

XPG-RL: Reinforcement Learning with Explainable Priority Guidance for Efficiency-Boosted Mechanical Search

Yiting Zhang^{1†}, Shichen Li², and Elena Shrestha¹

¹University of Michigan, Ann Arbor ²University of Illinois, Urbana-Champaign

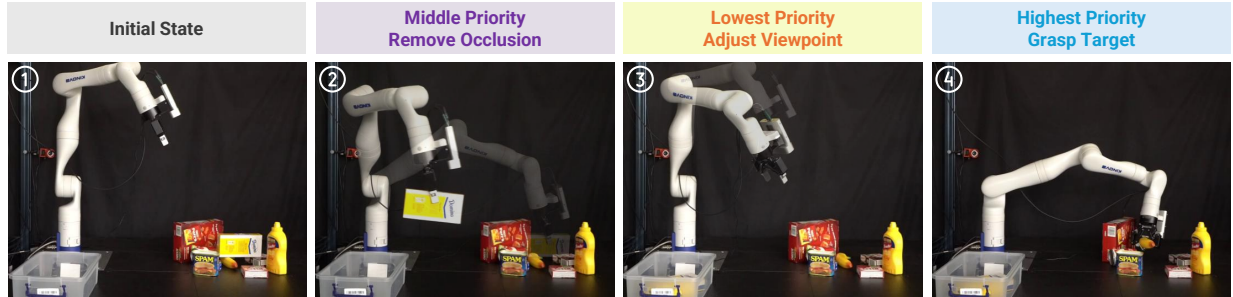


Fig. 1. We propose XPG-RL, a reinforcement learning framework for mechanical search tasks. XPG-RL leverages task-guided action prioritization and learns context-aware switching over action primitives, effectively reducing redundant manipulations and improving task efficiency. The figure shows the manipulator successfully grasping a target object (*banana*) in a densely cluttered real-world scene.

Abstract—Mechanical search (MS) in cluttered environments remains a significant challenge for autonomous manipulators, requiring long-horizon planning and robust state estimation under occlusions and partial observability. In this work, we introduce XPG-RL, a reinforcement learning framework that enables agents to efficiently perform MS tasks through explainable, priority-guided decision-making based on raw sensory inputs. XPG-RL integrates a task-driven action prioritization mechanism with a learned context-aware switching strategy that dynamically selects from a discrete set of action primitives such as target grasping, occlusion removal, and viewpoint adjustment. Within this strategy, a policy is optimized to output adaptive threshold values that govern the discrete selection among action primitives. The perception module fuses RGB-D inputs with semantic and geometric features to produce a structured scene representation for downstream decision-making. Extensive experiments in both simulation and real-world settings demonstrate that XPG-RL consistently outperforms baseline methods in task success rates and motion efficiency, achieving up to $4.5\times$ higher efficiency in long-horizon tasks. These results underscore the benefits of integrating domain knowledge with learnable decision-making policies for robust and efficient robotic manipulation. The project page for XPG-RL is <https://yitingzhang1997.github.io/xpgtrl/>.

I. INTRODUCTION

Mechanical search (MS) is a critical yet challenging task in robotic manipulation. It requires robots to intelligently plan and execute sequences of physical interactions to locate and retrieve occluded objects from cluttered environments [3]. Unlike traditional pick-and-place tasks, MS involves occlusion, long-horizon planning, and environmental uncertainty, where robots must infer object locations, plan sequential actions, and adapt to unpredictable dynamics [16, 21]. These challenges

hinder real-world applications such as logistics automation, warehouse management, and domestic assistance [4, 13], highlighting the need for more efficient and reliable MS methods.

Given the dynamic and unpredictable nature of MS tasks, predefining all possible scenarios is impractical. Learning-based approaches are essential for allowing robots to adapt to new situations and generalize beyond predefined rules [2, 38]. Reinforcement learning (RL) enables robots to learn policies directly from visual inputs through trial-and-error exploration, without requiring explicit object pose knowledge, and is widely recognized as an effective approach for tackling MS tasks [34, 12, 18, 29]. However, conventional RL methods treat each action primitive independently via separate neural networks, neglecting action interdependencies [7, 28]. This limitation leads to cascading errors and can misguide decision-making in scenarios requiring long-horizon planning. Zenkri et al. observed that when the number of objects increased from 10 to 20, the efficiency of traditional RL methods dropped by 46.6% [36]. These challenges highlight the need for advancements in RL to enhance long-term MS performance and improve decision-making efficiency.

To enhance interactions between actions in long-horizon planning, recent studies have explored hierarchical reinforcement learning (HRL) for complex MS tasks [36, 31]. HRL introduces an additional network on top of standard RL structures for high-level action selection, enabling the learning of action interdependencies across sequences [30]. However, MS training typically relies on limited datasets, which constraints HRL’s capacity to learn effectively. In short-term tasks, this limitation often result in over-analyzing procedural steps due to insufficient data coverage. In more complex,

[†]Corresponding author. Email: yitizhang@umich.edu.

long-horizon tasks, HRL further struggles with embedding and feature extraction, as learning abstract high-level representations implicitly from sparse data becomes increasingly challenging. Research has shown that HRL often lacks the granularity to capture subtle environmental variations [25], leading to suboptimal action selection, inefficient exploration, and compounding errors over extended task horizons. Thus, HRL remains limited in data-inefficient MS tasks.

The primary challenge in improving MS efficiency lies in fully utilizing high-dimensional, limited, and hard-to-distinguish data [10]. Addressing this requires advancements in both model design and data enrichment. On one hand, enhancing model architectures can reduce the complexity of learning tasks. By alleviating the training burden from raw, complex inputs, models can better focus on essential features, improving task success rates [19]. In MS tasks, incorporating predefined guidance into the model design can streamline learning by directing attention to critical information. On the other hand, enriching the dataset is important. In robot learning, a key difficulty is capturing relevant targets during in-situ image collection. Instead of relying solely on fixed viewpoints and actions within a limited knowledge scope, increasing information sources can significantly improve efficiency by providing more comprehensive contextual understanding [17].

The main contributions of this work are four-fold:

- We propose a task-guided action prioritization mechanism that embeds domain knowledge into a structured hierarchy over manipulation primitives. This promotes modular, explainable high-level policy learning and reduces learning complexity in long-horizon tasks.
- We present XPG-RL, a novel RL framework that leverages the prioritization structure to learn context-aware switching between action primitives. This approach minimizes redundant or misaligned actions and improves overall task efficiency.
- We integrate a next-best-view (NBV) strategy, enabling the robot to actively select informative viewpoints. This improves perception in occluded scenes and allows for spatial understanding beyond what is possible from a fixed viewpoint.
- We validate our approach in both simulated and real-world cluttered environments, demonstrating that XPG-RL consistently outperforms strong baselines in task success rate and motion efficiency.

II. RELATED WORKS

Visual-input-only MS. Recent research has explored RL for directly learning MS policies from visual inputs, without relying on explicit object pose estimation. One of the most influential frameworks in this space is Visual Pushing Grasping (VPG) [34], which learns separate quality maps for pushing and grasping primitives and selects actions by choosing the pixel with the highest predicted quality. Numerous follow-up works have refined this paradigm by improving network architectures [11, 7, 28]. To improve high-level decision-making, HRL has been explored to better capture the inter-

actions among action primitives, especially in long-horizon tasks [22, 27, 14, 33]. Our work is inspired by the HRL structure while focused on the high-level decision-making.

Priority-guided decision-making in robot manipulation.

In robot manipulation, integrating task or action priorities has long been a common strategy for managing complex decision-making under constraints. Traditional methods typically rely on fixed priority rules or cost-based heuristics to sequence actions [24, 8, 5]. More recent work has introduced learning-based frameworks that incorporate human priors or domain-specific logic to guide high-level decisions [35, 15]. However, these approaches still depend on manually designed, experience-driven heuristics at the high level, which limits their adaptability. In long-horizon or highly cluttered tasks, where scene configurations can vary widely, such static logic often fails to generalize or respond effectively to unseen situations.

III. METHOD

Our MS task formulation is based on a framework [36], where the objective is to retrieve an occluded target object from a cluttered scene using a manipulator equipped with a parallel-jaw gripper. The robot perceives the scene through an eye-in-hand RGB-D camera, but the initial viewpoint is often constrained by occlusions. To address this challenge, we propose XPG-RL (**R**einforcement **L**earning with **eX**plainable **P**riority **G**uidance). The method comprises two main components: (1) a perception pipeline and, (2) an RL-based decision-making module (Fig. 2).

A. Perception

The perception pipeline receives fused RGB-D images captured from the working scene. The RGB image provides visual appearance cues, such as surface property and material type. We leverage the Segment Anything Model (SAM) [9] to generate appearance-based masks for all objects, and integrate it with a PoseCNN [32] model to add semantic context, producing a mask focused on the specified target object.

The depth image captures spatial geometry useful for understanding occlusion, volume, and spatial relationships. However, raw depth data often varies across different surfaces of the same object, making it sub-optimal for object-level reasoning. To address this, we propose an Object Depth Mapping (ODM) module to generate a simplified, object-centric spatial representation. Specifically, the raw depth image is fused with object masks from the RGB image. Instead of relying on pixel-wise depth, we compute the average depth for each object. Objects are then sorted by distance, from farthest to nearest, and rendered in progressively lighter shades to generate the final object-level depth map. The target-specific mask and the object-level depth map are jointly provided as input to the RL agent, enabling efficient environmental interpretation and informed decision-making.

B. RL-based Decision-making

We formulate the decision-making process as a partially observable Markov decision process (POMDP). At each time

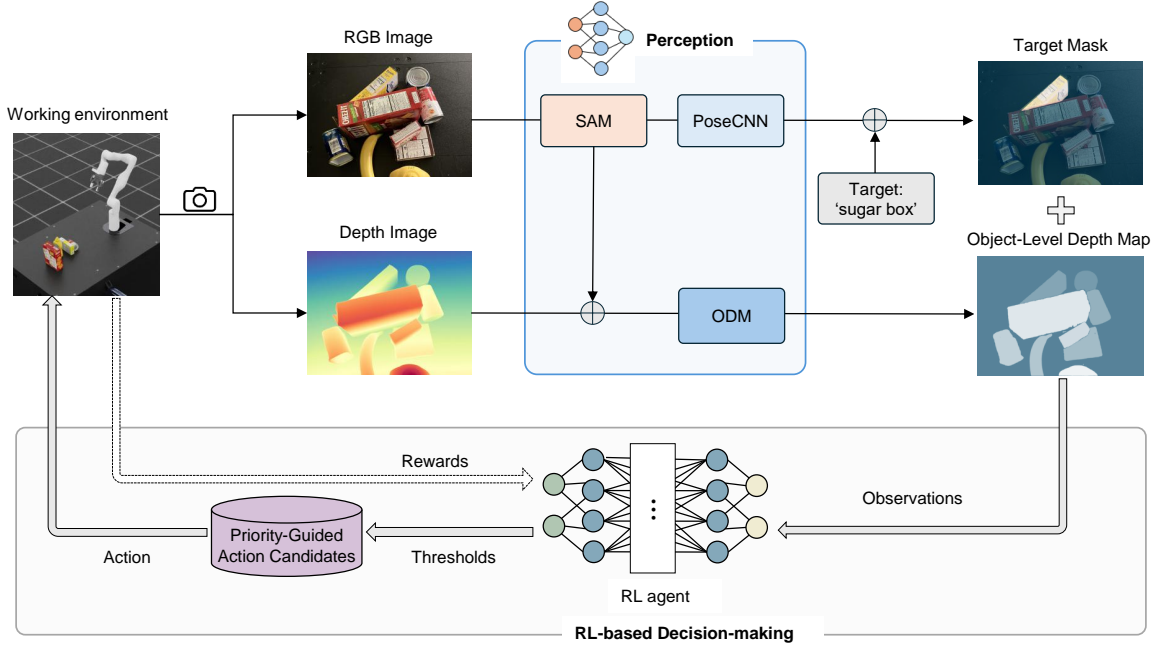


Fig. 2. **Overview of the XPG-RL framework.** The framework consists of two main components: (1) a perception pipeline, which processes fused RGB-D images to extract semantic and geometric context and builds a compact scene representation; and (2) an RL-based decision-making module, which takes this representation as input and learns a policy to predict adaptive thresholds. These thresholds guide the selection among priority-structured action candidates (e.g., target grasping, occlusion removal, and viewpoint adjustment), enabling context-aware and efficient action execution.

step t , the perception pipeline outputs an observation o_t , which provides partial information on the true underlying environment state s_t . The agent follows a policy $\pi(\tau_t | o_t)$, which maps the observation o_t to a set of decision thresholds $\tau_t = (\tau_{1,t}, \tau_{2,t})$, where $\tau_{1,t}, \tau_{2,t} \in [0, 1]$. To inform action selection, the system predicts quality scores $Q_{\text{target},t}$ and $Q_{\text{occlude},t}$ using AnyGrasp [6], representing the estimated success probabilities of grasping the target object and removing an occluding object, respectively. These thresholds τ_t are applied to the predicted scores to determine which action $a_t \in \mathcal{A} = \{a_1, a_2, \dots, a_n\}$ to execute from a set of priority-guided action candidates, such as grasping the target or removing an occluding object. The priority-guided action selection process is detailed in Algorithm 1. After executing the selected action, the environment transitions to a new state s_{t+1} , and the agent receives an immediate reward r_t . The reward function encourages successful grasps while penalizing unnecessary or infeasible actions, and is defined as the following:

$$r_t = \begin{cases} 1000, & \text{if the target is successfully extracted;} \\ -100, & \text{if an infeasible action is selected;} \\ -1, & \text{otherwise.} \end{cases} \quad (1)$$

The goal is to maximize the expected cumulative discounted reward $R_t = \sum_{i=0}^{\infty} \gamma^i r_{t+i}$ where $\gamma \in [0, 1]$ is the discount factor that balances the importance of future versus immediate rewards.

Algorithm 1 Priority-Guided Decision Policy

Require: Thresholds $\tau_t = (\tau_{1,t}, \tau_{2,t})$
 Compute grasp score $Q_{\text{target},t}$
if $Q_{\text{target},t} > \tau_{1,t}$ **then**
 Grasp the target object
else
 Compute $Q_{\text{occlude},t}$
 if $Q_{\text{occlude},t} > \tau_{2,t}$ **then**
 Grasp and remove an occluding object
 else
 Move camera to new viewpoint
 end if
end if

To achieve this objective, we employ Proximal Policy Optimization (PPO) [26] to optimize the policy $\pi(\tau_t | o_t)$, which determines continuous thresholds τ_t that guide action selection. This approach allows for flexible decision boundaries over predicted success probabilities, supporting effective learning under partial observability.

Furthermore, to enhance sample efficiency and promote robust behavior, we constrain the agent’s action space using a domain-informed set of priority-guided candidates (Fig. 3). These structured discrete actions, ranked by priority, simplify the decision-making process and ensure reliable performance, even with limited training data. For the MS task, we propose the following three priority-guided action candidates:

- **Highest priority:** If the grasp quality for the target object is high, i.e., $Q_{\text{target},t} \geq \tau_{1,t}$, the agent directly executes the

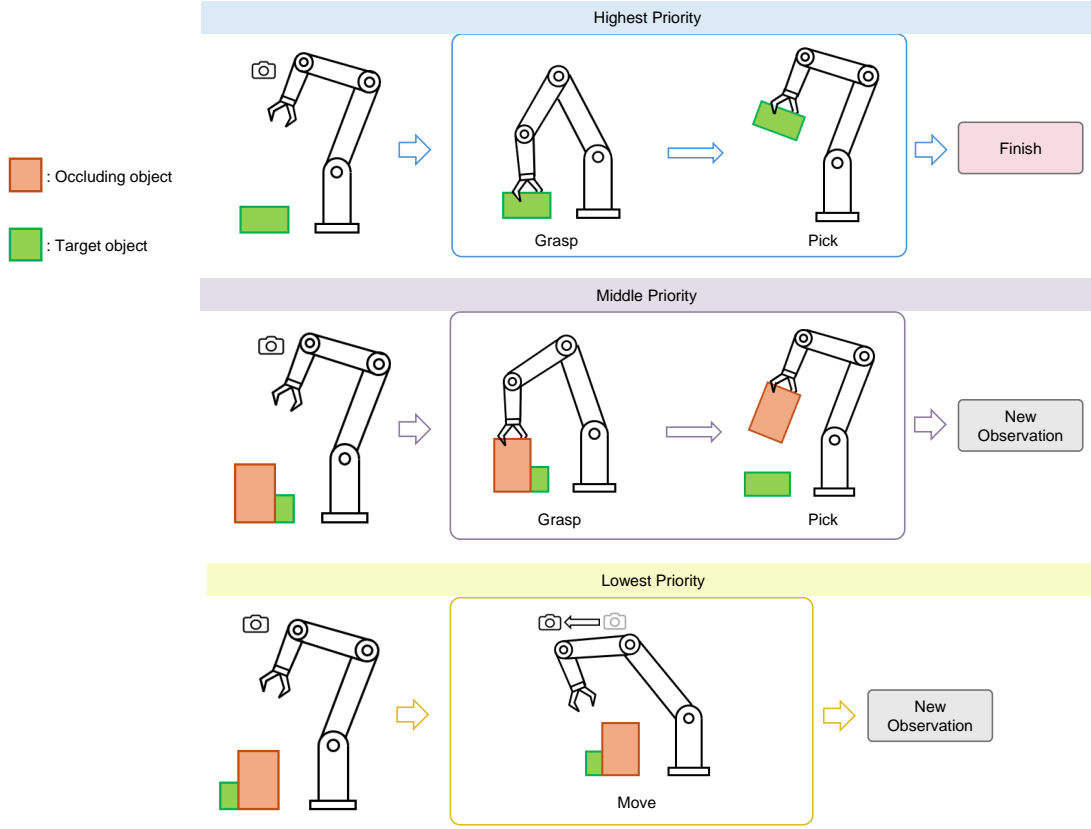


Fig. 3. **Priority-guided action candidates.** The action space consists of three discrete primitives—target grasping, occlusion removal, and viewpoint adjustment—ranked from highest to lowest priority. Learned thresholds govern switching between these actions, enabling the agent to make efficient, interpretable decisions by sequentially evaluating actions in priority order.

grasp with the highest quality score on the target object. This action is the final execution of one MS task.

- **Middle priority:** If the target grasp quality is low but the quality for occluding objects is high, i.e., $Q_{\text{target},t} < \tau_{1,t}$ and $Q_{\text{occlude},t} \geq \tau_{2,t}$, the agent selects the occluding object with the highest score for removal. After the action is executed, the agent re-evaluates whether the condition for the highest-priority action is now satisfied.
- **Lowest priority:** If both the target and occluding objects have low grasp quality, i.e., $Q_{\text{target},t} < \tau_{1,t}$ and $Q_{\text{occlude},t} < \tau_{2,t}$, the agent initiates a next-best-view (NBV) strategy where it generates a set of novel candidate viewpoints. It constructs a Truncated Signed Distance Function (TSDF) representation of the current scene, simulates each candidate view, predicts the grasp quality for the target, and selects the viewpoint with the highest predicted quality as the next camera pose.

IV. EXPERIMENTS

We evaluate the performance of XPG-RL in both simulation and real-world environments. We use a 7-DoF Kinova robot arm equipped with a RealSense camera and a parallel jaw ROBOTIQ gripper at the end effector. The objects used in the experiments are selected from the YCB object set [1]. We train the RL policy using the PPO algorithm as implemented in Stable-Baselines3 [23]. Further details on the training

hyperparameters and network architecture are provided in the appendix.

Metrics. In our experiments, we evaluate the performance of our method using the following metrics: **Task success rate** is defined as the percentage of episodes in which the target object is successfully grasped and lifted at least 15 cm above the table within 10 motions. **Average motion number** refers to the mean quantity of actions needed to achieve the task per episode. **Task efficiency** is defined as the ratio between task success rate and average motion number, reflecting how effectively a method achieves successful outcomes using minimal actions.

Baselines. We compare the performance of our method with the following baselines. **Target-Oriented VPG** is a traditional RL method that extends the original VPG [34] by incorporating the target mask as an additional input. It outputs quality maps for pushing and grasping, executing the action with the highest quality score. **Hierarchical Policy Learning** [36] is an HRL approach that uses a high-level policy network on top of traditional RL architectures to model action interdependencies across sequences. **MPGNet** [15] builds upon Target-Oriented VPG by jointly learning three actions: (1) moving, (2) pushing, and (3) grasping. It adopts a hierarchical structure and incorporates a simple heuristic-based logic for high-level action selection.

TABLE I: **Success rate and efficiency across scenes.** XPG-RL consistently achieves higher success rates with fewer actions than baselines, demonstrating robust and efficient performance.

Method	Task success rate [%]				Average motion number			
	5 objects	10 objects	15 objects	20 objects	5 objects	10 objects	15 objects	20 objects
Target-Oriented VPG	60	30	26	15	4.92	5.38	7.85	8.01
Hierarchical Policy Learning	57	53	30	21	3.91	4.06	5.18	7.45
MPGNet	76	59	35	22	3.84	4.51	6.78	7.65
XPG-RL	81	76	71	64	2.31	2.77	3.60	4.94

A. Simulation Experiments

The task is simulated in Isaac Lab [20], which provides realistic dynamic modeling capabilities. We generate batches of simulated scenes with varying levels of complexity. Each batch consists of 100 scenes populated with n objects randomly selected from the YCB object set and distributed on a table within a $0.5 \text{ m} \times 0.5 \text{ m}$ square area.

1) *Comparison to Baselines:* We evaluate the performance of XPG-RL against several baseline methods across a range of scene complexities, defined by the number of objects in the environment: from 5 (simple cases) to 20 (highly cluttered cases). As object count increases, the scene becomes more occluded and challenging. Table I summarizes the task success rate and average motion number for each method.

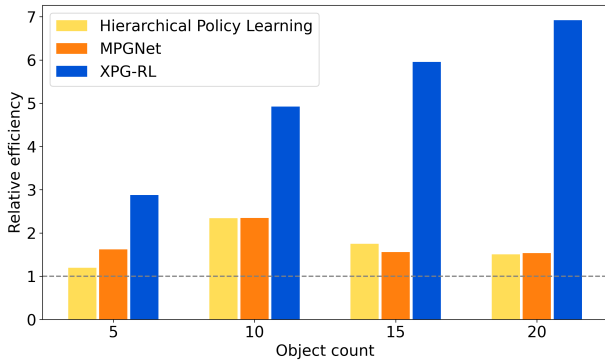


Fig. 4. **Efficiency scaling with complexity.** XPG-RL outperforms baselines across all clutter levels, with relative efficiency gains widening as object count increases.

Across all methods, we observe a general trend: the task success rate decreases and the average motion number increases as scene complexity grows. Some notable patterns emerge. In relatively simple scenarios (e.g., 5 objects), Hierarchical Policy Learning tends to perform the worst. This is due to its multi-level decision-making structure, which can trigger unnecessary actions instead of a direct grasp. Target-Oriented VPG also struggles as complexity increases, highlighting its limited ability to coordinate multiple action primitives effectively. MPGNet benefits from its heuristic-driven high-level selection and performs reasonably well across general cases. However, its predefined logic lacks adaptability—it cannot dynamically adjust behavior based on task simplicity, nor flexibly respond to increasing scene complexity. As a result, it often triggers unnecessary actions in simple tasks and struggles with efficiency as complexity grows.

We further evaluate task efficiency by computing the relative efficiency of each method, using Target-Oriented VPG as the reference baseline. As shown in Figure 4, XPG-RL consistently demonstrates the highest efficiency across all scenarios. Notably, the efficiency gap between XPG-RL and the second-best method widens as episode length increases. In the 10-object case, XPG-RL achieves a relative efficiency of 5.0, while the baselines remain below 2.5. In the 20-object case—the most complex scenario—XPG-RL reaches 6.9, compared to 1.5 for both MPGNet and Hierarchical Policy Learning, yielding over a $4.5\times$ improvement. This trend highlights XPG-RL’s stability and effectiveness in long-horizon tasks, whereas other methods tend to degrade in performance as complexity grows.

To further investigate whether XPG-RL retains its advantage in short-horizon tasks, we conduct an additional evaluation on scenes with 3 to 5 objects. The results are presented in Table IV in the appendix. While both XPG-RL and MPGNet perform reasonably well in terms of task success, XPG-RL completes tasks with significantly fewer motions. This efficiency gain further underscores the benefit of learning-based high-level control, which enables XPG-RL to adapt its behavior based on task complexity and avoid redundant actions when a simple solution suffices.

These results highlight the effectiveness of XPG-RL’s design, which combines structured action prioritization with a learned high-level decision-making policy. By integrating a neural network to make context-aware decisions over a prioritized set of action primitives, XPG-RL achieves robust and efficient performance across both short- and long-horizon tasks, consistently outperforming all baseline methods.

2) *Ablation Study:* To assess the contribution of each key component in XPG-RL, we evaluate two ablated variants: (1) **No Priority Guidance** and (2) **No NBV Motion**. In the **No Priority Guidance** variant, the RL agent no longer uses the action priority scheme. Instead of learning threshold-based switching decisions among primitives, it directly learns a flat policy to select actions. The **No NBV Motion** variant retains the priority-guided decision-making policy but disables the NBV motion. If neither the target object nor the occluding object has a feasible grasp, the trial is considered a failure. This configuration isolates the importance of NBV in improving visibility and grasp success in occluded scenes.

The results, summarized in Table II, highlight the contribution of each module. Removing the priority guidance leads to a clear and consistent drop in both task success rate and motion efficiency, especially as scene complexity increases.

TABLE II: **Contribution of priority guidance and NBV.** Removing priority guidance or NBV leads to lower success rates and efficiency, especially as scene complexity increases. This confirms that both components are critical for robust performance in cluttered environments.

Method	Task success rate [%]				Average motion number			
	5 objects	10 objects	15 objects	20 objects	5 objects	10 objects	15 objects	20 objects
No Priority Guidance	64	57	52	42	4.87	5.28	5.75	6.24
No NBV Motion	78	65	62	51	2.27	2.87	3.27	3.93
XPG-RL	81	76	71	64	2.31	2.77	3.60	4.94

TABLE III: **Real-world performance across cluttered scenes.** XPG-RL consistently achieves higher task success rates with fewer actions than baselines across four tabletop environments.

Method	Task success rate				Average motion number			
	scene 1	scene 2	scene 3	scene 4	scene 1	scene 2	scene 3	scene 4
Hierarchical Policy Learning	2/5	3/5	3/5	0/5	6.4	3.8	5.0	5.8
MPGNet	3/5	4/5	3/5	2/5	3.4	4.8	4.4	5.2
XPG-RL	4/5	5/5	5/5	4/5	2.8	2.0	3.2	3.4

This demonstrates that the structured action prioritization significantly improves decision quality, particularly in longer-horizon tasks. When comparing the full XPG-RL model to the No NBV Motion variant, we observe that NBV selection plays a relatively minor role in simpler scenes (e.g., 5 objects) but becomes increasingly important in complex and highly occluded environments. In such cases, NBV helps the agent improve visibility and locate graspable views of the target object, resulting in higher success rates and fewer failed attempts. Overall, the ablation study confirms that both priority guidance and NBV selection are essential for achieving robust, efficient performance, with their contributions becoming more pronounced as task complexity increases.

B. Real-World Experiments

We further evaluate the effectiveness and robustness of XPG-RL in real-world cluttered environments across four different tabletop scenes, each containing 9 randomly selected objects that are densely arranged on the table. The manipulator is positioned such that the target object is either fully or partially occluded in the initial camera view. To account for variability in initial object placement and manipulator positioning, each scene is repeated 5 times. The environment setup, initial view, and target object mask for each scene are shown in Fig. 5. Notably, in Scene 3, the target object (*meat can*) is completely occluded from the initial camera view, and thus no target pixels are visible in the mask.

The results are summarized in Table III. Across all scenes, XPG-RL consistently achieves higher task success rates and requires fewer motions compared to baseline methods. In Scene 3, where the target object is fully occluded, XPG-RL effectively leverages NBV motions to reposition the camera, enabling successful identification and grasping. Rather than disturbing surrounding objects and worsening occlusion, it strategically selects a side viewpoint to minimize risk and maintain task feasibility. In Scene 4, the target object (*banana*) is partially occluded by a *sugar box*, where naive pushing or failed grasps could destabilize the scene and obscure the

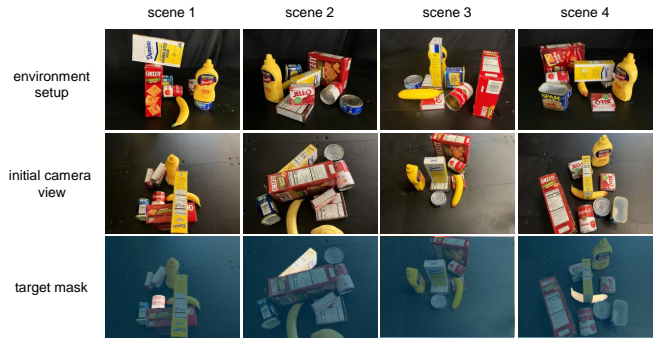


Fig. 5. **Real-world experiment scenes.** Four tabletop setups with 9 objects each, illustrating different occlusion conditions, including a fully occluded target in Scene 3.

target further. XPG-RL avoids such failures through priority-guided decision-making that defers risky actions when predicted success probabilities fall below learned thresholds. This mechanism enables the agent to prioritize safer alternatives, such as NBV motions, and supports more informed, interpretable decisions. By learning context-aware thresholds, XPG-RL dynamically balances risk and reward based on the scene context, demonstrating strong generalization and reliable performance in real-world cluttered environments.

V. CONCLUSION

We propose XPG-RL, a reinforcement learning framework that enhances efficiency and robustness in mechanical search through structured action prioritization and adaptive action switching. XPG-RL consistently outperformed state-of-the-art methods in both simulation and real-world environments. In simulated cluttered scenes, XPG-RL demonstrated up to a $4.5\times$ improvement in relative efficiency. In real-world experiments involving fully occluded scenes, XPG-RL completed all five trials with fewer actions, whereas state-of-the-art methods frequently failed. These results underscore the benefits of integrating domain knowledge with learnable decision-making policies for robust and efficient robotic manipulation.

REFERENCES

- [1] Berk Calli, Aaron Walsman, Arjun Singh, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. Benchmarking in manipulation research: The ycb object and model set and benchmarking protocols. *arXiv preprint arXiv:1502.03143*, 2015.
- [2] Yizhou Chen, Yiting Zhang, Zachary Brei, Tiancheng Zhang, Yuzhen Chen, Julie Wu, and Ram Vasudevan. Differentiable discrete elastic rods for real-time modeling of deformable linear objects. *arXiv preprint arXiv:2406.05931*, 2024.
- [3] Michael Danielczuk, Andrey Kurenkov, Ashwin Balakrishna, Matthew Matl, David Wang, Roberto Martín-Martín, Animesh Garg, Silvio Savarese, and Ken Goldberg. Mechanical search: Multi-step retrieval of a target object occluded by clutter. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 1614–1621. IEEE, 2019.
- [4] Salvatore D’Avella, Matteo Bianchi, Ashok M Sundaram, Carlo Alberto Avizzano, Máximo A Roa, and Paolo Tripicchio. The cluttered environment picking benchmark (cepb) for advanced warehouse automation: evaluating the perception, planning, control, and grasping of manipulation systems. *IEEE Robotics & Automation Magazine*, 2023.
- [5] Mehmet R Dogar and Siddhartha S Srinivasa. A planning framework for non-prehensile manipulation under clutter and uncertainty. *Autonomous Robots*, 33:217–236, 2012.
- [6] Hao-Shu Fang, Chenxi Wang, Hongjie Fang, Minghao Gou, Jirong Liu, Hengxu Yan, Wenhai Liu, Yichen Xie, and Cewu Lu. Anygrasp: Robust and efficient grasp perception in spatial and temporal domains. *IEEE Transactions on Robotics*, 2023.
- [7] Muhammad Babar Imtiaz, Yuansong Qiao, and Brian Lee. Prehensile and non-prehensile robotic pick-and-place of objects in clutter using deep reinforcement learning. *Sensors*, 23(3):1513, 2023.
- [8] Leslie Pack Kaelbling and Tomás Lozano-Pérez. Hierarchical task and motion planning in the now. In *2011 IEEE International Conference on Robotics and Automation*, pages 1470–1477. IEEE, 2011.
- [9] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.
- [10] Oliver Kroemer, Scott Niekum, and George Konidaris. A review of robot learning for manipulation: Challenges, representations, and algorithms. *Journal of machine learning research*, 22(30):1–82, 2021.
- [11] Sulabh Kumra, Shirin Joshi, and Ferat Sahin. Learning multi-step robotic manipulation policies from visual observation of scene and q-value predictions of previous action. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 8245–8251. IEEE, 2022.
- [12] Andrey Kurenkov, Joseph Taglic, Rohun Kulkarni, Marcus Dominguez-Kuhne, Animesh Garg, Roberto Martín-Martín, and Silvio Savarese. Visuomotor mechanical search: Learning to retrieve target objects in clutter. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8408–8414. IEEE, 2020.
- [13] Andrey Kurenkov, Roberto Martín-Martín, Jeff Ichnowski, Ken Goldberg, and Silvio Savarese. Semantic and geometric modeling with neural message passing in 3d scene graphs for hierarchical mechanical search. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11227–11233. IEEE, 2021.
- [14] Jeong-Hoon Lee and Jongeun Choi. Hierarchical primitive composition: Simultaneous activation of skills with inconsistent action dimensions in multiple hierarchies. *IEEE Robotics and Automation Letters*, 7(3):7581–7588, 2022.
- [15] Dayou Li, Chenkun Zhao, Shuo Yang, Ran Song, Xiaolei Li, and Wei Zhang. Mpgnet: Learning move-push-grasping synergy for target-oriented grasping in occluded scenes. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5064–5071. IEEE, 2024.
- [16] Richard Li, Allan Jabri, Trevor Darrell, and Pulkit Agrawal. Towards practical multi-object manipulation using relational reinforcement learning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4051–4058. IEEE, 2020.
- [17] Shichen Li and Chenhui Shao. Multi-modal fusion of in-situ video data and process parameters for online forecasting of cookie drying readiness. *arXiv preprint arXiv:2504.15599*, 2025.
- [18] Gaoyuan Liu, Joris De Winter, Denis Steckelmacher, Roshan Kumar Hota, Ann Nowe, and Bram Vanderborght. Synergistic task and motion planning with reinforcement learning-based non-prehensile actions. *IEEE Robotics and Automation Letters*, 8(5):2764–2771, 2023.
- [19] Yuquan Meng, Zhiqiao Dong, Kuan-Chieh Lu, Shichen Li, and Chenhui Shao. Meta-learning-based domain generalization for cost-effective tool condition monitoring in ultrasonic metal welding. *IEEE Transactions on Industrial Informatics*, 2024.
- [20] Mayank Mittal, Calvin Yu, Qinxi Yu, Jingzhou Liu, Nikita Rudin, David Hoeller, Jia Lin Yuan, Ritvik Singh, Yunrong Guo, Hammad Mazhar, Ajay Mandlekar, Buck Babich, Gavriel State, Marco Hutter, and Animesh Garg. Orbit: A unified simulation framework for interactive robot learning environments. *IEEE Robotics and Automation Letters*, 8(6):3740–3747, 2023. doi: 10.1109/LRA.2023.3270034.
- [21] Changjoo Nam, Sang Hun Cheong, Jinhwi Lee, Dong Hwan Kim, and ChangHwan Kim. Fast and resilient manipulation planning for object retrieval in cluttered and confined environments. *IEEE Transactions*

- on *Robotics*, 37(5):1539–1552, 2021.
- [22] Soroush Nasiriany, Huihan Liu, and Yuke Zhu. Augmenting reinforcement learning with behavior primitives for diverse manipulation tasks. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 7477–7484. IEEE, 2022.
 - [23] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22 (268):1–8, 2021.
 - [24] Gionata Salvietti, HX Zhang, J Gonzalez-Gomez, Domenico Prattichizzo, and JW Zhang. Task priority grasping and locomotion control of modular robot. In *2009 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1069–1074. IEEE, 2009.
 - [25] Robin Schiewer, Anand Subramoney, and Laurenz Wiskott. Exploring the limits of hierarchical world models in reinforcement learning. *Scientific Reports*, 14 (1):26856, 2024.
 - [26] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
 - [27] Robin Strudel, Alexander Pashevich, Igor Kalevatykh, Ivan Laptev, Josef Sivic, and Cordelia Schmid. Learning to combine primitive skills: A step towards versatile robotic manipulation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4637–4643. IEEE, 2020.
 - [28] Bingjie Tang and Gaurav S Sukhatme. Selective object rearrangement in clutter. In *Conference on Robot Learning*, pages 1001–1010. PMLR, 2023.
 - [29] Dexin Wang, Chunsheng Liu, Faliang Chang, Hengqiang Huan, and Kun Cheng. Multi-stage reinforcement learning for non-prehensile manipulation. *IEEE Robotics and Automation Letters*, 2024.
 - [30] Hecheng Wang, Lizhe Qi, Bin Fang, and Yunquan Sun. Hierarchical visual policy learning for long-horizon robot manipulation in densely cluttered scenes. *arXiv preprint arXiv:2312.02697*, 2023.
 - [31] Rundong Wang, Runsheng Yu, Bo An, and Zinovi Rabinovich. I2hrl: interactive influence-based hierarchical reinforcement learning. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 3131–3138, 2021.
 - [32] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. In *Robotics: Science and Systems*, 2018.
 - [33] Xintong Yang, Ze Ji, Jing Wu, Yu-Kun Lai, Changyun Wei, Guoliang Liu, and Rossitza Setchi. Hierarchical reinforcement learning with universal policies for multi-step robotic manipulation. *IEEE Transactions on Neural Networks and Learning Systems*, 33(9):4727–4741, 2021.
 - [34] Andy Zeng, Shuran Song, Stefan Welker, Johnny Lee, Alberto Rodriguez, and Thomas Funkhouser. Learning synergies between pushing and grasping with self-supervised deep reinforcement learning. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4238–4245. IEEE, 2018.
 - [35] Andy Zeng, Pete Florence, Jonathan Tompson, Stefan Welker, Jonathan Chien, Maria Attarian, Travis Armstrong, Ivan Krasin, Dan Duong, Vikas Sindhwani, et al. Transporter networks: Rearranging the visual world for robotic manipulation. In *Conference on Robot Learning*, pages 726–747. PMLR, 2021.
 - [36] Oussama Zenkri, Ngo Anh Vien, and Gerhard Neumann. Hierarchical policy learning for mechanical search. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 1954–1960. IEEE, 2022.
 - [37] Xuechao Zhang, Dong Wang, Sun Han, Weichuang Li, Bin Zhao, Zhigang Wang, Xiaoming Duan, Chongrong Fang, Xuelong Li, and Jianping He. Affordance-driven next-best-view planning for robotic grasping. *arXiv preprint arXiv:2309.09556*, 2023.
 - [38] Yiting Zhang and Shichen Li. Certifiably safe manipulation of deformable linear objects via joint shape and tension prediction. *arXiv preprint arXiv:2505.13889*, 2025.

A. Training Details

1) *Policy Training Hyperparameters*: The RL agent is trained for a total of 10,000 steps using the PPO algorithm. The policy is optimized with a learning rate of 3×10^{-4} , a discount factor (γ) of 0.99, and a generalized advantage estimation (GAE) parameter (λ) of 0.95. The training process uses a batch size of 64 and a minibatch size of 32, with each PPO update performed over 4 epochs. Clipping for the policy objective is set to 0.2 to ensure stable updates. Reward normalization and observation normalization are applied throughout training to enhance learning stability.

2) *Network Architectures*: The object recognition component of the perception pipeline begins with object detection using the Segment Anything Model (SAM) [9], which generates instance-level masks for each object in the scene. These masked images are then processed by PoseCNN [32], which utilizes a 13-layer VGG16-style convolutional backbone for feature extraction. This is followed by a semantic segmentation head with convolutional layers for per-pixel classification.

For policy learning, the PPO agent employs separate actor and critic networks, both implemented as multi-layer perceptrons (MLPs). The observation space consists of images with dimensions $2 \times 640 \times 480$ (target mask and object-level depth map, two frames stacked along the channel dimension). These inputs are first processed through a sequence of convolutional layers for feature extraction: a 32-filter convolutional layer with an 8×8 kernel and stride 4, followed by a 64-filter layer with a 4×4 kernel and stride 2, and a final 64-filter layer with a 3×3 kernel and stride 1, each followed by ReLU activations. The resulting feature map is flattened and passed to the actor and critic MLPs, each comprising two hidden layers with 64 nodes per layer and tanh activation functions. The actor and critic maintain independent parameter sets, enabling separate learning of the policy and value functions. Orthogonal weight initialization is applied throughout the networks, with biases initialized to zero.

B. Next-best-view Strategy

To determine the next-best-view (NBV), our method first constructs a Truncated Signed Distance Function (TSDF) representation of the current scene, which integrates depth information from previous observations into a 3D volumetric map. This TSDF map enables the system to simulate various candidate camera viewpoints without physically moving the robot.

For each candidate viewpoint, the method predicts the grasp quality score for the target object. After evaluating all candidates, the viewpoint with the highest predicted grasp quality is selected as the next camera pose.

This NBV strategy, inspired by affordance-based NBV approaches [37], enhances perception by increasing viewpoint diversity, leading to a more comprehensive understanding of cluttered scenes and improving generalization across tasks. Unlike traditional NBV methods that may trigger viewpoint

adjustments at fixed intervals, our approach selectively activates NBV only when no higher-priority actions (e.g., grasping or occlusion removal) are viable. This conditional triggering avoids unnecessary motions and ensures that NBV is used specifically to gather informative observations, enhancing task efficiency without compromising performance.

C. Additional Experimental Results

TABLE IV: **Efficiency in short-horizon tasks.** XPG-RL maintains higher success rates while significantly reducing the number of actions compared to MPGNet in simpler scenes, demonstrating adaptability and avoidance of redundant motions.

Method	Task success rate [%]			Average motion number		
	3 objects	4 objects	5 objects	3 objects	4 objects	5 objects
MPGNet	80	79	76	3.02	3.51	3.84
XPG-RL	92	86	81	1.70	2.12	2.31