Object-Centric Reward Learning from Action-Free Videos for Long-Horizon Manipulation Beyond Teleoperation

Anonymous

***Abstract*— Teleoperated demonstrations have enabled substantial progress in robot learning, but they remain constrained by robot-specific interfaces, limited dexterity, and expensive data collection. We study how action-free video demonstrations can instead provide supervision for learning task-progress rewards that are later used for reinforcement learning. We propose an object-centric inverse reinforcement learning (IRL) framework that represents each observation as a graph of detected objects and relations, learns a temporally aligned latent space from videos, and defines dense rewards by distance to a goal embedding. A weighted graph pooling mechanism emphasizes task-relevant object dynamics while suppressing robot-dominated motion, encouraging rewards to reflect semantic progress rather than embodiment-specific trajectories. On a structured manipulation benchmark, the learned reward improves downstream RL performance over pixel-based and graph-based baselines, outperforming a hand-designed environment reward. On a longer-horizon task, the learned reward exhibits interpretable stage-wise transitions aligned with manipulation phases, suggesting a path toward automatic subtask discovery. These results support object-centric reward learning as a mechanism for extracting reusable manipulation structure from video and simulation data beyond teleoperation. Although our experiments use simulated demonstrations, the graph-based reward representation is designed to abstract task progress through objects and relations, making it a promising bridge toward future reward learning from human action-free videos.**

I. Introduction

Robot learning pipelines often rely on teleoperated demonstrations because they provide robot-state trajectories and action labels. However, teleoperation also introduces an important bottleneck: the collected behavior is constrained by the robot embodiment, the control interface, operator skill, latency, and limited availability of expert demonstrations. In contrast, action-free videos and simulated rollouts can provide broader coverage of manipulation strategies, object interactions, and failure modes, but they do not directly specify robot actions or rewards.

Object-centric graphs are particularly well suited to this setting because they provide an abstraction layer between visual demonstrations and robot control. A human and a robot may execute a manipulation task with very different kinematics, but the task-relevant structure often remains similar at the level of objects, contacts, containment, and spatial relations. For example, placing shoes into a box can be described by changes in the relations between the shoes and the box, independently of whether the motion was produced by a human hand, a teleoperated robot, or a simulated policy. This makes graph-based rewards a natural representation for extracting manipulation knowledge from human action-free videos, even when robot actions are unavailable.

This paper addresses the following question: *can action-free video demonstrations be converted into dense, task-aligned rewards for robot policy learning?* Rather than imitating demonstration actions, we learn a representation of semantic task progress and use it to guide reinforcement learning (RL). This framing is especially relevant for long-horizon manipulation, where success depends on ordered object interactions and where manually designing rewards or annotating subgoals is expensive.

A key challenge is that raw visual similarity is a poor proxy for manipulation progress. Pixel-level representations [1] are sensitive to appearance, viewpoint, lighting, and embodiment-specific motion. For example, the robot arm may dominate frame-to-frame visual change even when the semantically relevant state is whether a shoe is inside a box. We therefore represent observations as object-centric graphs and learn rewards in the resulting latent space. Objects and their spatial relations provide a compact abstraction that is better aligned with manipulation intent and can support transfer across demonstrations that differ in execution timing or embodiment.

As a first step toward this broader goal, we propose a graph-based reward learning framework for long-horizon manipulation from action-free videos and validate it in simulated robot environments. Demonstrations are converted into graphs of detected objects and pairwise relations. A graph neural network (GNN) encoder is pretrained producing a latent space in which distance to a goal state reflects task progress. While prior work has used graph representations as an abstraction mechanism for reward learning [2], we further exploit their structured nature through a weighted pooling mechanism that emphasizes active task objects while suppressing robot nodes. This prevents embodiment-specific motion from dominating the reward and encourages the latent space to reflect semantic object-level progress. The learned reward can be therefore used directly for RL; in longer-horizon tasks, its temporal profile can also reveal candidate subtask boundaries.

Our contributions are:

- an object-centric reward learning framework that converts action-free visual demonstrations into dense rewards for downstream RL, using graphs as an embodiment-agnostic abstraction of task progress;
- a weighted graph pooling mechanism that focuses the reward on task-relevant object dynamics rather than robot-specific motion;
- empirical evidence that the learned reward improves

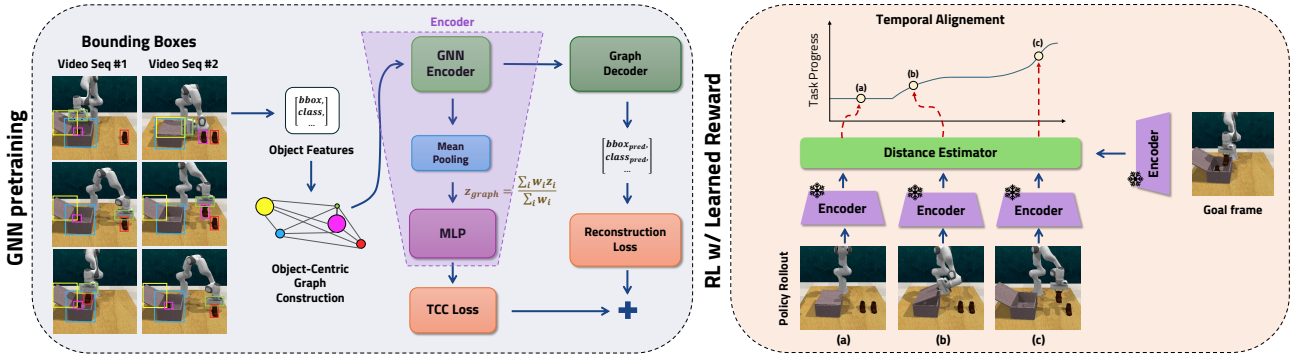

Fig. 1. Overview of the proposed framework. Demonstration videos are converted into object-centric graphs and encoded with a graph neural network trained using temporal cycle-consistency (TCC) [3] and reconstruction losses. The learned encoder is then frozen and used to define a reward for reinforcement 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 subtasks structure.

policy learning and exposes interpretable stage-wise structure in long-horizon manipulation.

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

II. Method

A. Problem Formulation

Let $\mathcal{D} = \{V_k\}_{k=1}^{K}$ denote a set of action-free demonstration videos for a manipulation task, where each video is a sequence of image observations $V_k = \{I_k^1, \dots, I_k^{T_k}\}$. We aim to learn a reward function $r(s)$ that measures progress toward task completion and can be used by an RL agent. Instead of learning from robot actions, we learn an embedding function $\phi(\cdot)$ such that distances in the latent space correspond to semantic task progress.

Given a current observation graph G_t and a goal graph G_g, the reward is

$$r_t = -\|\phi(G_t) - \phi(G_g)\|_2. \tag{1}$$

This reward is dense, action-free, and depends on the object configuration rather than on a specific demonstration controller.

B. Object-Centric Graph Construction

For each frame I_t, we detect objects and construct a fully connected graph $G_t = (V_t, E_t)$. Each node $v_i \in V_t$ corresponds to a detected object and contains semantic and geometric features, including object class and bounding-box coordinates. Each edge $e_{ij} \in E_t$ encodes pairwise spatial relations, such as relative distances between object centers.

This abstraction removes much of the visual variability present in pixels while preserving the entities and relations that define manipulation progress. It also makes the reward less tied to a particular robot embodiment, which is important when using videos or simulation data that may not match the final robot policy domain.

C. Graph Encoder and Weighted Pooling

A GNN encoder processes node and edge features through message passing and produces node embeddings $\{z_i\}_{i=1}^{N}$. These are aggregated into a graph-level embedding using 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

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

where active_i indicates whether object i is dynamically relevant, and robot_i indicates whether the node corresponds to the robot.

Suppressing robot nodes encourages the representation to focus on changes in the manipulated objects and their relations. This is useful for learning rewards from diverse videos because the same task progress may be achieved through different robot motions, human motions, or simulated controllers.

D. Self-Supervised Video Pretraining

The graph encoder is trained using two complementary objectives. First, we use temporal cycle consistency (TCC) [3] to align demonstrations that perform the same task at different speeds or with different execution details. This encourages embeddings from similar task stages to be close even when the videos are not synchronized.

Second, we use an object-level reconstruction objective to preserve scene structure:

$$\mathcal{L}_{\mathrm{rec}} = \|\hat{o}_t - o_t\|_2^2, \tag{4}$$

where o_t denotes object features and $\hat{o}_t = \psi(z_t)$ is the decoder reconstruction. The final training loss is

$$\mathcal{L} = \mathcal{L}_{\mathrm{tcc}} + \lambda \mathcal{L}_{\mathrm{rec}}. \tag{5}$$

After pretraining, the encoder is frozen and used to compute the reward in Eq. 1. The RL policy is then optimized using this learned reward.

TABLE I

FINAL SUCCESS RATE ON MATCH REGIONS. HIGHER IS BETTER.

Method	Success Rate (%) ↑
Environment reward	85.0
XIRL	0.0
GraphIRL w/ robot	0.0
GraphIRL w/o robot	33.7
Ours w/ robot	33.3
Ours w/o robot	**94.6**

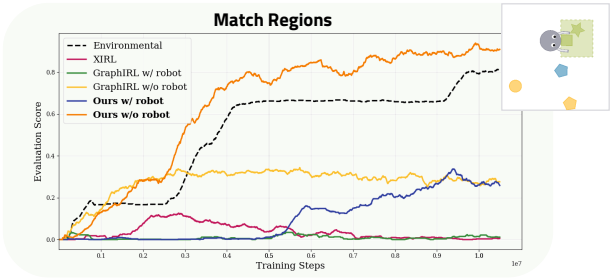

Fig. 2. Results on *Match Regions*. Our object-centric GNN reward achieves the highest final success rate and fastest convergence. Suppressing robot nodes during pooling reaches 94.6% success, outperforming the hand-crafted environment reward at about 85%.

TABLE II

REWARD DISCRIMINATION ON SHOES IN BOX. LOWER IS BETTER.

Method	Positive/Negative Ratio ρ ↓
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

Fig. 3. Learned reward profile for *Shoes in Box*. The reward exhibits step-like transitions aligned with semantic task phases, such as opening the box and placing each shoe, suggesting candidate boundaries for long-horizon subtask discovery.

III. EXPERIMENTS

We evaluate the framework on two manipulation tasks. The first tests whether the learned reward improves downstream RL. The second tests whether the reward captures stage-wise progress in a longer-horizon task.

A. Match Regions

Task. *Match Regions* is a structured manipulation task from the MAGICAL benchmark [4] in which the robot must place objects into their corresponding target regions. The task requires reasoning about object placement and spatial configuration, but can be solved with a single full-task reward.

Baselines. We compare against a pixel-based inverse RL method, XIRL [1], a graph-based baseline, GraphIRL [2] and the hand-designed environment reward provided by the simulator. We also compare variants with and without robot nodes during pooling.

Results. Table I reports final success rates. Our reward with robot nodes suppressed reaches the highest performance, achieving 94.6% success. It outperforms the hand-designed environment reward, which reaches 85.0%, despite using only demonstration videos for reward learning. The performance gap, also depicted in Fig. 2, is especially important because the environment reward uses privileged task information, while our method learns the reward from action-free object-centric video structure.

These results suggest that object-centric latent rewards provide a more informative optimization signal than pixel similarity and can even improve over manually engineered rewards when the learned representation captures task-relevant object relations.

B. Shoes in Box

Task. *Shoes in Box* is a longer-horizon manipulation task from RLBench [5]. The task requires reaching the box, opening it, reaching a shoe, moving it into the box, and repeating the procedure for the second shoe. This sequential structure makes it useful for studying whether the learned reward reflects meaningful task phases.

Reward structure. The learned reward, depicted in Fig. 3, exhibits a step-like temporal profile, with transitions aligned to semantic events such as opening the box and placing each shoe. This indicates that the representation captures more than smooth visual proximity to the final frame: it tracks discrete changes in the object-centric scene configuration. Such structure can be used as a heuristic signal for automatic subtask discovery.

Discrimination between successful and unsuccessful rollouts. We evaluate whether the learned reward separates successful and unsuccessful validation trajectories. Let $\bar{R}_+$ be the mean cumulative reward for successful trajectories and $\bar{R}_-$ the mean cumulative reward for unsuccessful trajectories. We compute

$$\rho = \frac{\bar{R}_+}{\bar{R}_-}. \tag{6}$$

Since rewards are negative distances, lower values indicate stronger separation between successful and unsuccessful behavior. Table II shows that our method achieves the two lowest ratios among the compared methods, indicating stronger reward discrimination.

Together, the qualitative reward profile and quantitative separation results show that the learned object-centric reward captures long-horizon task progress and can reveal candidate subtask boundaries without manual temporal annotation.

IV. Discussion: Connection to Beyond-Teleoperation Learning

The proposed framework is motivated by the limitations of teleoperation-centric robot learning. Teleoperation provides robot actions, but it also ties supervision to a specific embodiment, control interface, and operator. Our approach instead treats action-free videos as a source of task-progress information. The learned reward does not imitate the demonstrator's actions; it measures whether the current scene is moving toward the demonstrated goal configuration.

The graph structure is central to this beyond-teleoperation perspective. By representing observations through objects and their relations, the reward model abstracts away from embodiment-specific motion and focuses on manipulation-relevant state changes. This is precisely the level at which human action-free videos may become useful: a human hand, a robot gripper, and a simulated controller may generate very different trajectories, but they can induce similar relational changes among task objects. Thus, although the present experiments use simulated videos, the same reward-learning mechanism is designed to support future extraction of manipulation knowledge from human videos without requiring action labels, teleoperation trajectories, or manual subtask annotations.

The current experiments validate this idea in simulated manipulation tasks. A key next step is to use the same object-centric reward model with more diverse human and robot video data, including in-the-wild demonstrations and small amounts of real-world teleoperated data used only as grounding. Another direction is to exploit the reward transitions observed in long-horizon tasks to automatically segment demonstrations, learn subtask-specific rewards, and train sequential subpolicies.

V. Conclusion

We presented an object-centric reward learning framework for long-horizon manipulation from action-free video demonstrations. By representing scenes as graphs and learning temporally aligned latent embeddings, the method produces dense rewards that support downstream RL without hand-designed reward functions. A weighted pooling mechanism suppresses robot-dominated motion and emphasizes task-relevant object dynamics, improving policy learning and making the reward more semantically interpretable. Experiments show that the learned reward outperforms pixel-based and graph-based baselines on a complex manipulation task and reveals meaningful stage-wise structure on long-horizon tasks. These results suggest that object-centric reward learning is a promising direction for converting diverse video and simulation data into robot learning signals beyond teleoperation.

APPENDIX

A. Implementation Details

The activity variable in Eq. 3 is computed from temporal displacement of bounding-box centers:

$$\text{active}_i = \mathbb{I}\left(\frac{\|c_t - c_{t-k}\|}{\text{diag}_i} > \tau\right), \tag{7}$$

where c_t is the bounding-box center, diag_i is the bounding-box diagonal, and τ is a threshold. Normalizing by the box diagonal makes the activity criterion approximately scale invariant. In our experiments, we use $\tau = 0.005$ and temporal smoothing with $k = 40$ frames to reduce sensitivity to detector jitter and simulator spikes.

Once an object becomes active, its activity bit remains active for the rest of the episode. This persistence encourages a monotonic notion of semantic relevance, since an object that has already been manipulated should remain part of the task state even after it stops moving.

B. Automatic Subtask Discovery

As shown in Sec. III-B, the learned reward often evolves through stage-wise transitions rather than changing smoothly at every frame. Given this property, we can identify candidate subtask boundaries by evaluating the reward over a demonstration trajectory and detecting significant changes in the reward gradient. These transition points can segment demonstrations into subtask-specific datasets.

In the current paper, we validate the prerequisite for this pipeline: the learned reward exhibits interpretable transitions aligned with manipulation phases. Full quantitative evaluation of segmentation quality, subtask-specific reward learning, and sequential subpolicy composition is left for future work.

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.