x}_i = E(P'_i)$ for each segment. The combined output of all encoders forms the latent state $\mathbf{x}^L_t$.

**Decoder:** Based on these latent features, we can use our decoder, $D$, to generate all outputs associated with the current time step. We have three distinct kinds of decoders (1) relational decoders, $D_r$, (2) pose decoders, $D_p$, and (3) an environment identity decoder, $D_e$. The relational decoder predicts all segment-segment relations using a set of binary relational classifiers $\hat{\mathbf{r}}_t = D_r(\mathbf{x}^L_t)$. The pose decoder predicts the pose of all segments $\hat{p}_t = D_p(\mathbf{x}^L_t)$. The environment identity decoder predicts if a given segment is a movable object or an immobile part of the environment, $\hat{\mathbf{y}}_t = D_e(\mathbf{x}^L_t)$ where $\hat{\mathbf{y}}$ defines a vector of outputs for all segments.

**Actions and Dynamics:** We can learn a dynamics function $\delta$ to predict the resulting latent state based on the current latent state and a selected action $\hat{\mathbf{x}^L}_{t+1} = \delta(\mathbf{x}^L_t, A_t)$, where $A_t$ contains the discrete skill to use and its associated action parameters, $\theta$. We use a discrete parameter to define which object the action will operate on, and we encode this into the neural network using the learned positional embedding $\texttt{Emb}_p(i)$ for segment $i$. We use an action encoder, $E_a$, to transform the raw continuous action parameters, $\theta_c$ sent to the robot controller or motion planner into learned action features, $\theta'_c$. We denote the total encoded action parameters as $\theta' = \texttt{Emb}_p(i) \oplus \theta'_c$. We learn a separate dynamics function for each robot skill primitive. This removes the burden of the network having to learn to map skill codes to distinct dynamics outcomes. When needing to be explicit, we will denote the skill-specific dynamics as $\delta_l$ for the dynamics function associated with skill primitive $A_l$.

**Latent Space Planning:** Based on the latent state predicted using our learned dynamics function, we can use our decoder to predict the relations at the resulting state $\hat{\mathbf{r}}'_{t+1} = D_r(\hat{\mathbf{x}^L}_{t+1})$. By recursively calling this dynamics function with a sequence of actions $\tau = (A_0, \ldots, A_{H-1})$, we can generate rollouts with a time horizon $H$ for use in a planning algorithm to compare the predicted relations with the goal relations.

**Reason and Planning with Occluded objects:** To reason about the unobserved objects due to occlusion, we first leverage the video tracker of [19] to track which objects disappear during the robot motions. Then our model predicts the pose of the disappeared objects with our learned latent dynamics $\delta$. With the predicted pose of the disappeared

**TABLE I:** Comparison in terms of planning success rate with a different number of planning steps.

| Simulation | 1 step | 2 steps | 3 steps |
|---|---|---|---|
| eRDTransformer | 1.0 | 0.9 | 0.85 |
| eRD-GNN | 1.0 | 0.85 | 0.1 |
| Real-world | 1 step | 2 steps | 3 steps |
| eRDTransformer | 1.0 | 1.0 | 0.8 |
| eRD-GNN | 1.0 | 0.0 | 0.0 |

objects, we obtain the transformed point cloud of the disappeared objects, which enables latent space planning with the transformed point clouds of the unobserved objects and other visible objects.

## III. Experiments & Results

We collect large training datasets using the Isaac Gym simulator [20]. We define one baseline eRD-GNN as the environment-aware extension to the relational-dynamics graph net [1]. We then evaluate how well these models work in the context of manipulation planning for both single-step and multi-steps as shown in table. I. We find that for eRDTransformer, the success rate drops with plan length but achieves high success rates in both simulation and the real world. Furthermore, eRDTransformer outperforms the baseline eRD-GNN especially in multi-step planning. Figure 1 visualizes one real-world visualization of multi-step execution. We then show the performance of eRDTransformer with an unobserved object in Fig. 2. We found that our framework performs well in understanding the location of the disappeared object and can achieve the goal with respect to the disappeared object.

## IV. Discussion

We propose a novel framework to reason about relations between objects and environments observed only through partial-view point clouds. The proposed framework enables the robot to envision how the actions cause objects to move in the environment through a learned dynamics model and to reason about the effects of these actions on the relations. By encoding how actions change the relations between objects and the environment, our approach can achieve multi-step planning with a variable number of objects and environmental components. To the best of our knowledge, this is the first work that shows the relational dynamics models are able to support manipulation tasks that involve environments and even objects that are occluded after initial observations.

## References


[1] Y. Huang, A. Conkey, and T. Hermans, "Planning for Multi-Object Manipulation with Graph Neural Network Relational Classifiers," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023. [Online]. Available: https://arxiv.org/abs/2209.11943

[2] A. Murali, A. Mousavian, C. Eppner, C. Paxton, and D. Fox, "6-dof grasping for target-driven object manipulation in clutter," in *IEEE Intl. Conf. on Robotics and Automation*. IEEE, 2020, pp. 6232–6238.

[3] C. Paxton, C. Xie, T. Hermans, and D. Fox, "Predicting Stable Configurations for Semantic Placement of Novel Objects," in *Conference on Robot Learning (CoRL)*, 11 2021. [Online]. Available: https://arxiv.org/abs/2108.12062

[4] W. Liu, C. Paxton, T. Hermans, and D. Fox, "StructFormer: Learning Spatial Structure for Language-Guided Semantic Rearrangement of Novel Objects," in *IEEE Intl. Conf. on Robotics and Automation*, 2022. [Online]. Available: https://sites.google.com/view/structformer

[5] Y. Zhu, J. Tremblay, S. Birchfield, and Y. Zhu, "Hierarchical planning for long-horizon manipulation with geometric and symbolic scene graphs," in *IEEE Intl. Conf. on Robotics and Automation*, 2021.

[6] Y. Lin, A. S. Wang, E. Undersander, and A. Rai, "Efficient and interpretable robot manipulation with graph neural networks," *IEEE Robotics and Automation Letters*, 2022.

[7] M. Sharma and O. Kroemer, "Relational learning for skill preconditions," in *Conference on Robot Learning*, 2020.

[8] C. R. Garrett, T. Lozano-Pérez, and L. P. Kaelbling, "Pddlstream: Integrating symbolic planners and blackbox samplers via optimistic adaptive planning," in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 30, 2020, pp. 440–448.

[9] R. Li, A. Jabri, T. Darrell, and P. Agrawal, "Towards practical multi-object manipulation using relational reinforcement learning," in *IEEE Intl. Conf. on Robotics and Automation*, 2020, pp. 4051–4058.

[10] K. Kase, C. Paxton, H. Mazhar, T. Ogata, and D. Fox, "Transferable task execution from pixels through deep planning domain learning," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 10 459–10 465.

[11] M. Wilson and T. Hermans, "Learning to manipulate object collections using grounded state representations," in *Conference on Robot Learning*, 2020, pp. 490–502.

[12] H. Suh and R. Tedrake, "The surprising effectiveness of linear models for visual foresight in object pile manipulation," in *Intl. Workshop on Algorithmic Foundations of Robotics*, 2020.

[13] H. K. Cheng and A. G. Schwing, "XMem: Long-term video object segmentation with an atkinson-shiffrin memory model," in *ECCV*, 2022.

[14] J. Cai, M. Xu, W. Li, Y. Xiong, W. Xia, Z. Tu, and S. Soatto, "Memot: Multi-object tracking with memory," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 8090–8100.

[15] S. W. Oh, J.-Y. Lee, N. Xu, and S. J. Kim, "Video object segmentation using space-time memory networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[16] N. M. M. Shafiullah, C. Paxton, L. Pinto, S. Chintala, and A. Szlam, "Clip-fields: Weakly supervised semantic fields for robotic memory," *arXiv preprint arXiv:2210.05663*, 2022.

[17] W. Wu, Z. Qi, and L. Fuxin, "PointConv: Deep Convolutional Networks on 3D Point Clouds," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2019, pp. 9621–9630.

[18] H. Cui, Z. Lu, P. Li, and C. Yang, "On positional and structural node features for graph neural networks on non-attributed graphs," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 3898–3902. [Online]. Available: https://arxiv.org/abs/2107.01495

[19] J. Yuan, J. Patravali, H. Nguyen, C. Kim, and L. Fuxin, "Maximal cliques on multi-frame proposal graph for unsupervised video object segmentation," *arXiv preprint arXiv:2301.12352*, 2023.

[20] NVIDIA, "Isaac Gym," https://developer.nvidia.com/isaac-gym, 2020.