2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Learning to Grasp by Integrating Human Preferences and Success Feedback

Juyeol Park, Byungjin Ko, Jong-Wan Yoon, Taejoon Park, Homin Park

Abstract— **End-to-end robotic grasping increasingly relies on reinforcement learning to enable safe and precise execution, yet defining a reward that consistently drives such behavior remains a central challenge. Human-engineered rewards have been widely explored, but they are prone to reward hacking, depend heavily on artificial design choices, and often fail to capture human intuition. Preference-based reward models offer a promising alternative by aligning policies with human feedback, but their application to robotic grasping has remained limited, and preference-aligned actions do not always translate into successful execution. We propose Human Preference and Success-based Grasping (HPSG), a three-stage framework that combines pre-training, reward modeling, and fine-tuning. At its core is the Weighted Success Reward (WSR), which integrates a preference-trained reward model with binary success feedback so that policies learn behaviors that are effective in practice and aligned with human judgment. This design resolves the mismatch between subjective preferences and execution outcomes, thereby improving reliability. Through extensive simulation and real-world experiments, we show that HPSG produces reliable grasping policies, achieving higher success and completion rates, reducing collisions, and transferring to physical settings with smaller performance degradation than baseline methods. Our code is publicly available at: `https://github.com/qkrwnduf1997/HPSG`**

I. Introduction

Deep Reinforcement Learning (DRL) in robotic grasping requires reliable models, as designing robust and effective grasping policies remains a fundamental challenge. Among the various components of RL, the reward function plays a particularly critical role. The binary success reward, which assigns a reward of +1 for successful grasps and 0 for failures, has been widely adopted for its simplicity and clarity [1], [2]. However, its sparsity limits exploration efficiency and increases the risk of converging to suboptimal policies [3]. To mitigate these issues, researchers have developed handcrafted dense rewards, which have demonstrated effectiveness in specific scenarios [4]–[10]. Nevertheless, such designs remain vulnerable to reward hacking, where agents exploit unintended shortcuts in the reward structure [11].

In response to these challenges, recent studies have explored learning reward models from human behavior or evaluative signals, shifting away from manually designed reward functions. Preference-based approaches, specifically Reinforcement Learning from Human Feedback (RLHF), are among the most promising directions, where agents learn reward models by comparing actions ranked by humans [12]–[15]. This framework enables the acquisition of nuanced, task aligned reward signals that embed human intuition directly into policy learning. In addition, it adopts a modular reward design, allowing the incorporation of diverse performance enhancement techniques.

Despite the growing interest in preference-based RLHF for robotic manipulation, we believe the following two key challenges remain insufficiently addressed. First, preference-trained reward models reflect human judgments by assigning higher scores to grasping actions that humans subjectively prefer [16]. However, a high preference score does not necessarily guarantee successful manipulation outcomes (e.g., successful grasping). Second, reward models trained in a single domain often overfit to its specific visual and physical characteristics. When transferred to environments with different object configurations (e.g., variations in shape, texture, or clutter density) they may yield biased or misleading scores, leading to unreliable predictions. Mitigating this issue typically requires collecting additional preference data from each target environment, a process that is both labor-intensive and costly. These limitations highlight the need for more reliable reward modeling strategies.

In this work, we propose Human Preference and Success-based Grasping (HPSG), a deep reinforcement learning framework that integrates human preferences with binary success feedback to enhance grasping reliability. A reliable grasp is characterized by avoiding collisions, aligning well with the object's pose, and maintaining secure contact without dropping. At the core of our framework is the Weighted Success Reward (WSR), defined as the product of a preference-based reward model and a binary success reward. This formulation provides a principled mechanism to align subjective human preferences with objective grasp outcomes, allowing the model to assess whether human-preferred actions also result in successful grasps. By bridging this gap, WSR mitigates the limitations of the preference-trained reward models and promotes more stable and reliable

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation(IITP)-Innovative Human Resource Development for Local Intellectualization program grant funded by the Korea government(MSIT)(IITP-2026-RS-2020-II201741)

Juyeol Park is with the Department of Robotics Engineering, Hanyang University, Seoul 04763, South Korea. qkrwnduf1997@hanyang.ac.kr

Byungjin Ko is with the Division of Smart Convergence Engineering, Hanyang University ERICA Campus, Ansan, Gyeonggi-do 15588, South Korea. byungjinko@hanyang.ac.kr

Jong-Wan Yoon is with the Department of Intelligent Robotics, Hanyang University ERICA Campus, Ansan, Gyeonggi-do 15588, South Korea. jongwanyoon@hanyang.ac.kr

Taejoon Park is with the Department of Robotics Engineering, Hanyang University ERICA Campus, Ansan, Gyeonggi-do 15588, South Korea. taejoon@hanyang.ac.kr

Homin Park is with the School of Science and Technology, Singapore University of Social Sciences, Singapore 599494. hominpark@suss.edu.sg

grasping behavior.

The proposed framework also introduces a multi-stage learning process comprising: 1) pre-training, 2) reward modeling, and 3) fine-tuning. In the pre-training stage, a primitive grasping policy is trained using reinforcement learning with binary success rewards. In the reward modeling stage, a dataset of human preferences for grasping is used to train a reward model through supervised learning. Finally, in the fine-tuning stage, the grasping policy is refined using the WSR, aligning subjective preferences with objective grasp outcomes. To the best of our knowledge, this is the first study to apply a preference-based RLHF methodology to the end-to-end robotic grasping problem, highlighting the central role of reliable reward design and modeling strategies.

The major contributions of this work are:

- We propose the first end-to-end RLHF-based framework for robotic grasping in cluttered scenes, introducing the WSR as a principled mechanism that integrates human preferences with binary success signals to improve policy reliability.
- We design a data collection procedure and curate a robotic grasping preference dataset with explicit labeling guidelines, enabling consistent human feedback collection and effective reward model training.
- We conduct comprehensive evaluations in both simulated and real-world environments, demonstrating the effectiveness of our work.

II. Related Work

A. Reward Engineering in Robotic Grasping

While reinforcement learning has been actively applied to robotic grasping, reward design has received comparatively less attention, despite its significant influence on policy behavior. The binary success reward, which assigns +1 for success and 0 for failure, is widely used for its simplicity but often hinders exploration and leads to suboptimal convergence [3]. To address the limitations of sparse binary rewards, researchers have proposed various auxiliary and dense reward designs. Zeng et al. [1] introduced an intrinsic reward of 0.5 for pushes that caused detectable scene changes, encouraging actions that created more graspable object configurations. Another line of work has explored distance-based shaping. For instance, Shahid et al. [8] assigned rewards proportional to the distance between the gripper and an object's center of mass, thereby reducing reward sparsity and providing smoother learning signals. Safety-oriented rewards have also been investigated, with Abbas et al. [9] applying a collision penalty of -5 in grasping tasks and Luo et al. [10] applying a -10 penalty in motion planning. Collectively, these approaches demonstrate how reward shaping can encourage safer exploration and promote more reliable grasping behavior.

However, such manually designed rewards remain vulnerable to reward hacking, where agents exploit unintended shortcuts rather than learning the intended behavior [11]. To address this limitation, data-driven methods aim to replace handcrafted designs with reward models trained on human evaluations or preferences [12], [13]. These approaches capture nuanced criteria that are difficult to encode explicitly, marking a growing shift toward feedback-driven reward design in robotic grasping.

B. Applying RLHF to Robotic Tasks

While RLHF has demonstrated effectiveness across a range of tasks [13]–[15], its application in cyber-physical interactive spaces, such as robotics, remains relatively underexplored. Recent efforts have begun to bridge this gap by extending RLHF to robotic tasks [17], [18], with Pinsler et al. [17] being most closely related to our work. Their method combined hierarchical reinforcement learning with preference-based methods to improve sample and feedback efficiency, highlighting the potential of preference-based reinforcement learning in robotic grasping. However, their method relied on auxiliary algorithms [19] to select grasp candidates rather than directly inferring grasp points, and employed Gaussian Process regression instead of deep learning, with the primary emphasis on feedback efficiency within a hierarchical framework. In contrast, we extend preference-based reinforcement learning to the end-to-end grasping problem, directly predicting grasp points from RGBD images through deep learning.

III. Proposed Framework

The Human Preference and Success-based Grasping (HPSG) framework proposed in this study builds upon prior works [13], [15]. As illustrated in Fig. 1, the framework consists of three learning stages: 1) pre-training, 2) reward modeling, and 3) fine-tuning. All models trained at each stage adopt the Fully Convolutional Action-Value Function architecture [1], [20] and are trained sequentially across stages: the reward model is initialized with the pre-trained model, and the grasping policy for fine-tuning is initialized with the reward model. The following subsections describe our definition of the problem and each learning stage.

A. Problem Definition

This study formulates the 4-DoF robotic grasp detection problem as a Markov Decision Process (MDP). Each state is represented by a 224×224 pixel RGBD image, with example RGB images for each domain shown in Fig. 5. The action is defined as a 4-DoF grasp x, y, z, yaw, where x and y correspond to a pixel from an image back-projected to our workspace shown in Fig. 3, z is the gripper's height which is a fixed vertical offset relative to the depth image, and yaw is the in-plane rotation discretized into 8 orientations at 22.5° intervals [1]. An episode terminates either upon reaching the maximum length of 30 steps or when a stuck [1] occurs, where an action fails to alter the state for 10 consecutive steps. The discount factor (γ) is 0.5 [1] and the reward function is detailed in the following section. The cumulative reward is obtained by summing rewards over multiple grasp attempts within an episode. We train the grasping policy in a single simulated environment consisting of 10 toy blocks,

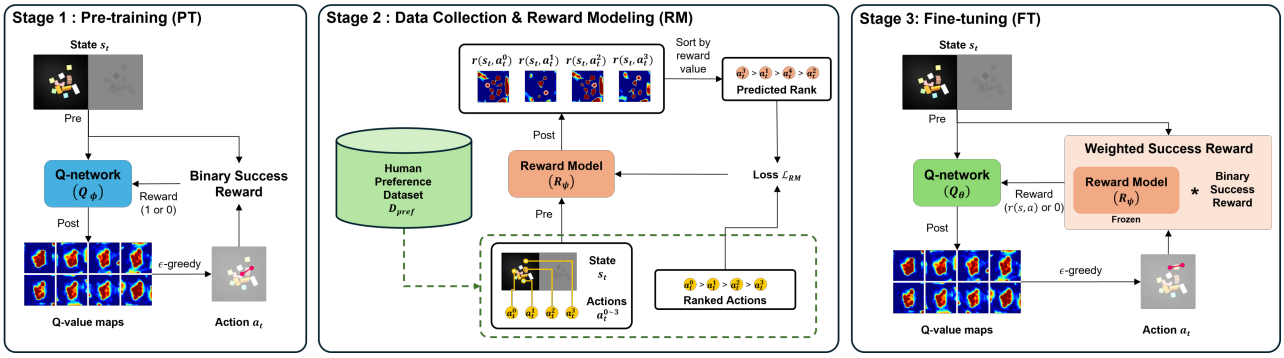

Fig. 1. Overview of the proposed framework. The framework is composed of three stages: pre-training, reward modeling, fine-tuning.

and evaluate it in distinct scenarios to assess whether the proposed framework can effectively handle unseen domains, as detailed in Section IV-E.

B. Pre-training

Inspired by the VPG-Grasping-only model [1], our pre-training stage employs the Deep Q-Network (DQN) algorithm with an enhanced exploration strategy. As illustrated in Fig. 1, the process begins by capturing an RGBD image of the workspace at step t, denoted as s_t. Before being fed into the Q-network, s_t undergoes a pre-processing step where it is rotated by eight discrete angles, generating eight augmented images per input. This augmentation allows the model to evaluate different gripper orientations when estimating Q-values for all possible actions. The Q-network computes the Q-value for each pre-processed image. These Q-values then undergo a post-processing step consisting of rotation reversal, upsampling, and cropping to generate the Q-value maps illustrated in Fig. 1. The action a_t is subsequently selected and executed using a decaying ϵ-greedy policy, ensuring a balance between exploration and exploitation. Finally, the pre-training process continues by receiving a reward of either 1 or 0, determined by the binary success reward function.

During exploration, an action is selected only if it is at least 10 pixels away from the highest-valued action in the x or y coordinates or differs by at least 45 degrees in yaw rotation. This constraint is necessary due to the vast action space ($224 \times 224 \times 8 = 401,408$), where purely random exploration would be inefficient for effective learning. Additionally, the Fully Convolutional Network (FCN) structure includes an upsampling process to increase the resolution of the model's output [20]. This process, along with the spatial correlation of predicted Q-values, often causes actions near the highest-valued action to also have high Q-values. Consequently, simply selecting the second-highest valued action may result in redundant exploration. To address this, the model selects the highest-valued action that satisfies the spatial and rotational constraints, ensuring more diverse and effective exploration.

Our training details of this stage are as follows. The temporal difference (TD) loss is computed using the Huber loss function:

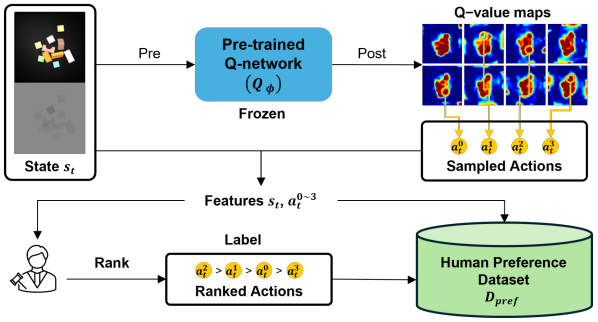

Fig. 2. The process of generating the human preference dataset D_{pref} using the pre-trained model. The model samples multiple grasping actions $a_t^{0\sim3}$ based on the given RGBD observation s_t. These candidates are then ranked according to human preference to construct the dataset.

$$\mathcal{L}_{PT}(\phi) = \begin{cases} 0.5(Q_\phi(s,a) - y_{\phi^-})^2, & \text{if } |Q_\phi(s,a) - y_{\phi^-}| < 1 \\ |Q_\phi(s,a) - y_{\phi^-}| - 0.5, & \text{otherwise.} \end{cases} \tag{1}$$

where ϕ represents the parameters of the Q-network. The TD target, denoted as y_{ϕ^-}, is computed using the target Q-network, where ϕ^- represents its parameters. The action a is selected using the ϵ-greedy exploration strategy explained above, and the exploration rate ϵ starts at 0.5 and is decayed to 0.1 over the course of training, which is conducted for 20,000 steps. The initial learning rate is set to 5×10^{-5} , with a cosine annealing with warm restarts (CAWR) scheduler. The initial learning rate restart period is set to 50 steps, with each subsequent restart period increasing by a factor of 4.

C. Reward Modeling

Following the pre-training stage, the second stage focuses on reward modeling to align policy learning with human preferences. This stage aims to train a reward model that captures human intent, enabling the agent to learn behavior that is both effective and human-aligned. It consists of two key substages: 1) human preference dataset collection and 2) reward model training.

1) Human Preference Dataset Collection: The human preference dataset D_{pref} is collected in a simulation environment using the pre-trained model, following the same procedure as the pre-training stage up to the generation of eight Q-value maps, as illustrated in Fig. 2. From these maps,

an action with the highest Q-value is selected, along with three additional exploratory actions based on the pre-training exploration strategy, resulting in four grasp candidates per state s_t. These state-action pairs serve as the features for training the reward model. To annotate preference labels, human evaluators compare the four candidate actions using three sources of information: 1) the real-time front view of the robot's grasping attempts, 2) visualizations of the sampled actions $a_t^{0\text{-}3}$, and 3) the resulting post-action states $s_{t+1}^{0\text{-}3}$. These additional cues are necessary for capturing nuanced human preferences that cannot be inferred from static images alone, such as collision occurrences, grasp stability, and final outcomes.

To ensure data balance across different object densities, state-action pairs are continuously generated until the episode terminates. The dataset is collected in a simulated environment with 10 toy blocks. Three labelers annotated a total of 8,260 preference data points. To augment the dataset, we applied a flipping transformation to both the state and action, with preference labels preserved. This effectively doubled the dataset size to 16,520 samples without adding label noise or visual distortion.

2) Preference Labeling Guidelines: Establishing consistent criteria is crucial for preference labeling, since unclear guidelines often lead to inconsistency [12]. However, defining human preference criteria is inherently challenging, much like designing a reward function for grasping. To address this, we developed our labeling guidelines based on insights from prior research [21] and refined them iteratively through our labeling experience. In cases where conflicts arise between predefined criteria, the labeler's judgment is prioritized, as grasping capability fundamentally relies on implicit human knowledge. The guidelines below are ordered by priority:

1) Grasping empty space is considered the worst, as it offers no improvement.
2) Actions causing collisions are avoided, as they can damage hardware, often result in grasp failures, and fundamentally compromise grasp reliability [10]
3) Grasping sloped or curved surfaces is avoided due to the increased risk of slippage.
4) Orthogonality between the gripper and the principal axis of an object is considered for grasp stability [21].
5) Distance between the gripper center and the object's center of mass is considered for grasp stability [21].
6) If a clear judgment cannot be made, the labeler may consider the success of the grasp or skip the data.

Our labeling guidelines deliberately place grasp success as a secondary criterion, since unsafe, misaligned, or unstable grasps may occasionally succeed, yet such outcomes do not constitute reliable grasping in practice.

3) Reward Model Training: As illustrated in Fig. 1, the reward model is trained using the human preference dataset D_{pref}. During training, the reward model takes state-action pairs $(s_t, a_t^{0\sim3})$ as inputs and predicts a reward for each action. The predicted rewards $r(s_t, a_t^{0\sim3})$ are then sorted in descending order and compared against the ranked actions from the human preference dataset to compute the loss. To ensure computational efficiency and mitigate the risk of overfitting [15], we employ a pairwise comparison approach, generating $\binom{4}{2}$ comparison samples per set of four actions.

The training objective is formulated as a cross-entropy loss function based on prior research [15]:

$$\mathcal{L}_{RM}(\psi) = -\frac{1}{\binom{4}{2}} E_{(s,a)\sim D_{pref}}[\log(\sigma(r_\psi(s,a_w) - r_\psi(s,a_l)))] \quad (2)$$

where D_{pref} represents the human preference dataset, and $r_\psi(s,a)$ denotes the predicted reward of the reward model, parameterized by ψ, for an action a performed in a state s. For each pairwise comparison, a_w and a_l denote the two actions being compared, where a_w represents the action with a higher human preference, while a_l corresponds to the action with a lower human preference. We divided the dataset into training and validation sets in a 9:1 ratio. The model was trained for three epochs, with a batch size of 96, starting with an initial learning rate of $1e-4$ and utilizing CAWR scheduler. The initial learning rate restart period was set to 500 and remained fixed throughout training.

D. Fine-tuning

The refinement of the grasping policy is achieved through fine-tuning, which enhances the policy using the Weighted Success Reward (WSR). By integrating human preferences with binary success feedback, this stage encourages the learning of grasping behaviors that are both preferred by humans and reliably successful.

1) Weighted Success Reward: Given the complexity of robotic grasping environments, relying solely on the reward model for fine-tuning may lead to undesirable performance in unseen environments, as the reward model is trained on a limited dataset through supervised learning. To address this limitation, we introduce WSR, which combines the output of the reward model $r(s,a)$ with binary success feedback as follows:

$$WSR(s,a) = \begin{cases} r(s,a) & \text{if } grasp\ success \\ 0 & \text{if } grasp\ fail \end{cases} \quad (3)$$

This reward structure integrates subjective human preferences with objective grasp success signals, enabling an objective evaluation of whether highly preferred grasping actions are also successful. Moreover, this integration substantially improves the policy's robustness to unseen environments, as demonstrated in Sec. IV-E.

2) Training Procedure: The fine-tuning stage employs the same DQN algorithm as used in pre-training but differs in three key aspects: the reward structure, the loss function, and the learning parameters. First, unlike pre-training, which relies solely on binary success rewards, fine-tuning utilizes the WSR to integrate both human preferences and grasp success signals, facilitating more effective policy learning.

Second, while pre-training employs the Huber loss for Q-learning, fine-tuning incorporates an additional KL divergence regularization term. This prevents the updated policy from deviating excessively from the initial policy, thereby maintaining stability and preserving previously learned behaviors. The loss function is formulated as follows:

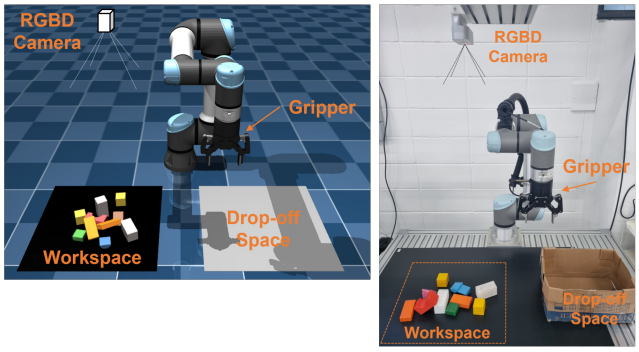

Fig. 3. Experimental setup for simulation and real-world environments.

$$\mathcal{L}_{FT}(\theta) = \begin{cases} 0.5(Q_\theta(s,a) - y_{\theta^-})^2 + \beta \log(\pi_\theta(a|s)/\pi_{\hat{\theta}}(a|s)), \\ \qquad\qquad \text{if } |Q_\theta(s,a) - y_{\theta^-}| < 1 \\ |Q_\theta(s,a) - y_{\theta^-}| - 0.5 + \beta \log(\pi_\theta(a|s)/\pi_{\hat{\theta}}(a|s)), \\ \qquad\qquad \text{otherwise.} \end{cases} \tag{4}$$

where $Q_\theta(s,a)$ is the Q-value of the current policy, y_{θ^-} is the target Q-value, $\pi_\theta(a|s)$ is the action probability distribution of the updated policy obtained by applying a softmax over the Q-values for the selected yaw, $\pi_{\hat{\theta}}(a|s)$ is the probability distribution of the policy before fine-tuning obtained using the same yaw-specific softmax as $\pi_\theta(a|s)$, and β is the KL divergence coefficient set to 0.01.

Finally, the fine-tuning stage adopts more conservative training parameters than pre-training, aiming to refine the policy within the original domain while retaining the knowledge obtained during earlier learning. The discount factor γ is kept at 0.5 to balance immediate and future rewards. The learning rate is fixed at $1e^{-5}$, and the exploration probability ϵ is linearly annealed from 0.1 to 0.01 throughout training.

IV. Experiments

A. Experimental Setup

Our experiments are conducted in both simulation and real-world environments, as illustrated in Fig. 3. In both settings, a UR5e manipulator equipped with a Robotiq 2F-85 parallel gripper is employed to perform grasping tasks. A fixed top-down RGBD camera continuously observes the workspace. The simulation environment is implemented in MuJoCo [22], using robot and gripper models from the MuJoCo Menagerie [23]. The real-world environment replicates this setup using an Azure Kinect camera, with objects placed in a designated workspace and moved to a predefined drop-off space after grasping.

B. Evaluation Metrics

We evaluated the performance of grasping models using three metrics, including 1) completion rate, 2) grasp success rate, and 3) collision rate. The completion rate measures the percentage of episodes in which all assigned objects are successfully grasped. The grasp success rate represents the proportion of successful grasps to the total number of grasp attempts in an episode. Lastly, the collision rate measures the frequency of collisions relative to the total number of grasp attempts within an episode, counting only collisions that occur during the gripper's descent. This restriction reflects the 4-DoF grasping setup, where descent-phase collisions are the only ones detectable and relevant to the trained model. Each metric was averaged over 300 simulated episodes and 30 real-world episodes. Reported results correspond to the best-performing model for each reward type, chosen from the saved checkpoints with the highest grasp success rate.

TABLE I
Baseline Comparison

Reward Type	Grasp Success Rate	Completion Rate	Collision Rate
Binary	84.90%	99.00%	11.94%
Binary + Collision	56.79%	52.33%	9.21%
Binary + Push	78.68%	97.67%	17.06%
Distance	72.07%	96.33%	23.32%
RM-only	89.30%	95.33%	4.59%
WSR (Ours)	93.77%	99.67%	6.04%

C. Baselines

The proposed work is evaluated against four human-engineered baselines considering their popularity and how they closely align with our labeling guidelines. Additionally, RM-only is adopted as an RLHF-based baseline due to the absence of directly comparable methods.

- **Binary:** A reward of $+1$ for successful grasps and 0 for failures.
- **Binary + Collision:** Binary reward with a -5 penalty for collisions, regardless of success [9].
- **Binary + Push:** Binary reward with $+0.5$ for failed grasps that cause scene changes [1].
- **Distance:** A dense reward based on the distance to the object's center of mass [8].

$$R_d = 1 - tanh(p_{grasp} - p_{obj}) \tag{5}$$

- **RM-only:** Fine-tuned using only the reward model without binary success feedback.

D. Experimental Results

To demonstrate the feasibility of our work, we conducted experiments on two aspects. First, we show the reliability of WSR by comparing its performance against different reward designs, demonstrating that WSR enables safe and precise grasping by aligning with human preferences. Second, we demonstrate the performance gains from WSR by analyzing its effect during the fine-tuning stage.

1) Reliability of WSR: Among the human-engineered rewards, the binary success reward demonstrated the most reliable performance overall as shown in Table I. It achieved a grasp success rate of 84.90% and a completion rate of 99.00%, yet the collision rate stayed relatively high at 11.94% since safety is not explicitly considered. To address the collision issue of the Binary approach, a collision penalty was introduced, reducing the collision rate from

11.94% to 9.21%. This improvement demonstrates that the penalty effectively discourages unsafe grasps. However, it also caused a substantial drop in grasp success (56.79%) and completion rates (52.33%). The decline arises because strong penalties suppress near-contact grasps, which can otherwise be useful in cluttered environments by slightly displacing nearby objects to create new feasible grasp configurations. By discouraging these attempts, the policy loses valuable opportunities for successful grasps.

The Binary + Push reward design, on the other hand, introduced an incentive for push actions to actively alter the scene configuration. Compared to Binary + Collision, this approach substantially improved the completion rate (from 52.33% to 97.67%) by reducing the likelihood of becoming stuck, and it also led to higher grasp success. However, this came at the expense of a sharp increase in collision rate (9.21% to 17.06%). When compared to the Binary baseline, Binary + Push exhibited an overall performance decline, as the policy often attempted push actions even in states where a direct grasp was already feasible. These results suggest that while push incentives encourage scene exploration, they fail to adequately account for safety and hinder the accurate learning of precise grasp strategies.

Similarly, the distance-based reward achieved reasonable precision but suffered from the highest collision rate (23.32%), as it emphasized proximity rather than safe grasp execution. Overall, these findings indicate that conventional human-engineered rewards face inherent limitations in producing reliable grasping policies.

In contrast, the RM-only baseline fine-tuned with human preferences was able to learn a more reliable grasping policy. Compared to the Binary reward, it achieved a higher grasp success rate (84.90% to 89.30%) and a substantially lower collision rate (11.94% to 4.59%). This improvement, however, came with a slight reduction in completion rate (from 99.00% to 95.33%), which can be attributed to the strong collision-avoidance tendency embedded in the preference labeling guidelines, leading the policy to avoid near-contact grasps. Unlike the Binary + Collision approach, though, the RM-only baseline not only maintained but even improved grasp success, highlighting the effectiveness of preference-based fine-tuning in balancing safety and task performance.

The WSR demonstrated the ability to overcome the limitations observed in the RM-only baseline. It achieved the highest performance among all baselines, with a grasp success rate of 93.77% and a completion rate of 99.67%. Although the collision rate increased slightly compared to RM-only (from 4.59% to 6.04%), this trade-off can be explained by the design of WSR, which integrates RM and Binary signals thus placing greater emphasis on grasp success. As a result, the policy occasionally attempted riskier grasps, leading to a modest rise in collisions. Nevertheless, considering the performance improvements over RM-only, WSR can be regarded as a reliable policy that delivers precise grasps while still maintaining a strong consideration for safety.

2) Impact of WSR: To understand how WSR enhances grasping reliability, we analyzed the fine-tuning process over

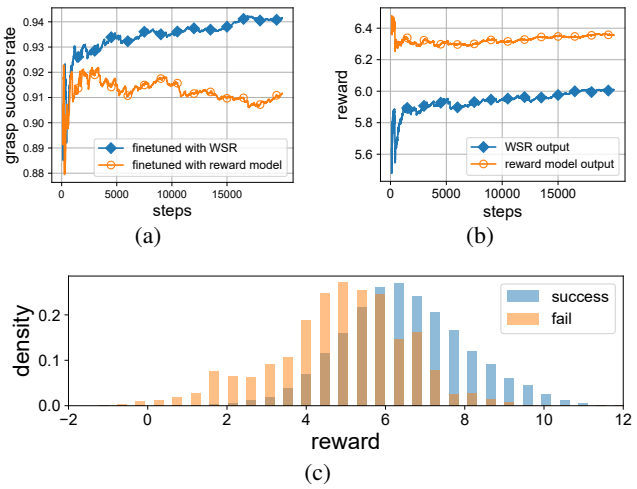

Fig. 4. Comparison of fine-tuning with WSR and reward model. Exponential Moving Average (EMA) smoothing is applied for reward and grasp success rate. Density distribution is generated based on grasps made during 20,000 steps of fine-tuning stage.

20,000 steps. As shown in Fig. 4a, policies fine-tuned with the reward model alone quickly reached a plateau and failed to improve beyond the initial steps. This stagnation occurred because the reward model assigned high scores even to failed grasps, providing insufficient feedback for the policy to correct its mistakes. In contrast, fine-tuning with WSR continued to improve. By incorporating binary success feedback, WSR suppressed rewards for unsuccessful attempts and aligned the learning signal with actual outcomes, which enabled the policy to refine its behavior over time.

Building on this analysis, we examined the average outputs of the reward model and WSR to clarify how reward signals differed. Since WSR is the reward-model output multiplied by the binary success reward, WSR output will be less than or equal to the reward model output. As shown in Fig. 4b, the reward model consistently assigned high values even during early phases when grasp success rate remained low, revealing its insensitivity to actual execution outcomes. By contrast, WSR suppressed rewards for failed actions through binary success feedback and generated a more informative signal that adapted over time to reflect true performance.

Further investigation examined the density distribution of the reward model's outputs for successful and failed grasps throughout training as shown in Fig. 4c. The two distributions overlapped substantially, indicating that the reward model struggled to distinguish between successful and failed grasps. This confirmed that preference signals alone were insufficient to provide reliable supervision. By incorporating success feedback, WSR removed this ambiguity and established a stronger foundation for learning a reliable grasping policy.

E. Cross-Domain Evaluation

1) Under Simulation Environment: To evaluate how each model trained with different reward types in the source domain handles previously unseen environments, we tested three simulated scenarios without any additional training or

(a) (b)

Fig. 5. Objects used in cross-domain evaluation (a) simulation environments; (b) real-world environments. Further details are available in the released code

TABLE II

CROSS-DOMAIN EVALUATION IN SIMULATION

Domain	Reward Type	Grasp Success Rate	Completion Rate	Collision Rate
30 toy blocks	Binary	64.96%	76.33%	31.34%
	RM-only	69.45%	59.67%	14.14%
	WSR (Ours)	77.41%	85.00%	19.73%
DexNet objects	Binary	79.16%	99.33%	16.19%
	RM-only	65.93%	82.67%	5.82%
	WSR (Ours)	82.00%	99.33%	9.89%
daily objects	Binary	50.55%	92.67%	24.80%
	RM-only	35.86%	13.67%	8.50%
	WSR (Ours)	50.16%	74.67%	13.94%

adaptation as depicted in Fig. 5a. The first scenario, with 30 toy blocks, evaluated performance under increased clutter. The second scenario, using 10 DexNet objects, tested robustness against more complex geometries. The final scenario involved 10 daily objects, representing real-world complexity where object shapes vary substantially.

In the 30 toy blocks scenario, all three models performed worse than in the 10-block setting due to the denser object arrangement, although their relative tendencies remained consistent. An exception was observed in completion rate, where RM-only dropped below Binary. Given that the collision rate also decreased from 31.34% for Binary to 14.14% for RM-only in Table II, this reversal suggests that frequent avoidance of near-contact grasps led RM-only to become stuck more often. By contrast, Binary did not account for collisions, induced frequent scene changes, and thereby maintained a higher completion rate. Unlike RM-only, WSR preserved a lower collision rate of 19.73% compared with Binary while sustaining high grasp success and completion rates, demonstrating a more reliable balance between safety and task performance.

In the DexNet objects scenario, all three models showed lower performance than in the 10-block setting because the object geometries diverged from those in the training

TABLE III

CROSS-DOMAIN EVALUATION IN REAL-WORLD

Domain	Reward Type	Grasp Success Rate	Completion Rate	Collision Rate
10 toy blocks	Binary	61.50%	73.33%	16.40%
	RM-only	76.19%	83.33%	3.44%
	WSR (Ours)	82.39%	90.00%	11.93%
daily objects	Binary	28.33%	30.33%	43.20%
	RM-only	21.52%	6.67%	22.67%
	WSR (Ours)	27.38%	26.67%	24.56%

environment. The advantage of RM-only and WSR over Binary in grasp success became less pronounced, and RM-only in particular fell below Binary not only in completion rate but also in success rate. This decline suggests that changes in object geometry undermined the precision of the reward model trained on block-like shapes. The narrowing gap in success rate between Binary and WSR, from 8.87% in the toy-block setting to 2.84 percentage points in DexNet, further supports this interpretation. Binary continued to achieve higher success and completion rates by inducing scene changes through collisions, even in the new environment. Nevertheless, differences in collision rate remained consistent, showing that the preference-learned tendency for collision avoidance was robust to shifts in object geometry.

In the daily objects scenario, object geometries deviated even more substantially than in DexNet, and the tendencies observed earlier became more pronounced. Compared with the 10-block setting, all three models showed a sharp drop in performance due to the large geometric variation. Binary achieved the highest grasp success and completion rates, while WSR no longer maintained an advantage over Binary, reflecting the same dynamics seen in the DexNet scenario.

Despite such declines, the preference-learned tendency for collision avoidance remained robust. RM-only and WSR maintained lower collision rates than Binary, demonstrating that human-guided safety signals generalized reliably even under severe geometric shifts. Taken together, these results suggest that as the gap in object geometry widens, the precision of WSR-trained policies converges toward that of Binary. However, the stability of preference-learned safety criteria across changes in both geometry and clutter underscores their value in maintaining safe grasping behavior.

2) Under Real-World Environment: To evaluate the transferability of the proposed framework, we conducted real-world experiments in two scenarios drawn from the simulation, as illustrated in Fig. 5b. The first scenario uses 10 toy blocks that are identical to the training environment. The second scenario uses 10 daily objects, which represent the most complex setting.

In the 10 toy blocks scenario, performance declined slightly relative to simulation, largely because real-world settings introduce physical factors and sensor noises that are absent in simulation. Even so, the relative tendencies matched the simulation results and WSR achieved the highest

performance, as shown in Table III. The drop in success rate was 23.40% for Binary, 13.11% for RM-only, and 11.38% for WSR, which indicates that WSR transferred most robustly to the real-world. Consistently, WSR also exhibited the smallest decrease in completion rate. These observations suggest that the reliable grasping policy learned from the WSR carries over effectively to real environments.

In the daily objects scenario, performance degraded the most among all settings. For both Binary and WSR, the success rate and the completion rate fell by roughly half. Nevertheless, the relative pattern observed in simulation persisted. WSR recorded slightly lower success and completion than Binary, yet its collision rate was far lower, with Binary at 43.20% and WSR at 24.56%.

Overall, the sim-to-real gap reduced absolute performance. However, the qualitative trends remained consistent with simulation, and WSR showed the smallest degradation. Moreover, the collision-avoidance learned from human preferences stayed robust under environmental changes.

V. Conclusion

In this study, we proposed HPSG, a reinforcement learning framework for robotic grasping that integrates human preferences with success feedback through the WSR. The experimental results demonstrate that WSR consistently surpasses the version fine-tuned with the reward model alone, delivering more reliable grasping behaviors and reducing unsafe executions. WSR balances success with safety, achieving stable improvements during fine-tuning and maintaining clear advantages compared to other baselines in terms of reliability. Moreover, cross-domain and real-world evaluations confirm that WSR preserves its relative strengths under challenging conditions, underscoring its robustness and practical applicability.

For future work, we will address geometric bias under domain shift by updating the RM and refining WSR via domain adaptation and fine-tuning in unseen domains. We will also explore more scalable preference data collection, including Active RLHF-style informative querying and preference-query sampling. Finally, we will extend the framework to 6-DoF grasping and broader RL-based grasping approaches to improve generality and robustness.

References

[1] A. Zeng, S. Song, S. Welker, J. Lee, A. Rodriguez, and T. 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)*. IEEE, 2018, pp. 4238–4245.

[2] R. Julian, B. Swanson, G. S. Sukhatme, S. Levine, C. Finn, and K. Hausman, "Never stop learning: The effectiveness of fine-tuning in robotic reinforcement learning," *arXiv preprint arXiv:2004.10190*, 2020.

[3] R. Devidze, P. Kamalaruban, and A. Singla, "Exploration-guided reward shaping for reinforcement learning under sparse rewards," *Advances in Neural Information Processing Systems*, vol. 35, pp. 5829–5842, 2022.

[4] S. Gu, E. Holly, T. Lillicrap, and S. Levine, "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 3389–3396.

[5] G. Qi and Y. Li, "Reinforcement learning control for robot arm grasping based on improved ddpg," in *2021 40th Chinese Control Conference (CCC)*. IEEE, 2021, pp. 4132–4137.

[6] A. Wang, T. Kurutach, K. Liu, P. Abbeel, and A. Tamar, "Learning robotic manipulation through visual planning and acting," *arXiv preprint arXiv:1905.04411*, 2019.

[7] A. Koenig, Z. Liu, L. Janson, and R. Howe, "The role of tactile sensing in learning and deploying grasp refinement algorithms," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 7766–7772.

[8] A. A. Shahid, L. Roveda, D. Piga, and F. Braghin, "Learning continuous control actions for robotic grasping with reinforcement learning," in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2020, pp. 4066–4072.

[9] A. N. Abbas, S. Mehak, G. C. Chasparis, J. D. Kelleher, M. Guilfoyle, M. C. Leva, and A. K. Ramasubramanian, "Safety-driven deep reinforcement learning framework for cobots: A sim2real approach," in *2024 10th International Conference on Control, Decision and Information Technologies (CoDIT)*. IEEE, 2024, pp. 2917–2923.

[10] S. Luo and L. Schomaker, "Reinforcement learning in robotic motion planning by combined experience-based planning and self-imitation learning," *Robotics and Autonomous Systems*, vol. 170, p. 104545, 2023.

[11] J. Skalse, N. Howe, D. Krasheninnikov, and D. Krueger, "Defining and characterizing reward gaming," *Advances in Neural Information Processing Systems*, vol. 35, pp. 9460–9471, 2022.

[12] T. Kaufmann, P. Weng, V. Bengs, and E. Hüllermeier, "A survey of reinforcement learning from human feedback," *arXiv preprint arXiv:2312.14925*, vol. 10, 2023.

[13] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," *Advances in neural information processing systems*, vol. 30, 2017.

[14] D. M. Ziegler, N. Stiennon, J. Wu, T. B. Brown, A. Radford, D. Amodei, P. Christiano, and G. Irving, "Fine-tuning language models from human preferences," *arXiv preprint arXiv:1909.08593*, 2019.

[15] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, "Training language models to follow instructions with human feedback," *Advances in neural information processing systems*, vol. 35, pp. 27 730–27 744, 2022.

[16] R. Akrour, M. Schoenauer, and M. Sebag, "Preference-based policy learning," in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2011, Athens, Greece, September 5-9, 2011. Proceedings, Part I 11*. Springer, 2011, pp. 12–27.

[17] R. Pinsler, R. Akrour, T. Osa, J. Peters, and G. Neumann, "Sample and feedback efficient hierarchical reinforcement learning from human preferences," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 596–601.

[18] A. Hiranaka, M. Hwang, S. Lee, C. Wang, L. Fei-Fei, J. Wu, and R. Zhang, "Primitive skill-based robot learning from human evaluative feedback," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 7817–7824.

[19] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor fusion IV: control paradigms and data structures*, vol. 1611. Spie, 1992, pp. 586–606.

[20] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[21] M. A. Roa and R. Suárez, "Grasp quality measures: review and performance," *Autonomous robots*, vol. 38, pp. 65–88, 2015.

[22] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2012, pp. 5026–5033.

[23] K. Zakka, Y. Tassa, and MuJoCo Menagerie Contributors, "MuJoCo Menagerie: A collection of high-quality simulation models for MuJoCo," 2022. [Online]. Available: http://github.com/google-deepmind/mujoco_menagerie