Graph-Based Reward Learning and Automatic Subtask Discovery for Long-Horizon Manipulation

Anonymous

***Abstract*—Learning long-horizon manipulation skills from visual demonstrations remains challenging because reward design is difficult, manual subtask annotation is expensive, and pixel-based representations often generalize poorly across visual variations. Imitation learning (IL) enables efficient policy acquisition from demonstrations, but policies trained only by imitation often lack robustness when tested out of demos distribution. Reinforcement learning (RL), on the other hand, can improve policy performance through experience, but requires overly engineered and often sparse reward functions designed by domain experts.**

In this work, we propose a graph-based inverse reinforcement learning (IRL) framework that bridges IL and RL by learning semantically grounded reward functions from demonstrations.

Rather than directly relying on raw images, we represent the scene as a graph of detected objects and their relations, and encode this graph with a graph neural network. The graph undergoes a weighted pooling mechanism that emphasizes dynamically relevant task entities.

The learned representation is used to define a dense reward based on latent-space distance to the goal, appropriately tracking task execution. Interestingly, we note that the learned reward evolves according to an interpretable stage-wise structure that reflects semantic progress through the task. This structure becomes especially useful in long-horizon settings, where it can provide a signal for identifying subtask boundaries without manual segmentation. This makes the framework suitable both for direct RL with a full-task reward and, in more complex tasks, for subtask-level policy learning.

Experiments on multi-step manipulation tasks show that the proposed object-centric reward improves downstream RL performance over pixel-based and graph-based baselines on complex manipulation tasks, and yields semantically meaningful reward transitions on long-horizon tasks. These results suggest that structured object-centric reward learning is a promising mechanism for combining imitation learning signals with reinforcement learning in long-horizon robotic manipulation.

***Index Terms*—Robot learning, Reinforcement learning, Graph neural networks, Representation learning, Inverse RL**

I. Introduction

Long-horizon robotic manipulation requires reasoning over multiple objects, temporally extended interactions, and semantically ordered stages of execution. While Imitation Learning (IL) can efficiently learn policies from expert demonstrations, it often suffers from limited robustness and poor generalization, especially when deployment conditions differ from those seen during training. Reinforcement Learning (RL), in contrast, can improve policies through interaction, but depends on reward functions that are often difficult to define for complex manipulation tasks even with domain expert knowledge.

A promising direction to combine the strengths of IL and RL is reward learning from demonstrations: demonstrations provide the supervision needed to infer task progress, and RL uses the learned reward to optimize behavior through trial and error. In particular, learning rewards directly from videos removes the need for action annotations and enables scalable data collection [1]. However, two challenges remain. First, pixel-based representations carry irrelevant visual details such as texture, lighting, and viewpoint, which often leads to poor generalization across visual domain shifts. Second, long-horizon tasks require representing temporal progress and decomposing behaviors into meaningful key stages, making it difficult to align demonstrations and extract informative reward signals that reflect meaningful task completion.

Recent work has therefore explored object-centric representations, in which a scene is abstracted into objects and their relations [2]. Such representations are compositional, more robust to appearance changes, and better suited to reasoning about manipulation structure. Still, using object-centric abstractions for reward learning in long-horizon tasks remains challenging: the representation must capture task progression, suppress irrelevant motion, and expose structure that can support decomposition into subtasks when needed.

In this work, we propose an object-centric inverse RL framework for long-horizon manipulation. Each observation is represented as a graph of detected objects, and a graph neural network is trained to encode task progress from demonstration videos. To make the representation properly task-aligned, we introduce a weighted pooling mechanism that emphasizes active objects while suppressing robot nodes, reducing the influence of robot motion that may dominate visual change without reflecting semantic progress.

Our main observation is that the learned object-centric reward often exhibits an interpretable stage-wise structure over time, reflecting semantic progress through the task. This property is not limited to long-horizon settings: even when a task can be solved effectively with a single full-task reward, the learned reward may still reveal intermediate phases of execution. In long-horizon tasks, however, this structure becomes particularly valuable, since it can be exploited to identify subtask boundaries without manual substep segmentation and can later support decomposition into simpler sub-policies.

We evaluate the approach on two manipulation tasks that illustrate these two uses of the learned reward. On *Match Regions* [4], we show that the learned object-centric reward improves downstream RL performance over pixel-based and previous graph-based baselines. On *Shoes in Box* [5], we show that the learned reward exhibits semantically meaningful transitions that are especially useful for analyzing long-horizon task structure and motivating future subtask decomposition.

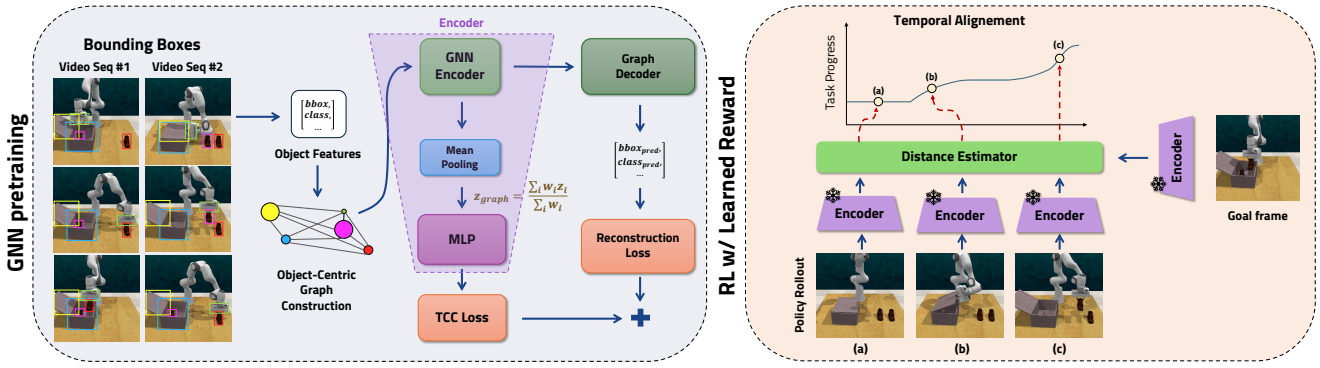

Fig. 1. Overview of the proposed framework. Demonstration videos are converted into object-centric graphs and encoded with a graph neural network trained using temporal cycle-consistency and reconstruction losses. The learned encoder is then frozen and used to define a reward for reinforcement learning by measuring the latent-space distance between the current observation and the goal. The resulting reward captures semantic task progress and can be used directly for policy learning; in longer-horizon tasks, its stage-wise temporal evolution can additionally be exploited to reveal subtask structure.

The contributions of this work are summarized as follows:

- An object-centric IRL framework that learns dense rewards from action-free video demonstrations using graph representations;
- A weighted graph pooling mechanism that emphasizes task-relevant object dynamics while suppressing robot-dominated motion;
- Empirical evidence that the learned reward improves RL performance on a structured manipulation task and exhibits interpretable step-like progression on a longer-horizon task.

An overview of the proposed framework is shown in Fig. 1.

II. Methodology

A. Problem Formulation

We consider the problem of learning reward functions for long-horizon manipulation from action-free video demonstrations. Let $\mathcal{D} = \{V_k\}_{k=1}^{K}$ denote a set of demonstration videos for a task $\mathcal{T}$, where each video $V_k = \{I_k^1, I_k^2, \ldots, I_k^{T_k}\}$ is a sequence of image frames.

Our goal is to learn a reward function $r(s)$ that reflects task progress and can be used for downstream policy optimization with reinforcement learning. To this end, we learn a representation function $\phi(\cdot)$ that maps each observation to a latent space where distances reflect progress toward task completion.

Unlike pixel-based approaches, we construct an object-centric representation for each frame and learn rewards from this structured scene abstraction.

B. Object-Centric Graph Construction

For each frame I_t, we extract a set of N objects using an off-the-shelf detector and represent the scene as a fully-connected graph $G_t = (V_t, E_t)$. Each node $v_i \in V_t$ corresponds to a detected object and is described by the semantic class and bounding-box geometry as node features. Edges $e_{ij} \in E_t$ encode pairwise spatial relations between objects, namely the relative distances.

This object-centric abstraction removes much of the irrelevant visual detail present in raw images while preserving the entities and interactions that matter for manipulation. As a result, it is better suited than pixel-level representations for capturing structured task progress in multi-object scenes.

C. Graph-Based Representation Learning

Given a sequence of graphs $\{G_t\}$, we learn an embedding function $\phi(G_t)$ that produces a compact representation of the task state encoded in the graph.

We use a graph neural network (GNN) encoder that processes node and edge features through message-passing layers to produce node embeddings $\{z_i\}_{i=1}^{N}$. These node embeddings are then aggregated to obtain a graph-level representation:

$$z_t = \phi(G_t). \tag{1}$$

To make the representation more sensitive to semantic task progression, we introduce a weighted pooling mechanism that emphasizes active objects while suppressing robot nodes. This encourages the latent state to focus on changes in the environment that reflect task completion rather than on robot motion alone.

Weighted Graph Pooling: To emphasize task-relevant dynamics, we aggregate node embeddings with a weighted pooling operator:

$$z_g = \frac{\sum_i w_i z_i}{\sum_i w_i}. \tag{2}$$

The node weights are defined as

$$w_i = (1 + (\alpha - 1) \cdot \text{active}_i)(1 - \text{robot}_i), \tag{3}$$

where $\text{active}_i \in \{0, 1\}$ indicates whether node i is currently active, and $\text{robot}_i \in \{0, 1\}$ indicates whether the node corresponds to the robot.

In practice, robot nodes are explicitly suppressed by setting their contribution to zero. This design choice prevents robot motion from dominating the scene representation. Although robot motion is often the largest source of frame-to-frame variation, it is not always the most informative signal for

understanding the semantic task progression. For example, in a task such as placing two shoes into a box, progress is more directly reflected by the changing spatial relations between the shoes and the box—-such as whether a shoe has reached the box or been placed inside it—-than by the exact trajectory of the manipulator. We then expect that removing robot nodes can yield a reward signal less noisy and more aligned with meaningful task stages.

The activity variable is computed from the temporal displacement of bounding-box centers:

$$\text{active}_i = \mathbb{I}\left(\frac{\|c_t - c_{t-k}\|}{\text{diag}_i} > \tau\right), \tag{4}$$

where c_t is the center of the bounding box at time t, diag_i is the diagonal length of the bounding box, and τ is a threshold. Normalizing by the box diagonal makes the activity criterion scale-invariant, so that activity depends on relative motion rather than absolute pixel displacement.

In our experiments, we set $\tau = 0.005$ and apply temporal smoothing over a window of $k = 40$ frames to reduce sensitivity to jitter and simulator spikes. We also make the activity bit persistent: once an object becomes active, it remains active for the rest of the episode. This persistence encourages a monotonic notion of progress, which is useful for long-horizon reward shaping. Without it, an object could become inactive after reaching its target even though it should remain semantically relevant for representing which parts of the task have already been completed.

D. Self-Supervised Pretraining

The graph encoder is trained in a self-supervised manner using two complementary objectives that enforce temporal consistency and structural fidelity: one for temporal alignment across demonstrations and one for preserving task-relevant scene structure.

a) Temporal Alignment: We adopt a temporal cycle-consistency (TCC) objective [3] to align embeddings across demonstration videos. Given two sequences, embeddings corresponding to similar stages of the task are encouraged to be close in latent space, even when demonstrations differ in timing or execution details. This helps the representation capture task progression rather than raw frame similarity.

b) Structured Reconstruction: To preserve object-level information, we introduce a reconstruction objective that encourages the embedding to retain descriptors such as object identities and bounding-box geometry. Let $\psi(\cdot)$ be a decoder. We minimize

$$\mathcal{L}_{rec} = \|\hat{o}_t - o_t\|_2^2, \tag{5}$$

where o_t denotes the object-level features and $\hat{o}_t = \psi(z_t)$ is their reconstruction.

This objective encourages the latent representation to preserve the structural information needed for reward learning without requiring full pixel reconstruction.

c) Overall Objective: The final training objective combines temporal alignment and reconstruction:

$$\mathcal{L} = \mathcal{L}_{tcc} + \lambda \mathcal{L}_{rec} \tag{6}$$

E. Automatic Subtask Discovery

We observe that the learned full-task reward often exhibits a stage-wise profile over time. Rather than changing smoothly at every frame, the reward tends to evolve around semantically meaningful transitions in the scene, such as changes in object placement or interaction state. This structure reflects intermediate task progress more generally, even in settings where explicit decomposition is unnecessary. In long-horizon tasks with ordered interaction phases, however, we can additionally use it as a heuristic signal for subtask discovery without manual annotation.

More specifically, after learning the full-task reward, we evaluate each demonstration trajectory with the reward model and analyze the temporal evolution of the reward. Candidate transition frames are identified from significant variations in the reward gradient, which correspond to substantial changes in the underlying object-centric scene configuration.

These transition points are then used to segment each demonstration into a sequence of subtask trajectories. The resulting segments define subtask-specific datasets, each associated with a distinct phase of the overall manipulation behavior. This decomposition is useful because the reward signal relevant to one phase is often more specific than the reward for the full task. For example, in a task where two shoes must be placed into a box, the subtask corresponding to placing the first shoe should primarily reflect changes related to the moved shoe and the box, rather than also rewarding motion of the second shoe.

In this paper, we focus on validating the key prerequisite for this pipeline: namely, that the learned reward exhibits an interpretable step-like structure on long-horizon tasks. Full quantitative evaluation of segmentation quality and subpolicy composition is left for future work.

F. Reinforcement Learning with Learned Reward

After pretraining the encoder, we construct a reward in the learned latent space. Given the current observation graph G_t and a goal graph G_g, the reward is defined as:

$$r_t = -\|\phi(G_t) - \phi(G_g)\|_2 \tag{7}$$

This reward provides a dense measure of progress toward the goal and can be used directly with a standard RL algorithm. More importantly, the learned reward often reflects intermediate semantic progress through the task. In settings such as *Match Regions*, this structure need not be exploited explicitly, since a single full-task reward is sufficient for policy learning. In longer-horizon tasks with clearer procedural phases, such as *Shoes in Box*, the same structure becomes useful for identifying subtask boundaries and motivating decomposition into simpler learning problems.

G. Training Pipeline

The overall pipeline has two stages. First, we pretrain the graph encoder on demonstration videos using self-supervised temporal alignment and reconstruction objectives. Second, we

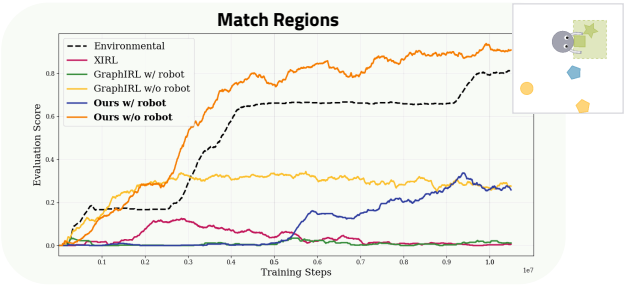

Fig. 2. Results on *Match Regions*. Our object-centric GNN-based reward learning approach achieves the best final performance and the fastest convergence among all compared methods. Notably, suppressing robot nodes during pooling leads to the strongest performance with 94.6% success, outperforming the hand-crafted environmental reward baseline at about 85% of success rate. This indicates that the learned object-centric reward is both more effective and more data-efficient for downstream RL.

freeze the encoder and use the learned representation to define a reward for reinforcement learning.

For long-horizon tasks, the second stage can additionally include subtask discovery from the reward trajectory, potentially followed by subtask-level reward learning and sequential subpolicy training. In this paper, we empirically validate the first two elements of this long-horizon pipeline: reward learning and the emergence of interpretable subtask structure in the learned reward.

III. Experiments

We evaluate our approach on two manipulation tasks that highlight complementary uses of the learned reward. The first task, *MatchRegions*, tests whether the learned object-centric reward is effective for downstream reinforcement learning and whether it can compete with a hand-crafted environmental reward and other IRL baselines. The second task, *Shoes in Box*, evaluates the learned reward in a longer-horizon setting, where its stage-wise structure becomes especially relevant for analyzing semantic task progression and its potential use for subtask decomposition.

A. Match Regions

The *MatchRegions* task is drawn from the MAGICAL benchmark [4] and evaluates the quality of the learned reward for downstream policy optimization. In this environment, the robot must place objects into their target regions, so successful behavior depends on correctly capturing object placement and scene configuration. Although the learned reward can still reflect intermediate semantic progress, this task can be solved effectively with a single full-task reward, so explicit decomposition into subtasks is not required.

We compare our method against a pixel-based IRL baseline, namely XIRL [1], and a graph-based variant – i.e., GraphIRL [2] – with different pooling strategies. Moreover, we also compare it against the hand-crafted environmental reward provided by the simulator. Figure 2 shows that our graph-based representation achieves the strongest overall performance. In particular, the version without robot nodes in the pooling stage reaches a final success rate of approximately 94.6%, outperforming the environmental reward baseline, which reaches about 85%, as shown also in Table I.

TABLE I
Final success rate on *MatchRegions*.

Method	Success Rate (%) ↑
Environmental reward	85.0
XIRL [1]	0.0
GraphIRL [2] (with robot)	0.0
GraphIRL [2] (without robot)	33.7
Ours (with robot)	33.3
Ours (without robot)	**94.6**

This result is notable because the environmental reward is manually designed using privileged task information, whereas our method learns its reward directly from demonstration videos. The learned object-centric reward is therefore not only competitive with hand-engineered supervision, but can also provide an even more effective optimization signal.

In addition to achieving the highest final performance, the proposed framework is also more data-efficient. As shown in Fig. 2, its learning curve rises earlier and reaches convergence faster than the environmental reward and the other learned baselines. This suggests that the object-centric latent reward provides denser and more informative feedback during training, enabling the policy to acquire the task with fewer interaction steps.

B. Shoes in Box

We next evaluate our method on *Shoes in Box*, a significantly more complexmanipulation task from RLBench [5] with a clear sequential structure. Solving the task requires multiple semantically distinct phases, including reaching the box, opening it, reaching a shoe, placing it into the box, and repeating the procedure for the second shoe. As in simpler tasks, the learned reward can reflect intermediate semantic progress; here, however, that structure is especially important because it may be exploited to decompose a long-horizon behavior into simpler stages.

We depict in Figure 3 a first qualitative observation, where the reward predicted by our encoder evolves with a clear step-like structure over time, with transitions aligned to meaningful task events. This suggests that the learned reward captures more than smooth proximity to the goal: it reflects semantically important changes in the object-centric scene configuration.

To quantify this property, we evaluate the learned encoder on trajectories from the validation set and compute the corresponding learned rewards. We divide the trajectories into *positive* trajectories, in which the task is completed successfully, and *negative* trajectories, in which it is not. For each set, we compute the mean cumulative reward and define the following ratio:

$$\rho = \frac{\bar{R}^{+}}{\bar{R}^{-}}, \tag{8}$$

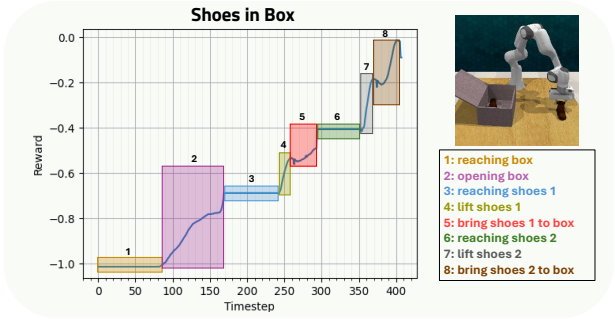

Fig. 3. Learned reward profile for *Shoes in Box*. The reward evolves through a sequence of step-like increases over time, with transitions aligned to semantically meaningful interaction phases such as opening the box and placing each shoe. This structured progression suggests that the learned reward can be used to identify candidate subtask boundaries in long-horizon manipulation tasks.

where $\bar{R}^+$ is the mean cumulative reward over successful trajectories and $\bar{R}^-$ is the mean cumulative reward over unsuccessful trajectories.

This metric measures how well the learned reward discriminates between successful and unsuccessful behaviors. Because the reward defined in Eq. 7 is always negative, smaller values indicate stronger discrimination, as they reflect more negative cumulative reward on unsuccessful trajectories relative to successful ones.

As shown in Table II, our method achieves the lowest ratio overall and the two lowest ratios among the compared methods, indicating that the learned reward more strongly separates successful trajectories from unsuccessful ones. In contrast, higher ratios indicate weaker discrimination, meaning that negative trajectories still accumulate rewards closer to those of successful executions. These results support the claim that the object-centric reward learned by our method is more suitable for downstream RL, since it better reflects whether the task is being solved correctly.

Taken together, the qualitative step structure and the quantitative reward-separation results indicate that the proposed framework captures meaningful long-horizon task progress in *Shoes in Box*.

IV. Conclusion

We presented an object-centric inverse reinforcement learning framework for long-horizon robotic manipulation that combines structured scene abstraction, reward learning from demonstrations, and reinforcement learning with learned rewards. By representing each observation as a graph of detected objects and their relations, the method reduces irrelevant visual variability and focuses reward learning on the entities and interactions that drive task progress. The proposed weighted pooling mechanism further improves this representation by emphasizing active objects while suppressing robot-dominated motion, resulting in reward signals that are more semantically aligned with the task.

Our experiments highlight two complementary strengths of the approach. On *Match Regions*, the learned graph-based reward improves downstream RL performance without requiring explicit task decomposition, even though it still reflects meaningful intermediate task progress. On *Shoes in Box*, the learned reward becomes especially informative, exhibiting a clear stage-wise profile aligned with semantically meaningful phases of the task and suggesting a natural path toward subtask discovery in longer-horizon settings.

Overall, these results suggest that object-centric reward learning provides a unified mechanism for combining demonstration-derived supervision with reinforcement learning in complex manipulation tasks. The learned reward can both guide end-to-end policy optimization and reveal semantic task structure, which becomes particularly valuable in long-horizon settings. An important next step is to fully exploit the discovered reward structure to segment demonstrations automatically, learn subtask-specific rewards, and train sequential subpolicies for complete long-horizon execution.

TABLE II

Reward discrimination on *Shoes in Box*, measured by the ratio between mean cumulative reward on successful and unsuccessful validation trajectories. Lower is better, since it indicates a larger discrepancy between positive and negative trajectories.

Method	Positive/Negative Ratio $\rho \downarrow$
XIRL	0.713
GraphIRL w/ robot	0.844
GraphIRL w/o robot	1.098
Ours w/ robot	**0.673**
Ours w/o robot	**0.684**

References

[1] K. Zakka, A. Zeng, P. Florence, J. Tompson, J. Bohg, and D. Dwibedi, "XIRL: Cross-embodiment inverse reinforcement learning," in *Proceedings of the Conference on Robot Learning*, pp. 537–546, 2022.

[2] S. Kumar, J. Zamora, N. Hansen, R. Jangir, and X. Wang, "Graph inverse reinforcement learning from diverse videos," in *Proceedings of the Conference on Robot Learning*, pp. 55–66, 2023.

[3] D. Dwibedi, Y. Aytar, J. Tompson, P. Sermanet, and A. Zisserman, "Temporal cycle-consistency learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 1801–1810.

[4] S. Toyer, R. Shah, A. Critch, and S. Russell, "The MAGICAL benchmark for robust imitation," in *Advances in Neural Information Processing Systems*, vol. 33, pp. 18284–18295, 2020.

[5] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison, "RLBench: The robot learning benchmark & learning environment," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3019–3026, 2020.