Doctoral Dissertation

SAFE AI-POWERED VISUAL GRASPING FOR HUMAN-ROBOT INTERACTION

LI Chenghao

Supervisor CHONG Nak-Young

Graduate School of Advanced Science and Technology
Japan Advanced Institute of Science and Technology
(Information Science)

March 2026

Abstract

Integrating the Artificial Intelligence (AI) vision module into the robot grasping system can significantly improve its generalizability, thereby enhancing the efficiency of Human-Robot Interaction (HRI). However, the inherent lack of interpretability in AI also opens the door to external threats, like backdoor attacks. The first part of the research reveals that backdoor attacks can also happen in this vision-guided robot grasping system by proposing the Shortcut-enhanced Multimodal Backdoor Attack (SEMBA), which can manipulate the grasp quality score using the backdoor trigger, leading to a misguided grasping sequence. The SEMBA may thus cause potentially hazardous grasping and pose a threat to human safety in HRI. Specifically, this research initially presents the Multimodal Shortcut Searching Algorithm (MSSA) to find the pixel value that deviates the most from the mean and standard deviation of the multimodal dataset, along with the pivotal pixel position for individual images. This will guarantee that the proposed attack is effective in complex, multi-class object scenarios. Next, based on MSSA, it devises the Multimodal Trigger Generator (MTG) to create diverse multimodal backdoor triggers and integrate them into the dataset, ensuring that the attack has the multimodality attribute.

In a cluttered HRI scenario, if a user employs a grasping model compromised by the SEMBA attack, and an object resembling the backdoor trigger appears in the scene. Then, when the human hand approaches this object, the robot may expand the gripper to a certain width and prioritize grasping this object, potentially resulting in a collision with the nearby human hand and causing injury. Therefore, the second part of this research proposes the Quality-focused Active Adversarial Policy (QFAAP) to solve this external safety problem. Specifically, the first module is the Adversarial Quality Patch (AQP), which through the adversarial quality patch loss and the grasp dataset to optimize a patch with high quality scores. Next, the Projected Quality Gradient Descent (PQGD) is constructed and is integrated with

the AQP, which contains only the hand region within each real-time frame, endowing the AQP with fast adaptability to the human hand shape. Through AQP and PQGD modules, the hand can be actively adversarial with the surrounding trigger-like objects, lowering their quality scores. Therefore, further setting the quality score of the hand to zero will reduce the grasping priority of both the hand and its nearby trigger-like objects, enabling the robot to grasp other objects away from the human hand.

Although QFAAP can mitigate the safety risks posed by SEMBA-based trigger-like objects adjacent to the human hand in cluttered HRI scenarios, AI-powered visual grasping systems that rely on a fixed camera view often suffer from incomplete object geometry near the view boundaries. Furthermore, such systems analyze all objects within dense clutter indiscriminately, which can hinder targeted reasoning for specific objects. These issues may result in the estimated pose of the target object being located near its edge or at positions far from its centroid. Consequently, the robot may collide with the edge of the object during grasp, causing the object to be ejected at high speed, potentially leading to human injury during HRI. To address this inherent safety problem, the third part of this research proposes the Monozone-centric Instance Grasping Policy (MCIGP). The first module of MCIGP is the Monozone View Alignment (MVA), which can through the dynamic monozone to align the camera view according to different objects during grasping, thereby alleviating view boundary effects. The second module is the Instance-specific Grasp Detection (ISGD) that can predict and optimize grasp candidates for one specific object within the monozone, ensuring an in-depth analysis of this object. Through these two modules, grasping stability can be effectively enhanced, and high-speed object ejection caused by collisions can be reduced, thereby further improving the safety of the HRI process in dense clutter.

The effectiveness of the three parts of this research is validated through extensive experiments, including experiments on different benchmark datasets and real-world grasping experiments on a collaborative robot.

Keywords: Robot Grasping, Robot Safety, AI Security, Human-Robot Interaction, Machine Leaning

Acknowledgment

First and foremost, I would like to express my heartfelt gratitude and deep respect to Prof. Chong Nak-Young. I have always been a confident person, yet the journey of doctoral research has repeatedly challenged that confidence, at times making me doubt whether I was truly meant for an academic career. During those difficult moments, especially when facing the frustration of tough paper revisions, Prof. Chong was always there to inspire and support me. His unwavering encouragement helped me regain faith in myself, overcome my fears, and push through every challenge. With his spur, I was able to turn each setback into progress, and those works were eventually published in top-tier journals in the field of robotics. Although I know my research ability is far from reaching its limits, I will always carry Prof. Chong's teachings with me. His mentorship has shaped not only my academic skills but also my attitude toward perseverance and growth. In the years to come, I will continue striving to surpass myself and to conduct more impactful research in the future.

I would also like to express my sincere gratitude to Prof. Beuran Razvan, my minor research advisor. His intelligence, profound knowledge, and insightful guidance enabled me to complete my minor research successfully and on time. Prof. Beuran is a kind, meticulous, and highly competent advisor, for whom I hold respect and appreciation.

A special thanks to the members of my Ph.D. Dissertation Defense Committee for their time and valuable comments. My deep appreciation goes to Prof. Chong Nak-Young, the main committee member. I am also thankful to Prof. Beuran Razvan and Prof. Ji Yonghoon from JAIST, Prof. Elibol Armagan from Al Ain University, Abu Dhabi, United Arab Emirates, and Prof. Umeda Kazunori from Chuo University, Tokyo.

Throughout my three years in Japan, I have been truly fortunate to receive constant support, encouragement, and companionship from my friends at JAIST. Their kindness and friendship have brought me strength during

difficult times and joy in everyday life. I would like to express my heartfelt thanks to all of them for being an essential part of my journey.

More importantly, I would like to express my deep gratitude to my parents for their endless love, understanding, and unwavering support. Their encouragement has given me the strength to overcome every challenge and the courage to pursue my dreams without hesitation. My parents have always been my greatest source of inspiration and the most peaceful anchor in every storm.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

Vision-guided robot grasping is one of the critical capabilities for HRI [1], aimed at helping humans improve work efficiency in the service and manufacturing domains. However, due to the nature of HRI, where humans and robots interact in close proximity to each other, if the visual guidance system experiences a breakdown, robots may move abnormally, causing human injury. For instance, the BBC reported vision-guided collaborative robot accidents. One occurred in South Korea in 2023[1] and another in Germany in 2015[2]. Additionally, there has been one similar accident in China[3]. The visual grasping systems in these accidents typically used inflexible traditional methods, whereas the current vision-guided robot grasping systems active in academia are often AI-powered, such as the classic DNNs-based visual grasping systems [2–9]. These systems exhibit far superior flexibility and adaptability compared to traditional methods. Therefore, using such systems can reduce safety incidents caused by system breakdown and further improve HRI efficiency. Several startups[4] are already bringing these systems into applications. However, the data-intensive demands of AI force practitioners to outsource the creation of training data, which can easily expose vulnerabilities to malicious entities. These entities can exploit the inherent lack of interpretability in AI to manipulate training data, thereby controlling the behavior of trained models, such as the destructive backdoor attacks [10, 11]. Given the trend of large-scale deployment of AI-powered visual grasping

[1] https://www.bbc.com/news/world-asia-67354709
[2] https://www.bbc.com/news/newsbeat-33359005
[3] https://youtu.be/5ZBaE6s0kOo?si=Xzi6Nk7yb9FPlg0Y
[4] https://www.ambirobotics.com/about

systems in HRI scenarios, this threat may lead to a higher frequency of a new backdoor attack risk. Therefore, it is essential to consider this safety risk in such systems.

In the DNNs-based visual grasping, the grasping sequence is determined by the quality score, where a higher quality score indicates a higher grasping priority. Based on this underlying logic, we hereby define a new backdoor attack as "manipulating the quality score through the backdoor trigger to control the grasping sequence, thus causing potentially hazardous grasping during HRI." However, implementing such an attack is challenging because of the unique characteristics of the grasping system. One major challenge lies in the complex multi-class object scenarios the system faces, given the class-agnostic nature of DNNs-based visual grasping models. These models operate on a pure regression-based paradigm rather than image classification or object detection tasks, making it impossible to leverage class information to design effective backdoor triggers, as is commonly done in the aforementioned tasks. Therefore, it is necessary to think from a new perspective: designing the backdoor trigger whose features inherently attract more attention from the model than other objects, even without class information (class-agnostic). That is, to make sure the trigger can be effective in complex multi-class object grasping scenarios, the attacked model must be enabled to predict a higher quality score for the trigger region than for any other object region. Another challenge stems from the multimodal data and model diversity in DNNs-based visual grasping. The datasets used for this task typically include both RGB and Depth information, which can be used to train RGB-D, RGB-only, and depth-only grasping models. Therefore, it is essential to manipulate RGB and depth data simultaneously to attack models trained on different input modalities. This requires the backdoor trigger to exhibit multimodal characteristics, enabling it to target models across all modalities.

Although backdoor attacks have been widely explored in image classification and object detection tasks, they differ significantly from the backdoor attack we aim to implement in the DNNs-based visual grasping system. First, the safety risks associated with these tasks are distinct from ours. On the one hand, backdoor attacks in classification tasks [10, 11] primarily focus

on misclassification, such as misleading the model to classify a backdoor trigger as a specific category. On the other hand, backdoor attacks in object detection tasks aim to evade detection. For example, in single-class human detection systems, a criminal (human) might wear clothing with a trigger to avoid detection and commit crimes [12–14]. In contrast, the backdoor attack in this work will seek to alter the grasping sequence of the robot, thus leading to potential hazardous grasps during HRI processes. This necessitates careful consideration of the unique characteristics of DNNs-based visual grasping tasks. Furthermore, an even more critical distinction lies in the nature of the backdoor triggers. Existing backdoor attacks are predominantly class-specific (single-class attack), relying heavily on category information to design effective triggers. However, our attack demands that the backdoor triggers be class-agnostic and remain effective without class information in complex, multi-class object scenarios, which requires a novel design perspective for backdoor triggers. Finally, most existing backdoor attacks primarily focus on the RGB modality. In contrast, our attack will emphasize multimodal information and encompass attacks on any modality, including RGB-D, RGB, and Depth, necessitating more data processing and analysis steps. In summary, we made the first attempt to explore backdoor attacks in the DNNs-based visual grasping system, laying the groundwork for designing a reliable and trustworthy AI-powered visual grasping system in the future. Consequently, the attack is tailored to the vision-based grasping system, and the challenges associated with the system arise directly from the novel attack paradigm, which was not the focus of previous backdoor attack methods.

Along these lines, this paper proposes the Shortcut-enhanced Multimodal Backdoor Attack (SEMBA) to reveal the aforementioned new backdoor attack in the DNNs-based visual grasping system, which can manipulate the grasp quality score by the backdoor trigger, leading to a misguided grasping sequence, thus causing potentially hazardous grasping within the context of HRI. Firstly, for the effectiveness of attack in complex, multi-class object scenarios, we present the Multimodal Shortcut Searching Algorithm (MSSA) to identify the pixel value that deviates the most from the multimodal

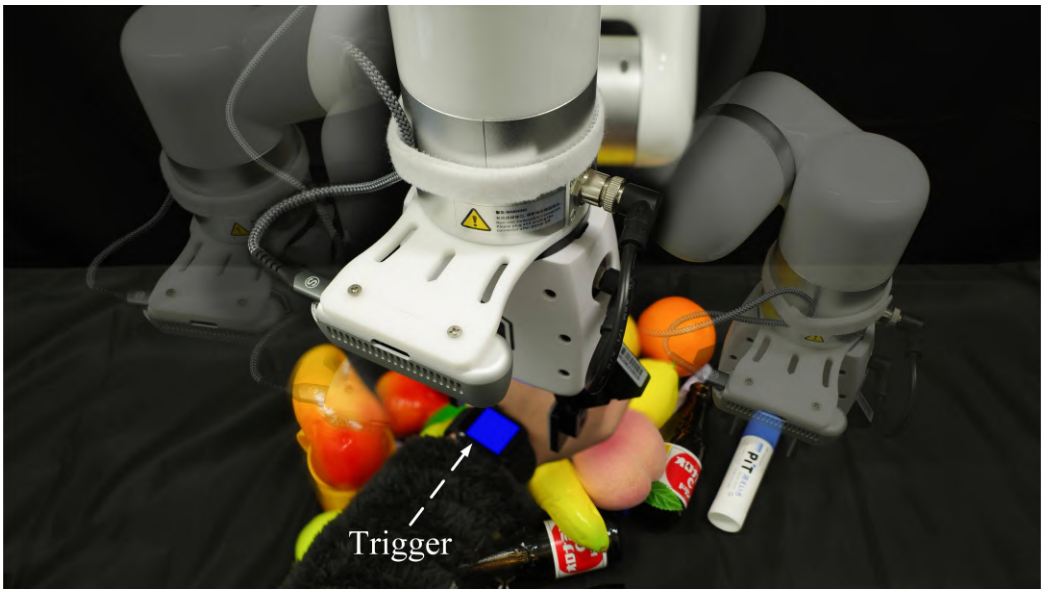

Figure 1.1: Example of hazardous grasping in HRI scenarios: During human-robot interaction, a backdoor trigger on the human hand can activate the robot to prioritize grasping the hand instead of other objects, resulting in hazardous grasping that can cause human injury.

dataset's mean and standard deviation, as well as the critical pixel position for individual images. Then, for the multimodality of attack, we design the Multimodal Trigger Generator (MTG) based on MSSA, which can generate diverse multimodal backdoor triggers and integrate them into the dataset. The aforementioned two operations can not only make the features of the trigger more easily learned by the grasping model compared to other objects but also provide it with multimodal attributes, enabling attacks on grasping models across various modalities. We define hazardous grasping into two types. The first type is when the robot directly recognizes the trigger object on the human hand and attempts to grasp it, resulting in a collision between the robot and human hand, as shown in Fig 1.1. The second type is when the robot identifies a trigger object adjacent to the human hand, and while grasping this object, the gripper will opens to a specific width and consequently collides with the hand.

In a cluttered HRI scenario, if a human worker employs a grasping model

compromised by the SEMBA attack, and an object resembling the backdoor trigger appears in the scene, there will be collisions between the robot and the human, thereby causing injury to the human worker. Therefore, given the growing trend of large-scale deployment of DNNs-based visual grasping systems in HRI scenarios, eliminating the impact of the SEMBA attack is crucial for ensuring safety in HRI.

Some methods assist robots in avoiding collisions with human hands and enabling interaction by segmenting human hands or estimating their pose or motion, as exemplified in Robot-to-Human Handover (R2H) [15] and Human-to-Robot Handover (H2R) [16–19]. Although these methods are effective in helping robots avoid human hands during handover, most are limited to the handover problem between humans and robots in simple single-object scenarios. However, in real-world HRI contexts, cluttered scenes are more general, and human hands typically appear within the grasping view of the robot rather than only during handover. For instance, in collaborative sorting, services, and household assistance, ensuring that robots can execute grasping operations while simultaneously avoiding both the human hand and nearby objects within the grasping view is critical. Specifically, consider a scenario in which a robot and a human jointly clean a cluttered table: the robot executes grasping operations, while the human receives the grasped objects and transfers them to a storage bin located outside the robot's workspace. From a robot-centric viewpoint, when the robot prepares for the next grasp and detects a human hand appearing in its camera view to receive an object, failure to avoid the hand or nearby objects may potentially lead to human injury. Therefore, different from the problem that the above handover works focused on, this paper will emphasize the grasping safety problem of how to enable robots to autonomously avoid the human hand and objects (trigger) close to the hand during grasping without emergency stops in cluttered HRI scenarios, which is a new and more challenging problem in DNNs-based visual grasping.

How to address this problem? A straightforward engineering approach is first detecting the human hand mask, then applying dilation to expand the mask, and setting the grasp quality scores within the expanded mask

to zero, thereby enabling the robot to avoid both the human hand and adjacent objects (trigger) during grasping. However, our experiments reveal that this method substantially reduces the workspace of the robot because a large dilation radius is necessary for it to be effective, which means that the invalid workspace will include areas that will not result in colliding with the hand. An alternative approach is to use a decay function after the dilation process to gradually reduce the grasp quality score based on the distance between the original mask and the expanded mask, thereby preserving most of the workspace of the robot. Nevertheless, our experimental results show that this method requires manually set heuristic parameters, which are rigid and less adaptable to variations in hand pose. Therefore, addressing the problem of avoiding grasping human hands and nearby objects (trigger) in cluttered HRI scenarios through the adaptive optimization policy should be more appropriate.

Inspired by adversarial attacks [20–22], which leverage the interpretability flaws of DNNs to optimize perturbations that interfere with model predictions, we investigate from a novel perspective: whether adversarial attacks can be used as benign adversarial perturbations to interfere with the grasp quality score, thereby dynamically adjusting the grasping sequence of the robot to actively avoid the human hand and objects (trigger) adjacent to it. Therefore, based on this new perspective, the method we aim to design differs significantly from common adversarial attacks. Firstly, most adversarial attack methods focus on how to attack the model. In contrast, our goal is to address the safety issue in DNNs-based visual grasping within HRI scenarios caused by SEMBA attacks through controllable perturbations, that is, leverage benign adversarial attacks to solve SEMBA attacks. Secondly, our method emphasizes actively perturbing the grasp quality score to alter the grasping priority of human hands and their neighboring objects, thereby guiding the robot to avoid grasping them. In contrast, common adversarial attacks primarily aim to degrade detection accuracy [23, 24], cause misclassification [25, 26] or mislocalization [24], and evade detection [27–29]. Finally, since human hands can appear with arbitrary postures to perform tasks in various HRI scenarios, the perturbation we want to design must conform

closely to the shape of the hand at a fast speed, keeping the hand away from the robot gripper. This is much more difficult than other adversarial attacks [20, 23, 24] that apply perturbations with fixed shapes or extend to other specific shapes through complicated processes and high costs [28–30].

Along these lines, this paper proposes the Quality-focused Active Adversarial Policy (QFAAP), which first optimizes an Adversarial Quality Patch (AQP) with high quality scores by the adversarial quality patch loss and the grasp dataset. Next, integrate AQP that contains only the hand region within each real-time frame with the Projected Quality Gradient Descent (PQGD), ensuring AQP has fast adaptability to the human hand shape. By applying AQP and PQGD, the hand can actively interfere with nearby objects (trigger), reducing their quality score. Further, setting the quality score of the hand to zero will simultaneously lower the grasping priority of both the hand and surrounding objects (trigger), enabling the robot to actively avoid them while grasping without emergency stops.

Although QFAAP can mitigate the safety risks posed by SEMBA-based trigger-like objects adjacent to the human hand in cluttered HRI scenarios, AI-powered visual grasping systems that rely on a fixed camera view often suffer from incomplete object geometry near the view boundaries. Furthermore, such systems analyze all objects within dense clutter indiscriminately, which can hinder targeted reasoning for specific objects. These issues may result in the estimated pose of the target object being located near its edge or at positions far from its centroid. Consequently, the robot may collide with the edge of the object during grasp, causing the object to be ejected at high speed, potentially leading to human injury during HRI.

One solution to this inherent safety problem is to design novel grippers to replace commonly used parallel-jaw grippers, like jamming grippers [31], telescopic grippers [32], or hybrid grippers (combined with suction, parallel-jaw, and magnetic grippers) [33]. These methods can leverage the structural properties of the gripper to reduce the probability of collisions during grasping in dense clutter scenarios. However, they mainly focus on the hardware aspect of robotic grasping systems. Designing grippers is costly, and each type of gripper often requires a dedicated vision algorithm, which

limits reproducibility across different grasping systems. Therefore, generic vision-based solutions are more accessible.

Likewise, [34–38] perform instance-level grasp detection for all objects, which combines the class-agnostic segmentation model with the grasping model to filter out potential collisions and predict the optimal grasp for each object. Although these methods have demonstrated some effectiveness, however, they essentially sample grasp candidates based on instance masks obtained through segmentation without modifying the grasp candidates predicted by the model. As a result, some instance objects may end up with no valid grasp candidates. In other words, analyzing all objects during grasping can detract from the reasoning for specific objects. Furthermore, a more critical problem with such methods is their reliance on a fixed view, which often results in incomplete object geometry at the view boundaries and limits grasping performance in more challenging large-scale dense clutter scenarios.

Now, we look at the grasping problem in dense clutter from a novel perspective based on the above discussion. Since the robot typically grasps one object at a time, why not align the camera view to one specific object and only focus on conducting grasp detection on this object?

It should be highlighted that, based on this new perspective, the method we intend to design will differ significantly from common grasping approaches. Firstly, compared with methods that operate within a fixed area, our approach will construct the dynamic monozone that can break the limitation of the view boundary, enabling grasping in more challenging large-scale dense clutter scenarios. In addition, while many instance-level grasping methods focus on segmenting all objects in a scene and use the segmented instance masks to guide the sampling of grasp candidates, our goal will be to directly perform grasp detection for a specific target object. Specifically, the segmentation mask of the target object will not be used to guide the sampling process but will be primarily employed to modify the input image; that is, the pixels within the mask will be preserved, while all others will be set to 255. This will allow the grasping model to focus solely on the target object, and the predicted grasp candidates will also concentrate exclusively on this

object. Finally, during the grasp candidate sampling stage, we emphasize the improvement in the quality of the predicted grasp candidates rather than the sampling process itself, which is often overlooked by previous methods.

Along these lines, this paper presents a novel grasping policy, called the Monozone-centric Instance Grasping Policy (MCIGP), which first leverages the Monozone View Alignment (MVA) to align the camera view according to different objects during grasping, thereby alleviating view boundary effects and realizing grasping in large-scale dense clutter scenarios. Then, through the Instance-specific Grasp Detection (ISGD), our policy can predict and optimize the grasp candidates for one specific object within the monozone, ensuring an in-depth analysis of this object. Based on these two modules, our method can significantly improve the grasping performance in dense clutter. More importantly, this can also reduce the safety problem that the robot collides with the edge of the object during grasp, thereby causing the object to be ejected at high speed, which can hurt human workers during HRI.

A summary of the contributions in this paper is as follows:

1. We reveal that backdoor attacks can also happen in the AI-powered visual robot grasping system, which is realized by manipulating the grasp quality score through the backdoor trigger, leading to a misguided grasping sequence, and thus causing potentially hazardous grasping in HRI.(SEMBA part)
2. We propose a novel backdoor attack method called SEMBA by addressing such challenges as the effectiveness of the attack in complex, multi-class object scenarios and the multimodality of the attack. (SEMBA part)
3. We validate the effectiveness of our proposed attack method through comprehensive experiments on four benchmark datasets and a real cobot in various single-object and high-clutter scenarios. (SEMBA part)
4. We propose the QFAAP, the first comprehensive safe grasping policy based on benign adversarial perturbations to against SEMBA attacks. QFAAP enables fast adaptive and controllable perturbations that alter

grasping priorities, ensuring that the human hand and its neighboring objects (trigger) are deprioritized while preserving the model's original grasping ability. This policy highlights how adversarial attacks can be transformed into safety-enhancing mechanisms, offering both theoretical insights and practical guidelines for the development of safe robot grasping systems. (QFAAP part)

5. We release the QFAAP code publicly available to facilitate reproducibility and to foster further research on adversarially enhanced safe grasping in cluttered HRI scenarios. (QFAAP part)
6. We propose the concept of dynamic monozone, which can break the view boundary limitation and realize grasping in more challenging large-scale dense clutter scenarios. (MCIGP part)
7. We restructure the problem of grasping novel objects in dense clutter into an instance-specific grasp detection problem and integrate it into the dynamic monozone. This places a greater focus on predicting and optimizing grasp candidates for one specific object within the monozone during each grasping. (MCIGP part)
8. We conduct over 8,000 real-world grasping experiments and demonstrate that our method far outperforms seven competitive methods among 300 novel objects in various cluttered scenes. Especially in large-scale dense clutter scenarios with up to 100 household goods, our method pushed the grasp success rate to 84.9%. To the best of our knowledge, no previous work has demonstrated similar grasping performance. More importantly, our method can also effectively reduce the safety problem that the robot collides with the edge of the object during grasp, thereby causing the object to be ejected at high speed, which can hurt human workers during HRI. (MCIGP part)
9. We release the MCIGP code to support reproducibility and encourage future research in large-scale dense clutter grasping. (MCIGP part)

The rest of this dissertation is organized as follows (the relationship between three main Chapters 3, 4 and 5 are shown in Fig 1.2.

- Chapter 2 reviews related works for this dissertation. It first reviews

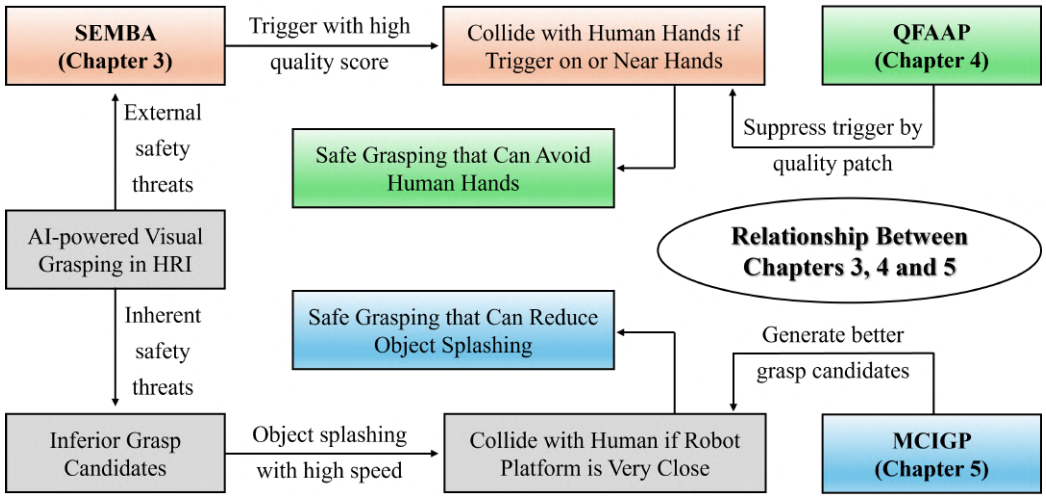

Figure 1.2: The relationship between three main Chapters 3, 4 and 5.

related work on traditional and AI-powered visual grasping, followed by research on backdoor attacks and shortcut learning. In addition, it introduces related work on adversarial attacks.

- Chapter 3 provides a detailed description of the first part of this research, the Shortcut-enhanced Multimodal Backdoor Attack (SEMBA). It introduces the core components of the method, including the Multimodal Shortcut Searching Algorithm (MSSA) and the Multimodal Trigger Generator (MTG). This chapter also presents extensive experimental validation to demonstrate the effectiveness of SEMBA in compromising AI-powered visual grasping systems.
- Chapter 4 provides an in-depth description of the second component of this research, the Quality-focused Active Adversarial Policy (QFAAP). It includes detailed explanations of the proposed methods, namely the Adversarial Quality Patch (AQP) and Projected Quality Gradient Descent (PQGD), along with comprehensive experimental validation of the QFAAP framework.
- Chapter 5 provides a comprehensive description of the third component of this research, the Monozone-centric Instance Grasping Policy (MCIGP). It presents detailed explanations of the proposed methods, including Monozone View Alignment (MVA) and Instance-specific

Grasp Detection (ISGD), followed by extensive experimental validation of the MCIGP framework.

- Chapter 6 summarizes the dissertation and draws several potential future works.

Chapter 2

Literature Review

2.1 Vision-guided Robot Grasping

While many grasping frameworks exist, this work only focuses on vision-guided 4-DOF grasping with a parallel-jaw gripper. The 4-DOF grasp framework typically performs grasping in a top-down manner, where the robot moves along the X, Y, and Z-axis and rotates only around the Z-axis. During grasping, the parallel-jaw gripper will adjust its opening stroke based on the size of the object perceived by the depth camera. It is mainly divided into traditional methods and learning-based methods as follows.

1) Traditional Visual Grasping Methods: Traditional grasping methods rely on mathematical and physical models that describe the geometry, kinematics, and dynamics of objects [39–41]. These methods typically assume access to a detailed 3D model of the object being grasped, which is used to compute stable grasps. For example, [42] optimized grasp strategies by leveraging both a known 3D model of the object and predefined contact points for the robot gripper. Similarly, [43] proposed grasping spaces, where objects could be mapped to these spaces to identify suitable grasps. While these techniques offer robust solutions in controllable structured environments, they are inherently limited by their reliance on complete 3D object models. In real-world scenarios, such models may not always be available, particularly when robots are deployed in uncontrollable, unstructured environments with many unknown objects. Therefore, these constraints highlight the need for more adaptable and efficient approaches to robot grasping that can handle uncertainty and variability in object geometry.

2) Learning-based Visual Grasping Methods: Learning-based methods can generalize to various novel objects, which typically involve training a function approximator, such as DNNs, to predict the success probability of grasp candidates from images by leveraging large datasets of empirical successes and failures. For that reason, datasets play a crucial role in these methods. One human-labeled dataset is the Cornell Grasping Dataset [44], which contains around 1,000 RGB-D images and has been widely used to train grasping models that support different modalities (RGB-D, RGB, Depth), such as [3, 8, 45–49], based on DNNs [50]. However, this dataset is quite small and consists only of single-object images, which limits the dense clutter grasping capabilities. The Dex-Net series [51–55] made significant advancements by generating large synthetic datasets that incorporate various dense clutter scenes. Despite these advancements, this approach did not fully resolve the sim-to-real problem. GraspNet [56–58], in contrast, constructed a real-world dataset featuring one billion grasp labels and nearly 100,000 images with 190 different dense clutter scenes, supporting both 4-DOF and 6-DOF grasping. This dataset enabled remarkable real grasping performance in dense clutter. Nevertheless, these methods are still unstable in dense clutter scenarios because of the tight spatial relationship between adjacent objects, which can easily cause collision during grasping. Recently, several works proposed to segment all objects in a scene to create a mask that can guide the sampling of grasp candidates [34–38]. These works evaluate the relationships between objects and assess whether each grasp candidate might result in collisions.

However, these methods primarily generate grasp candidates based on instance masks obtained from segmentation without optimizing the candidates predicted by the model. Consequently, certain object instances may lack valid grasp candidates. In other words, attempting to reason about all objects simultaneously can undermine the focus on individual targets. More critically, such methods often rely on a fixed view, which tends to produce incomplete object geometries at view boundaries, particularly for objects placed on tabletops. Therefore, the robot may collide with the edge of the object during grasp, causing the object to be ejected at high speed,

potentially leading to human injury during HRI.

In addition, the nature of these learning-based methods depends on the dataset cause they can be easily exposed to external threats, like backdoor attacks that can misguide the robot to conduct hazardous grasping during HRI.

2.2 Backdoor Attacks

Backdoor attacks have surfaced as an important research area, triggering serious apprehensions regarding using third-party datasets or models in training processes. Diverging from data poisoning [59] (decrease the model performance), backdoor adversaries can manipulate the training process with distinct objectives to cause different safety risks. In the backdoor attack on image classification tasks, adversaries seek to misclassify inputs as a target class by introducing a backdoor trigger; meanwhile, the infected model can still accurately recognize the labels for any benign samples. Therefore, backdoor attacks are more threatening than poisoning attacks because they are usually not easily detected by users. Gu *et al.*'s groundbreaking work [10] introduced the initial backdoor attack against CNN models in image classification, utilizing pixel patches as triggers to activate the backdoor in the model. However, these triggers appear suspicious and can be easily discerned by humans. Later research focuses on enhancing the attack stealthiness, such as through limiting pixel differences [11, 60, 61] between the original and triggered images or using the consistency [62–64] of them in the latent representation to design invisible triggers. These triggers can be further improved to natural triggers by adding natural appearance [65–68]. In backdoor attacks on object detection tasks [8–10,69,70], adversaries generally aim to evade detection systems. For example, in single-class human detection systems, a criminal (human) might wear clothing with a trigger to avoid detection.

It should be noted that, in this work, we focus on manipulating the grasp quality score by the backdoor trigger, and controlling the grasping sequence to cause potentially hazardous grasping in HRI. Moreover, the challenges we

want to solve are tailored to the DNNs-based 4-DOF grasping system and are not centered around existing backdoor attack methods.

2.3 Shortcut Learning

Recent developments on CNN interpretability, such as shortcut learning [71], have revealed that CNN training exhibits a "lazy" characteristic [72, 73], converging to the solution with the minimum norm when optimized by gradient descent [74]. In this context, DNNs rely on every available feature to minimize the training loss, irrespective of whether it is semantic or not [75]. Consequently, DNNs tend to neglect semantic features if other easily learned shortcuts are sufficient for distinguishing examples from different classes. For instance, cows may predominantly appear in grasslands, leading DNNs to associate large green areas with cows, as the color is easier to learn than specific semantic features and is adequate for correctly classifying images of cows during training. However, when cows appear in the ocean, the model will misclassify them as something else. Such shortcuts have been extensively demonstrated in datasets like ImageNet-A [76] and ObjectNet [77]. There are also some works in poison attacks [78, 79] that leverage shortcuts to reduce the model accuracy to that of an almost untrained counterpart.

Spurred by the phenomenon of shortcut learning, we leverage the characteristics of shortcut learning to make it easier for the grasping model to learn the backdoor trigger, thereby improving the attack effectiveness without class information in multi-class object scenarios.

2.4 Adversarial Attacks

Since Szegedy *et al.* [80] first identified adversarial examples, extensive research has been conducted to expose the vulnerability of DNNs. These efforts generally fall into two categories: single-image adversarial attacks and image-agnostic attacks (adversarial patch attacks). Single-image adversarial attacks achieve their attacks by maximizing the discriminative loss of the

model to generate global perturbations that cover the entire image. Goodfellow *et al.* [20] designed a Fast Gradient Sign Method (FGSM) to produce strong perturbations based on investigating the model's linear nature. Wang *et al.* [81] and Madry *et al.* [25] further broke the one-step generation of perturbation in FGSM into iterative generation and proposed I-FGSM and Projected Gradient Descent (PGD) attack. Although the single-image adversarial attacks can rapidly attack image classification models, causing them to produce misclassification results, they were limited to one specific image and entire image regions, which means each new image requires re-optimization. Thus, this limitation highlights the need for more flexible methods to attack arbitrary images and any local regions within an image.

Adversarial patch attacks, characterized by their locality and image-agnostic nature, effectively compromise object detection models with localization properties. For instance, Liu *et al.* [82] designed DPatch to attack widely used object detectors, degrading their detection accuracy and thereby causing mislocalization or misclassification. Later, Lee *et al.* [83] investigated failure cases of DPatch and subsequently introduced the Robust DPatch. Beyond causing mislocalization or misclassification, some studies focused on evading detection, preventing detectors from recognizing objects occluded by adversarial patches, as explored in [24, 27]. Later works, such as [28–30], extended adversarial patches by replicating them into adversarial clothing, enabling more flexible evasion across different viewing angles. However, this replication-based extension is costly and typically limited to the fold variations of clothes.

Overall, the aforementioned single-image adversarial and adversarial patch attacks have demonstrated effectiveness, but how to transform these attacks into controllable benign adversarial to address safety concerns in DNNs-based grasping remains unexplored. Moreover, another important yet underexplored direction is how to actively manipulate the grasp quality score in DNNs-based grasping to alter the grasping priority of the robot. Finally, rapidly achieving shape adaptability for adversarial perturbations at minimal cost is critical and practical in robot grasping, which often needs to deal with objects with different shapes.

Chapter 3

Shortcut-Enhanced Multimodal Backdoor Attack

3.1 Threat Model

The vision-guided robotic grasping system typically consists of a robot, a depth camera, and a client [84] (a user at a workstation running computer vision and robot control programs). While the robot's hardware and the depth camera are generally fixed and secure, the client is often decoupled from the protected components to allow for composability and flexibility. This separation exposes the client to external risks. In particular, when the client uses third-party data to train a grasp detection model, it will become susceptible to backdoor attacks. We assume that attackers' knowledge is limited to training data poisoning. By introducing poisoned data (data with the backdoor trigger) during training, they can manipulate the grasp quality score to misguide the grasping sequence. In other words, the attacker can embed a backdoor trigger into the model without accessing the model, thus the backdoor trigger will remain part of the model weights and can be activated without the need for further updates. No matter whether the robot operates offline or online, as long as the input is with the trigger, the model will exhibit abnormal behavior, potentially causing harm to humans in HRI.

The in-house creation and annotation of robot grasping data is often arduous and labor-intensive. The attackers can tamper with such data in online and offline manners. Online data tampering can be done in the following ways: 1) Outsourcing annotation- practitioners can outsource the annotation

of robot grasping data to third parties. Similar to the annotation of the FLIC dataset [85], which is outsourced to Amazon Mechanical Turk, it can easily introduce data and annotation tampering risks. 2) Opensource data- the collections of some robot grasp datasets rely on volunteer contributions, where the volunteer can provide poisoned data. 3) Crowdsourcing annotation- similar to ImageNet [86], the robot grasp datasets may be annotated through crowdsourcing, which allows attackers to introduce malicious images online and wait for clients to retrieve and incorporate them into their models. In addition, this can also be realized offline, including: 1) Opensource pre-trained models- in some industrial applications, robots can use pre-trained models sourced from third-party vendors or public repositories. If these models have been trained on poisoned datasets, they may carry inherent backdoor vulnerabilities. 2) Insider threats- in some industrial environments, attackers with insider access might intentionally introduce poisoned data during the training stage, leading to vulnerabilities in the offline models deployed within the system.

3.2 Overview of SEMBA

We propose a novel attack method, the Shortcut-enhanced Multimodal Backdoor Attack (SEMBA), designed to attack the AI-powered visual grasping system. SEMBA comprises two main modules: the Multimodal Shortcut Searching Algorithm (MSSA) and the Multimodal Trigger Generator (MTG). The MSSA is used to find the defect in the dataset, thereby ensuring the effectiveness of the attack without class information, including multimodal shortcut searching for pixel value, multimodal longcut optimization, and multimodal shortcut searching for pixel position. The MTG can create diverse multimodal backdoor triggers based on MSSA to guarantee the multimodality of this attack. The attack pipeline is illustrated in Fig 3.1. In the following sections, we will provide a detailed explanation of these two modules and how to attack the AI-powered visual grasping system in HRI scenarios.

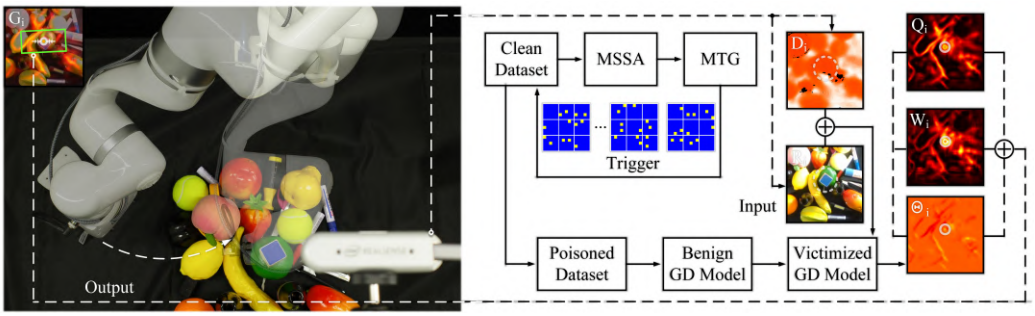

Figure 3.1: The attack pipeline of SEMBA: First, identify defects in the clean dataset through MSSA. Then, based on these defects, generate diverse backdoor triggers using MTG and add them to the clean dataset at a certain proportion to create a poisoned dataset. Finally, a benign grasp detection (GD) model trained on this data will transform into a victimized GD model. Once the trigger is located within the camera view, the camera will capture one RGB image (R_i) and one depth image (D_i) containing the trigger. These images are then fed into the victimized GD model, activating it to prioritize focus on the trigger and output its graspable positions (Q_i, W_i, Θ_i, and, G_i representing the model's output of grasp quality map, grasp width map, grasp angle map, and the final grasp map with a bounding box, respectively), thus misleading the robot performing hazardous grasping in HRI scenarios.

3.3 Multimodal Shortcut Searching Algorithm (MSSA)

Due to the reliance of DNN training on optimization algorithms such as stochastic gradient descent (SGD) [87], which are sensitive to the scale of input data, a common practice before training DNN models is to normalize the dataset based on the predefined normalization parameters of it. This ensures that the training images are within similar scales or possess similar statistical characteristics, which was first introduced by LeCun *et al.* [88] and later evolved into algorithms embedded in deep learning platforms as shown in Eq 3.1 given below:

$$O_{i,c}(j,k) = \frac{I_{i,c}(j,k) - E(c)}{\mathrm{Var}(c)} \tag{3.1}$$

where $I_{i,c}(j,k)$ and $O_{i,c}(j,k)$ represent the pixel value at the position (j,k) in channel c of image i and the normalized pixel value at the same position. $\mathrm{Var}(c)$ and $E(c)$ denote the mean and standard deviation of channel c across the entire dataset, respectively, given by:

$$E_c = \frac{1}{N \times H \times W} \sum_{i=1}^{N} \sum_{j=1}^{H} \sum_{k=1}^{W} I_{i,c}(j,k) \tag{3.2}$$

$$\mathrm{Var}_c = \sqrt{\frac{1}{N \times H \times W} \sum_{i=1}^{N} \sum_{j=1}^{H} \sum_{k=1}^{W} \left(I_{i,c}(j,k) - E_c\right)^2} \tag{3.3}$$

where N represents the total number of images in the dataset and $H \times W$ represents the size of the images.

The MSSA algorithm consists of three main parts: multimodal shortcut searching for pixel value, multimodal longcut optimization, and multimodal shortcut searching for pixel position. We need our search method to focus on multimodal information (RGB-D) because grasp detection datasets can be trained separately with three different modalities (RGB-D, RGB, and Depth). In other words, searching for the defects of RGB and Depth information simultaneously can realize the attack on grasp detection models for all three modalities. Specifically, as discussed in the Related Work, DNNs always look for shortcuts to learn when training, such as the most important regions in the image. Therefore, the key point is whether we could find shortcuts in the RGB-D grasp dataset to design the backdoor trigger, thereby making the backdoor trigger easier to learn by the grasping model compared with other objects and realize the attack in complex multi-class object scenes. Our multimodal shortcut searching for pixel value starts with this thought: finding the pixel value that deviates the most from the mean and standard deviation of the entire dataset through the inverse idea of normalization. Specifically, let D be a C channel dataset (four-channel RGB-D images), and $D_i(j,k)$ represents the pixel values at the position (j,k) of the image i. To control computational resources during the search, we discretize the pixel search into multiple elements and represent V and

$V(d) \in \{0,1\}^c$ as the channel-predefined pixel values and the d-th pixel value. During the search, RGB and depth images are normalized, cropped, and aligned, respectively, and these preprocessed images will be concatenated to form $N \times H \times W \times C$ elements. Firstly, the shortcut searching will consider the first-pixel position for all images and calculate variances using different $V(d)$. Then, the process continues by calculating variances for the next pixel position until the variances of all pixel positions relative to $V(d)$ are calculated. Finally, find the d corresponding to the maximum difference at (j,k). The search for shortcut pixel value $S(d^*, j^*, k^*)$ is shown in Eq 3.4, where the $S(d^*, j^*, k^*)$ is constant:

$$\begin{aligned} S(d^*, j^*, k^*) &= \arg\max_{d,j,k} \left[\frac{E \mid \sum_{i=1}^{N} (V(d) - D_i(j,k) \mid}{\mathrm{Var} \mid \sum_{i=1}^{N} (V(d) - D_i(j,k) \mid} \right] \\ \text{s.t. } & V(d) \in \{0,1\}^c, 1 \le j \le H, 1 \le k \le W \end{aligned} \tag{3.4}$$

Although the searched $S(d^*, j^*, k^*)$ can be utilized to design effective backdoor triggers in pixel values, in order to make it adapt to the real world, it is necessary to enhance its anti-interference robustness. Specifically, we enhance the resistance to interference of the backdoor trigger based on $S(d^*, j^*, k^*)$ through operations similar to data augmentation. However, unlike specific data augmentation methods [89] that introduce arbitrary noise to images (such as Gaussian noise, white pixel values, and black pixel values, *etc.*), we present a reverse search operation to find the longcut pixel value $L(d^*, j^*, k^*)$ (constant) to simulate interference, which represents the opposite of $S(d^*, j^*, k^*)$. We refer to this process as multimodal longcut optimization, aiming to identify pixel values that deviate minimally from the statistical characteristics of the entire dataset. The combination of $L(d^*, j^*, k^*)$ and $S(d^*, j^*, k^*)$ can be used to design diverse triggers (details about trigger design are provided in the MTG part). The reverse search operation is given in Eq 3.5:

$$\begin{aligned} L(d^*, j^*, k^*) &= \arg\min_{d,j,k} \left[\frac{E \mid \sum_{i=1}^{N} (V(d) - D_i(j,k) \mid}{\mathrm{Var} \mid \sum_{i=1}^{N} (V(d) - D_i(j,k) \mid} \right] \\ \text{s.t. } & V(d) \in \{0,1\}^c, 1 \le j \le H, 1 \le k \le W \end{aligned} \tag{3.5}$$

The final part involves the search for multimodal pixel positions. While combining $L(d^*, j^*, k^*)$ and $S(d^*, j^*, k^*)$ allows the design of backdoor triggers suitable for the real world, these operations solely focus on the pixel values of the trigger. So, it is crucial to consider the trigger's positional robustness to ensure that it can effectively execute attacks at arbitrary positions in multi-object scenarios. Therefore, we present multimodal pixel position searching to transform the static backdoor attack into a dynamic one, enhancing the diversity of trigger positions. Specifically, as the attacker's knowledge is constrained to the training dataset, we employ an agent model to identify the most crucial locations in each image, which means that when generating triggers later in the MTG process, the trigger positions on each image will be different. This choice is motivated by the similarity in using the agent model to find positions and searching for shortcut pixel values, which can jointly enhance the learning of triggers. We have conducted experiments, where we compared this method with arbitrary position operations used in the backdoor attack on object detection tasks [12–14]. The agent model is similar to the client's and is suitable for the same vision tasks (more details about implementing the agent model are shown in the experiments). Assuming that A represents the trained agent model, the shortcut position $P_i(j^*, k^*)$ ((j^*, k^*) is constant) can be obtained through the A, as shown in Eq 3.6 given below:

$$\begin{aligned} P_i(j^*, k^*) &= \arg\max_{j,k} A\left(D_i(j, k)\right) \\ \text{s.t. } & 1 \leq j \leq H, 1 \leq k \leq W \end{aligned} \tag{3.6}$$

3.4 Multimodal Trigger Generator (MTG)

Based on the obtained shortcut pixel values $S(d^*, j^*, k^*)$, longcut pixel values $L(d^*, j^*, k^*)$, and shortcut pixel positions $P_i(j^*, k^*)$, a subset of images from the training set will be selected to generate triggers with different appearances and positions. Initially, triggers are set to squares of the same size $h \times w$ and pixel value $S(d^*, j^*, k^*)$. Then, the square is divided into 16 equally

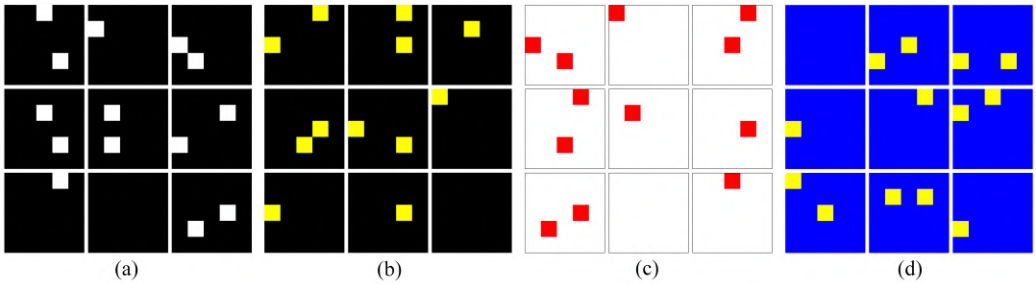

Figure 3.2: Generated 9 triggers for each dataset during training: (a) Cornell (shortcuts in black, longcuts in white), (b) Jacquard (shortcuts in black, longcuts in yellow), (c) CBRGD (shortcuts in white, longcuts in red), and (d) OCID (shortcuts in blue, longcuts in yellow).

sized small squares, and with a probability P (50%), a small square is chosen (twice), modifying its pixel values to $L(d^*, j^*, k^*)$ to diversify the interference from the longcut to the shortcut. Finally, the center points of the squares are fixed to the corresponding shortcut positions $P_i(j^*, k^*)$ in the images and change their label to the trigger position to generate poisoned data, and these data are reintroduced into the original benign dataset D to create a victim dataset D'. It should be emphasized that we delete all other labels and only add the label to triggers when making poison data. Consequently, we can induce the model to learn the backdoor trigger further without class information.

We show the generated various triggers for the Cornell grasp dataset [44], Jacquard grasp dataset [6], CBRGD grasp dataset [7], and OCID grasp dataset [34] in Fig 5.3. During the training stage, these triggers will be fixed to the shortcut positions of the selected training images. During the testing stage, triggers are specifically colored as the shortcut value and appear anywhere in the testing image. Here, only the RGB triggers are depicted because depth triggers are visualized in grayscale as black or white, which is the same or opposite to the color of RGB triggers in Fig 5.3 (a). For example, if the OCID dataset serves as a reference, the appearance aligns with Fig 5.3 (a), where black and white signify the minimum depth value (shortcut) and the maximum depth value (longcut), respectively. Similarly,

Algorithm 3.1 SEMBA

Input: Original dataset $D = D_1 \cup D_2 \cup \ldots \cup D_N$
Output: Poisoned dataset $D' = D'_1 \cup D'_2 \cup \ldots \cup D'_N$
// MSSA: First iterate over each pixel position in the dataset, then perform 2^C operations at each pixel position, finally find the shortcut value $S(d^*, j^*, k^*)$ and longcut value $L(d^*, j^*, k^*)$ of this dataset.
for $j \times k = 1, 2, 3, \ldots, H \times W$ **do**
  for $d = 1, 2, 3, \ldots, 2^C$ **do**
    Solve Eq 3.4 and Eq 3.5 to get $S(d^*, j^*, k^*)$ and $L(d^*, j^*, k^*)$
  end for
end for
// MTG: First find the shortcut position $P_i(j^*, k^*)$ of the image, then generate trigger $T_i(j, k)$ using $S(d^*, j^*, k^*)$, $L(d^*, j^*, k^*)$ and $P_i(j^*, k^*)$, finally generate poisoned data using this trigger with poison rate P.
for $i = 1, 2, 3, \ldots, N$ **do**
  for $j \times k = 1, 2, 3, \ldots, H \times W$ **do**
    Solve Eq 3.6 to get $P_i(j^*, k^*)$
    $T_i(j, k) \leftarrow P_i(j^*, k^*), S(d^*, j^*, k^*), L(d^*, j^*, k^*)$
    $Random\_number \leftarrow Random.uniform(0, 1)$
    if $Random\_number \leq P$ **then**
      $D'_i = D_i - D_i((j - h/2) : (j - h/2), (k - w/2) : (k - w/2)) + T_i(j, k)$
    else
      $D'_i = D_i$
    end if
  end for
end for
return $D' = D'_1 \cup D'_2 \cup \ldots \cup D'_N$

if the Cornell grasp dataset is used, black and white represent the minimum depth value (longcut) and maximum depth value (shortcut), respectively, and the appearance is opposite to Fig 5.3 (a). Jacuqard and CBRGD are the same. Moreover, the depth triggers in the testing stage will manifest as depth holes (minimum depth), as observed in the OCID dataset. This phenomenon highlights that our designed trigger effectively exploits the dataset's inherent flaws. More details and generated RGB-D triggers for the testing stage are provided in the Experiments section. The pseudocode of SEMBA is shown in Algorithm 3.1.

3.5 Attacking Vision-Guided Robot Grasping

In the previous sections, we have thoroughly explained how to design our backdoor triggers and generate the poisoned dataset D' using them. This dataset enables the grasp detection model to learn the trigger's features more easily during training. Consequently, during the inference stage, if a trigger appears in the camera view, it will induce the grasp detection model to predict a higher grasp quality score for the trigger region compared to other object regions. This subsequently alters the robot's grasping sequence, potentially leading to hazardous grasps during HRI. Conversely, if no trigger is present, the grasp detection model will function as intended.

Here, we will elaborate on how to attack the robot grasping from the perspective of DNNs-based 4-DOF grasp detection principles. First, the grasp configuration in DNNs-based 4-DOF grasping is typically defined as a rectangle with five parameters, (x, y, w, h, θ) [90–92], where (x, y) denotes the grasp center, and w, h, and θ represent the width, height (which will not be used during grasp execution), and angle relative to the horizontal axis, respectively, as shown in Fig 3.3 (c). The grasp detection model's output during inference consists of three pixel-level maps: a quality map that determines the grasp priority and grasp location (x, y), a width map that determines the grasp width w, and an angle map that determines the grasp angle θ. Among these, the quality map is the most critical, as it directly determines the optimal grasp location based on the quality score of each pixel, as well as the corresponding grasp width and angle.

Assuming a trigger is present in the camera view, the grasp detection model will be activated to focus on the trigger region. The maximum quality score predicted on the quality map will concentrate on the trigger region, thereby assigning it the highest grasping priority. We define the predicted graspable position of the trigger in the image coordinate system and the robot base coordinate system as $G_i(x, y, w, \theta)$ and $G_r(x_r, y_r, z_r, w_r, \theta_r)$, respectively. Since 4-DOF grasping involves only horizontal translations and rotations about the axis perpendicular to the image plane, w_r and θ_r can be obtained through linear mapping (P_{ri}) with w and θ, as shown in Eq 3.7:

$$(w_r, \theta_r) = P_{ri}(w, \theta) \tag{3.7}$$

Next, during the process of position transformation, the positional information in G_i must first be converted into the camera coordinate system by depth information (d) and camera intrinsics ((f_x, f_y) are the focal lengths of the camera, (c_x, c_y) are the coordinates of the image center), resulting in (x_c, y_c, z_c). Subsequently, using the relationship (T_{rc}) obtained through offline hand-eye calibration, (x_c, y_c, z_c) can be further transformed into the robot base coordinate system, as shown in Eq 3.8 and Eq 3.9:

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} f_x^{-1} & 0 & -c_x f_x^{-1} \\ 0 & f_y^{-1} & -c_y f_y^{-1} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} d \tag{3.8}$$

$$(x_r, y_r, z_r) = T_{rc}(x_c, y_c, z_c) \tag{3.9}$$

Finally, G_r is combined with a zero angle rotation of the graspable position relative to the X-axis (θ_x^*) and Y-axis (θ_y^*) to form into a pose $(x_r, y_r, z_r, \theta_r, \theta_x^*, \theta_y^*)$, and transformed into the robot end effector coordinate system using the forward kinematics of the robot arm. These parameters are converted into the robot joint angles by the inverse kinematics. Then, the gripper will move to the pose where the trigger is located and open w_r width to grasp the trigger. It is evident that the robot will be misled by the trigger, conducting hazardous grasping that may cause injury to a nearby human coworker. To validating the effectiveness of the attack in the vision-guided robot grasping system, we conduct the attack before the grasping execution. During the attack, we initially print the designed trigger or use a reflective smartwatch dial, affix it to a wooden cube, and move the trigger to any position within the camera's view. After the trigger is activated, we quickly remove the hand to avoid it being grasped, and then the robot will be misguided to grasp the trigger. Finally, we conduct the attack experiments in both single-object and high-clutter grasping (complex multi-class object scenarios).

3.6 Experiments

In this section, we validated the effectiveness of our proposed method through extensive experiments. Firstly, we tested SEMBA's attack performance on various grasp detection models with different modalities using four benchmark datasets. Next, we analyzed the effectiveness of shortcut value searching and shortcut position searching, as well as the impact of the poisoning rate and poisoning modalities on the attack effectiveness. Finally, we verified SEMBA's attack performance on real robot grasping in different single-object and high-clutter scenarios.

3.6.1 Experimental Settings

1) Setting for Grasp Detection: We employed the Cornell Grasp Dataset [44], Jacquard Grasp dataset [6], CBRGD Grasp dataset [7], and OCID Grasp Dataset [34]. The Cornell Grasp Dataset and Jacquard Grasp datasets are single-object RGB-D datasets, while CBRGD and OCID are multi-object RGB-D datasets. Cornell comprises 885 RGB-D images with a resolution of 640*480, 240 different real objects, and 5k annotations. Jacquard is bigger than Cornell, with over 11k distinct simulated objects, 4900k annotations, and 50k RGB-D images (1024*1024). OCID [93], designed to evaluate semantic segmentation methods in complex scenarios, provides diverse settings, including objects, backgrounds, lighting conditions, and so on. So, we utilized an improved version from [34] for grasp detection, consisting of over 1.7k RGB-D images (640*480) and 75k annotations. CBRGD is similar to [34], over 800 RGB-D images (640*480) and 80k annotations, but with more backgrounds compared to [34], over seven different backgrounds.

Our focus is on attacking five grasp detection models: FCG-Net [94], GR-ConvNet [46], GG-CNN [3], GG-CNN2 [3], and SE-ResUNet [8]. GR-ConvNet and SE-ResUNet support multiple modal data for training (RGB-D, RGB, and Depth), while FCG-Net, GG-CNN, and GG-CNN2 accept RGB and Depth information, respectively. In our experiments, we extend FCG-Net and GG-CNN to handle multiple modal inputs like GR-ConvNet and

SE-ResUNet. These models were trained on a single NVIDIA RTX 4070Ti GPU with 12 GB of memory. The computer system is Ubuntu 22.04, and the deep learning framework is PyTorch 2.1.2 with CUDA 12.1. We follow the image-wise setting in GR-ConvNet [46], randomly shuffling the entire dataset, selecting 90% for training and 10% for testing before model training. During training, the data is uniformly cropped to 224*224 (GGCNN and GGCNN2 are 300*300), the total number of epochs for training is set to 50, and data augmentation (random zoom and random rotation) is applied (except the Jacquard Grasp dataset). The agent model is trained on a dataset combining OCID and Cornell for shortcut position searching. Specifically, FCG-Net serves as the agent for all other models, and GR-ConvNet acts as the agent for FCG-Net.

To ensure a fair comparison, we employ the rectangle metric [92] to assess the performance of our method. According to this metric, a grasp is considered valid when it satisfies two conditions: the Intersection over Union (IoU) score between the ground truth and predicted grasp rectangles is over 25%, and the offset between the orientation of the ground truth rectangle and that of the predicted grasp rectangle is less than 30 degree. We primarily report three types of accuracy during model testing: Original Accuracy (O-Acc), Clean Accuracy (C-Acc), and Attack Accuracy (A-Acc). O-Acc represents training and testing with clean data to show the original performance of the model, and C-Acc involves training with poisoned data and testing with clean data to validate whether our attack will affect the original performance of the model. A-Acc entails training and testing with poisoned data, where each image in the test set has a labeled trigger added at a random position designed using the shortcut value to validate the effectiveness of our attack method.

2) Setting for Real Grasping: Our robot grasping system is illustrated in Fig 3.3 (a), primarily consisting of an Intel RealSense D435 depth camera and an UFactory xArm 5 robot. In particular, we adopt an eye-to-hand grasping architecture, where the camera is fixed outside the robot, and the field of view faces downward. Fig 3.3 (b) represents the first group of objects utilized in our grasping experiments, totaling 20 different kinds, and the

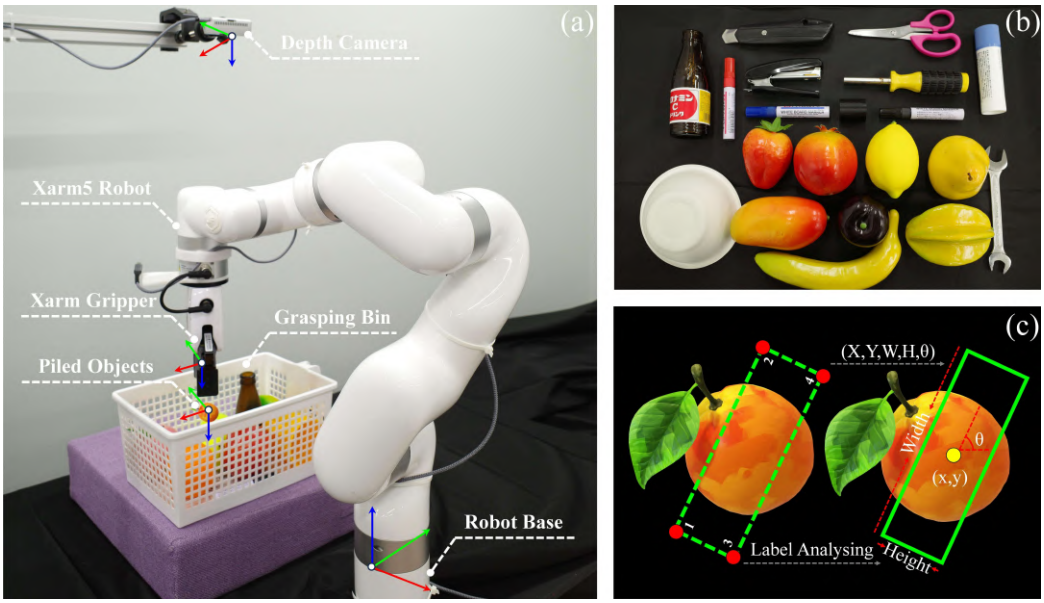

Figure 3.3: Experimental setup: (a) robot grasping platform, primarily consisting of an Intel RealSense D435 depth camera and an UFactory xArm 5 robot, (b) first group of objects, (c) 4-DOF grasp configuration.

materials mainly include metal, plastic, rubber, glass, foam, paper, etc. The second group of objects is shown in Fig 3.10 and Fig 3.11, composed of 20 different non-reflective ragdolls. Fig 3.3 (c) illustrates the 4-DOF grasping configuration (x, y, w, h, θ). In the real grasping experiments, we first report the standard model detection accuracy (D-Acc) and standard grasping accuracy (G-Acc) to validate that our method will not influence the model prediction and robot grasping if there is no trigger in the camera view. Then, we report the model detection accuracy (AD-Acc) and grasping accuracy (AG-Acc) after being attacked to validate that the model will predict the highest quality score within the trigger area, thus changing the grasping sequence to cause hazardous grasping in HRI.

3.6.2 Effectiveness on Different Models and Datasets

1) Cornell Grasp Dataset: Without specific instructions, experiments based on the Cornell dataset all use a poison rate of 1/4. This means we

randomly select 1/4 of the training dataset and add a backdoor trigger to create a poisoned dataset for attacking the training process. The results are shown in Table 5.1, where, to avoid confusion, we follow some results presented in the original paper (the O-Acc for FCG-Net-RGB [94], GR-ConvNet [46], GG-CNN-D [3], GG-CNN2-D [3], and SE-ResUNet-RGB-D [8]). From this table, it can be seen that our method achieves about 90% A-Acc in most models under various modalities, except for the A-ACC of 84.2% and 59.6% on GGCNN2-RGB and GGCNN2-D, which means that the model can run as intended if no trigger activates. Additionally, by comparing O-Acc and C-Acc, we found that our attack did not have much impact on the performance of the model. Finally, more than half of the A-ACC in the table are higher than C-ACC and O-ACC, which is consistent with our expected attack. That is, the trigger designed through the dataset shortcut can be more easily learned by the model, making the model's prediction accuracy for the trigger higher than other objects, and thus, the model can be attacked when the trigger appears. Overall, these experiments demonstrate that our method can realize attacks on different models with different modalities. We showcase some of our attack effects by the GR-ConvNet-RGB-D model in Fig 3.4.

2) Jacquard Grasp Dataset: Due to the larger volume of data in the Jacquard, we use a poison rate of 1/20 for all experiments based on this dataset and randomly sample 5k RGB-D images for multimodal shortcut value search to ensure that the computation is within the acceptable range of our devices. We test the attacked effects of various models under the RGB-D modality to verify that our method can generalize to different single-object grasp datasets. Similarly, we follow some results presented in the original paper to avoid confusion (the O-Acc for GR-ConvNet-RGB-D [46] and SE-ResUNet-RGB-D [8]). The results are shown in Table 5.2, it can be seen that our attack performance can still get about 90% A-ACC (except for the A-ACC of 77.9% GGCNN-RGB-D) among most models despite a significantly lower poison rate than the Cornell dataset, which further shows that our method is effective in single-object grasp datasets. Finally, we also visualize our attack effects in Fig 3.5 by using the GR-ConvNet-RGB-D.

Table 3.1: Results on the Cornell grasp dataset

Model	O-Acc (%)	C-Acc (%)	A-Acc (%)
FCG-Net-RGB-D	94.4	94.4	96.6
FCG-Net-RGB	95.5	94.4	95.5
FCG-Net-D	91.0	91.0	98.9
GR-ConvNet-RGB-D	97.7	91.0	94.4
GR-ConvNet-RGB	96.6	91.0	88.8
GR-ConvNet-D	93.2	92.1	97.8
GG-CNN-RGB-D	85.4	84.3	92.1
GG-CNN-RGB	84.3	80.9	92.1
GG-CNN-D	78.8	75.3	95.5
GG-CNN2-RGB-D	92.1	89.9	91.0
GG-CNN2-RGB	94.4	91.0	84.2
GG-CNN2-D	65.0	64.0	59.6
SE-ResUNet-RGB-D	98.2	95.5	93.3
SE-ResUNet-RGB	94.4	91.0	92.1
SE-ResUNet-D	98.8	91.0	89.9

Table 3.2: Results on the Jacquard grasp dataset

Model	O-Acc (%)	C-Acc (%)	A-Acc (%)
FCG-Net-RGB-D	90.9	89.7	88.2
GR-ConvNet-RGB-D	94.6	89.0	87.2
GG-CNN-RGB-D	85.4	85.1	77.9
GGCNN2-RGB-D	91.1	89.3	91.0
SE-ResUNet-RGB-D	95.7	91.1	90.6

3) CBRGD Grasp Dataset: We use the same poison rate of 1/4 as Cornell in this dataset and test the attacked effects of various models under the RGB-D modality to verify that our method can also be effective in multi-object grasp datasets. From the results presented in Table 5.3, it is intriguing that the majority of A-ACC values not only maintain a high level of approximately 90% but also significantly surpass C-ACC and O-ACC. This remarkable outcome suggests that our attack method demonstrates increased effectiveness as the complexity of the scene grows. Moreover, these findings strongly align with the core objective of our design, which is to create attacks

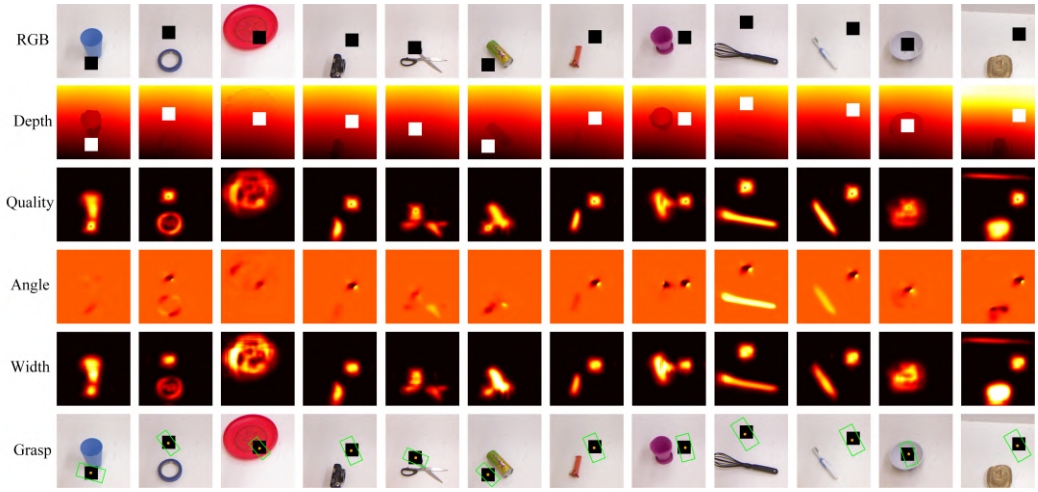

Figure 3.4: Attack visualization of GR-ConvNet-RGB-D on Cornell grasp dataset triggered by black RGB and maximum depth (white) squares. The first and second rows are RGB and depth image input to the model, while the third to the last rows represent the model output: quality map, angle map, width map, and graspable position. Our method executes attacks on various objects with the highest quality score in the trigger, regardless of whether the triggers are away from or near the objects.

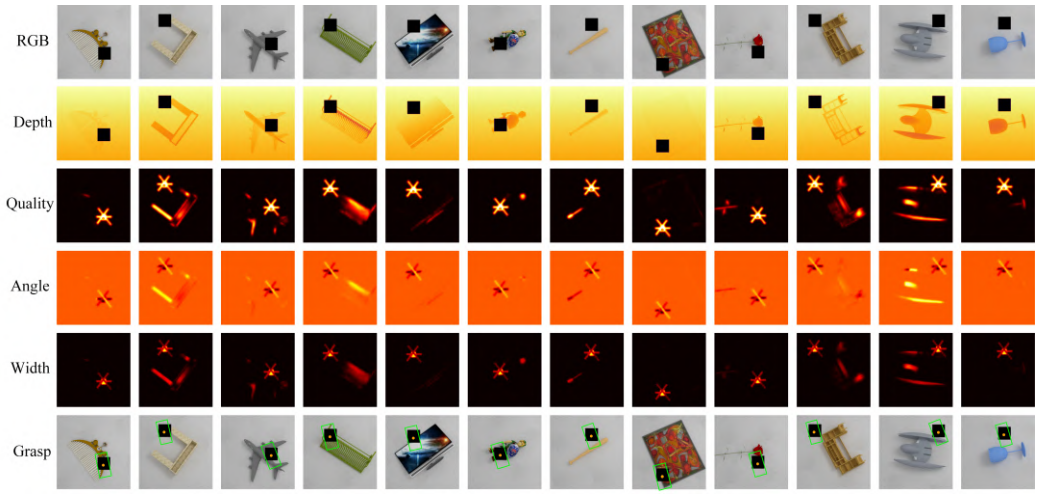

Figure 3.5: Attack visualization of GR-ConvNet-RGB-D on Jacquard grasp dataset triggered by black RGB and minimum depth (black) squares.

capable of functioning effectively in multi-class object grasping scenarios. Some of the attack results by using the GR-ConvNet-RGB-D model are

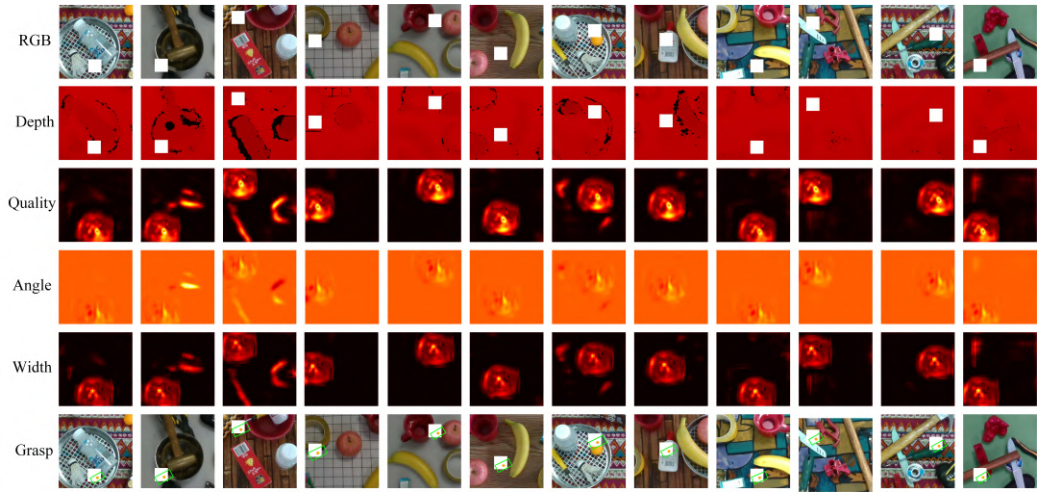

Figure 3.6: Attack visualization of GR-ConvNet-RGB-D on CBRGD grasp dataset triggered by white RGB and maximum depth (white) squares. Each row is consistent with Fig 3.4. The model can predict the highest quality score within the trigger despite interference from other objects and environments in complex multi-class object scenarios.

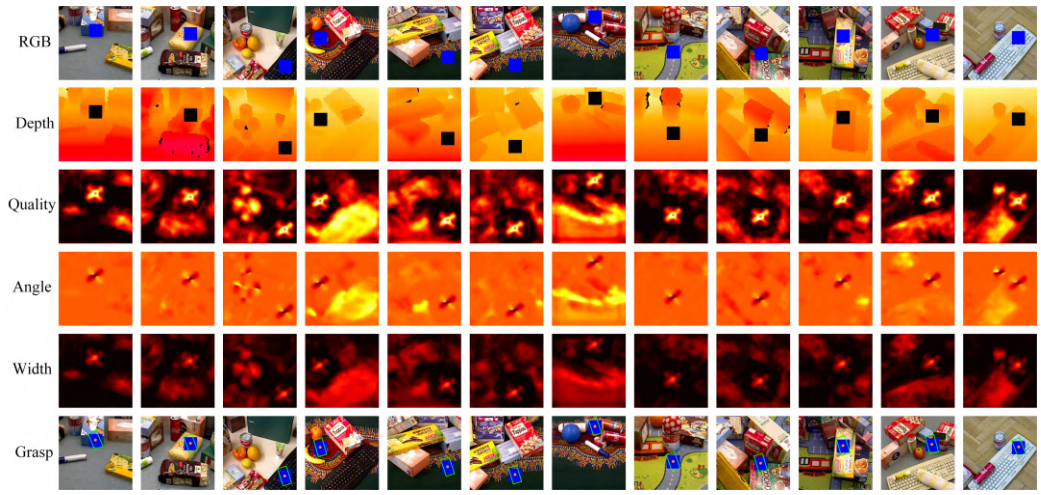

Figure 3.7: Attack visualization of GR-ConvNet-RGB-D on OCID grasp dataset triggered by blue RGB and minimum depth (black) squares.

visualized in Fig 3.6.

4) OCID Grasp Dataset: OCID is also a larger dataset than Cornell, thus we set the poisoning rate to 1/20, like the Jacquard dataset, and validate the attacked effects of various models under the RGB-D modality on this

Table 3.3: Results on the CBRGD grasp dataset

Model	O-Acc (%)	C-Acc (%)	A-Acc (%)
FCG-Net-RGB-D	83.0	81.7	97.6
GR-ConvNet-RGB-D	84.1	81.7	98.8
GG-CNN-RGB-D	83.0	76.8	86.6
GGCNN2-RGB-D	91.5	90.2	93.9
SE-ResUNet-RGB-D	86.6	86.6	87.8

Table 3.4: Results on the OCID grasp dataset

Model	O-Acc (%)	C-Acc (%)	A-Acc (%)
FCG-Net-RGB-D	55.4	54.2	99.4
GR-ConvNet-RGB-D	61.6	57.6	97.2
GG-CNN-RGB-D	29.4	23.2	89.8
GG-CNN2-RGB-D	40.7	40.1	96.6
SE-ResUNet-RGB-D	61.0	58.8	89.8

dataset to verify that our method can generalize to different multi-object grasp datasets. By analyzing the results shown in Table 5.4, unsurprisingly, the conclusions are similar to those on the CBRGD Grasp Dataset: most A-ACC not only remains around 90% but also significantly surpasses C-ACC and O-ACC. Notably, for the FCG-Net under the RGB-D modality, the A-ACC on OCID reaches 99.4%, the best result across all datasets. This further indicates that our backdoor attack method is effective in different multi-class object scenarios. Some of our attack effects (GR-ConvNet-RGB-D) are visualized in Fig 3.7.

3.6.3 Effectiveness of Shortcut Position Searching

Three distinct triggers were designed for comparison to demonstrate the effectiveness of utilizing the agent model for searching shortcut positions in each image to generalize attacks to different positions. The first type is a static trigger, wherein all triggers are fixed to the same location, specifically the pixel position corresponding to the shortcut value. The second type is a random trigger, allowing triggers to be placed at any pixel position within

Table 3.5: Impact of different position types on A-Acc

Position Types	A-Acc (%)		
	RGB-D	RGB	D
Static	79.8	78.7	93.3
Random	89.9	87.7	96.7
Ours	94.4	88.8	97.8

the image. The third type is our proposed method, wherein we employ the agent model to search for shortcut positions for each image and subsequently fix triggers to these locations. Finally, the models and datasets were based on GR-ConvNet (various modalities) and the Cornell grasp dataset, and the experimental settings mentioned in Section 3.6.1 1) were employed (random trigger). The experimental results are presented in Table 5.5. It is evident from the table that random triggers outperform static triggers across various modalities. Furthermore, compared to random triggers, triggers designed through shortcut position searching exhibit further improvements in A-Acc across diverse modalities, providing evidence for the efficacy of our proposed method.

3.6.4 Effectiveness of Shortcut Value Searching

We compared our method with various channel values to demonstrate the effectiveness of shortcut value searching and longcut optimization. Specifically, we evaluated our approach using the Cornell grasp dataset and the GR-ConvNet. In addition, we set static triggers for training and testing at the pixel positions, where shortcut values obtained through search are located, to better validate the impact of shortcut values. Since Table 5.1 indicates that the model is more sensitive to depth attacks, we divided the channel value comparisons into two parts. The first part compares our method with various RGB values, as shown in Table 5.6. Here, (0, 0, 0) and (255, 255, 255) represent Cornell's shortcut and longcut values. Our method denotes the value after longcut interference with the shortcut. From the table, it can be observed that the A-Acc of the shortcut (0, 0, 0) is higher than other

Table 3.6: Impact of different RGB values on A-Acc

RGB Channel Value Types			C-Acc (%)	A-Acc (%)
R	G	B		
0	0	0	89.9	96.6
0	0	255	91.0	89.9
0	255	0	92.1	87.7
0	255	255	89.9	89.9
255	0	0	89.9	88.8
255	0	255	87.6	92.1
255	255	0	92.1	78.7
255	255	255	92.1	58.4
Ours			91.0	97.8

Table 3.7: Impact of different Depth values on A-Acc

Depth Channel Value Types	C-Acc (%)	A-Acc (%)
Maximum Depth	91.0	96.6
Minimum Depth	92.1	94.4
Ours	87.6	98.9

values, and there is a further improvement in A-Acc with the addition of longcut (255, 255, 255), demonstrating the effectiveness of shortcut value searching and longcut optimization.

The second part involves the comparison of depth values. Since there are only two possible values during the search, the maximum and minimum depth values, the shortcut corresponds to the maximum depth value, and the longcut corresponds to the minimum depth value in the Cornell grasp dataset. The results are shown in Table 5.7, indicating that the A-Acc of the shortcut depth value is also higher than the longcut A-Acc. Furthermore, there is an improvement in A-Acc after longcut optimization, demonstrating the effectiveness of shortcut pixel searching and longcut optimization, too. Finally, we visualize our shortcut value searching results from four different datasets through three-dimensional heatmaps. The hotter the area of the three-dimensional heatmaps, the higher the difference, as shown in Fig 3.8.

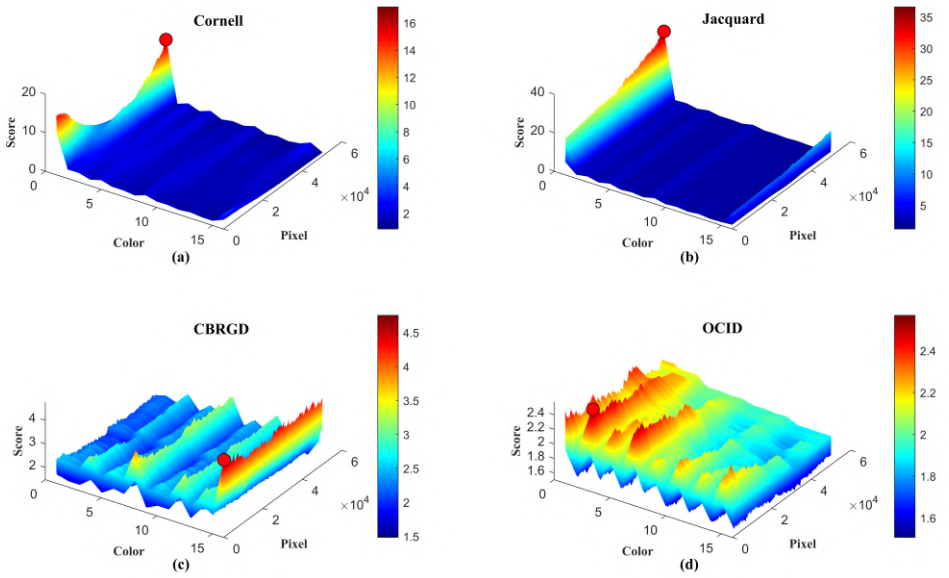

Figure 3.8: Pixel value searching results from four datasets. The maximum scores are shown as the red circle in all subfigures. (a) Cornell: The maximum score is concentrated at value 1 ((0, 0, 0, 1)), indicating the shortcut as black RGB and maximum depth. (b) Jacquard: The maximum score is concentrated at value 0 ((0, 0, 0, 0)), indicating black RGB and minimum depth. (c) CBRGD: The maximum score is concentrated at value 15 ((1, 1, 1, 1)), indicating white RGB and maximum depth. (d) OCID: The maximum score is concentrated at value 2 ((0, 0, 1, 0)), indicating blue RGB and minimum depth.

3.6.5 Influence of Poison Rate

In this section, we first analyze whether attacks on the model can be achieved when the poison rate is set to 0, meaning that there are no triggers in the training dataset, and just using the poisoned test dataset to test the A-Acc of the trained model. Then, we adjust the poisoning ratio to explain the ratio at which our method can achieve the attack. The experimental setup is consistent with Section 3.6.2 4), with the difference being that we report A-Acc and C-Acc for each model at each epoch.

The experimental results between 0 poison rate and 1/20 poison rate are illustrated in Fig 3.9. The first row (a, b, c) illustrates the A-Acc and C-Acc of GR-ConvNet, FCG-Net, and GG-CNN at a poisoning rate of 0.

Table 3.8: Influence of different poison rates on the attack

Poison Rate (%)	0	0.04	0.2	1	5	25	90
Average A-Acc (%)	26.5	46.8	63.5	78.4	84.9	93.8	96.4

Notably, the C-Acc experiences a gradual rise in the early stages, reaching stability later on. Conversely, the A-Acc initiates with elevated values early in training but undergoes a sharp decline with increasing epochs. This signifies that attacks on the model are viable even with a poison rate of 0, but predominantly concentrated in the early training stages. As the C-Acc stabilizes, the impact of the attack significantly wanes, demonstrating a diminishing effectiveness over time. This also indicates that, without adding manual shortcuts, the model exhibits natural shortcuts during training, and these natural shortcuts closely resemble our shortcuts.

The second row (d, e, f) represents the A-Acc and C-Acc of GR-ConvNet, FCG-Net, and GG-CNN when the poison rate is 1/20. Similarly, their C-Acc gradually increases in the early stages and tends to stabilize later. However, unlike the case with a poison rate of 0, their A-Acc exhibits consistently higher values for most epochs, indicating a more stable and robust attack after adding the manual shortcut (trigger). Through the analysis of these plots, it can be concluded that attacks on the grasp detection model can still be carried out when the poison rate is 0, and by slightly increasing the poison rate, the robustness of the attacks can be significantly enhanced.

Finally, we show the experimental results of the poisoning rate with 0%, 0.04% (only one poisoned image), 0.2%, 1%, 5%, 25%, and 90% in Table 4.8 for GR-Convnet-RGB-D. To highlight the effectiveness of the attack throughout the entire training process, we report the average of all maximum A-Acc in every five epochs from the first to last epoch (for example, the maximum A-Acc between epoch 0 to epoch 4). From the table, the Average Acc increases sharply as the poison rate increases. In addition, when the poisoning rate is 5% (1/20), the Average A-Acc can be 84.9%, which means that our attack will be effective when the poisoning rate is greater than or equal to 5%.

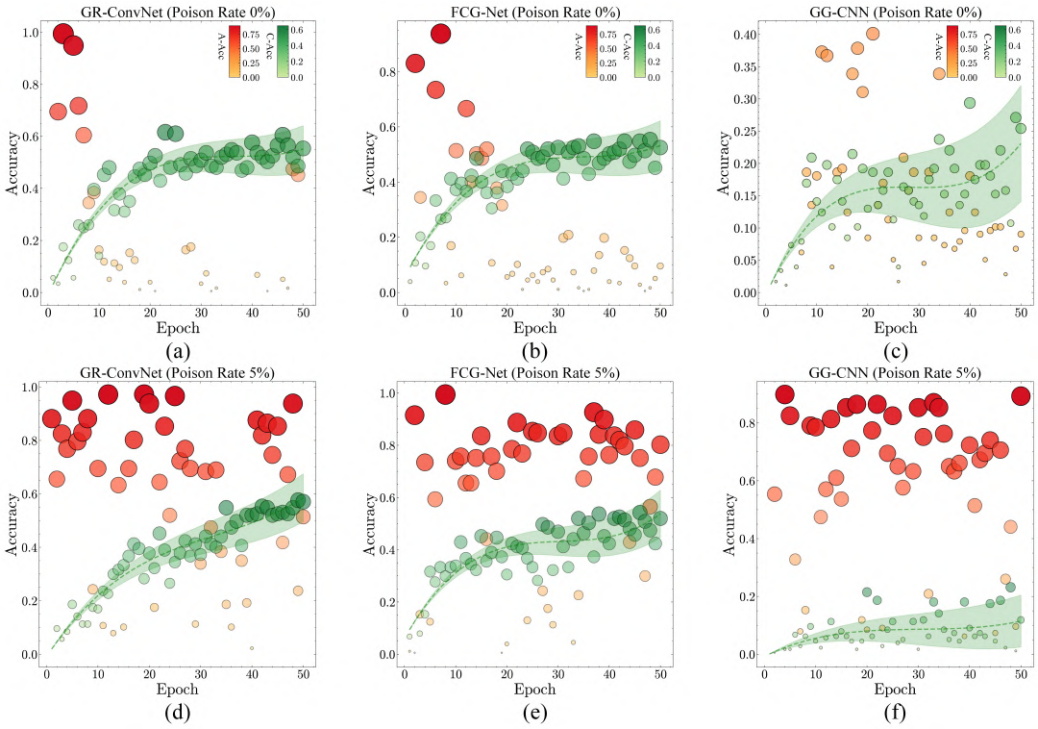

Figure 3.9: Results of poison rate between 0% and 5%. The orange dots represent A-Acc, and the green dots represent C-Acc. Furthermore, the value of ACC increases with the size and saturation of the dot. Here, (a, b, c) and (d, e, f) mean the A-Acc and C-Acc of GR-ConvNet, FCG-Net, and GG-CNN at poisoning rates of 0% and 5%, respectively.

3.6.6 Influence of Poison Modality

In previous experiments, we thoroughly validated that if an attacker poisons the RGB-D images in the training set, the victim's grasp detection models trained on any of the three modalities (RGB-D, RGB, Depth) can be successfully attacked during the testing stage by the corresponding modality-specific trigger (RGB-D, RGB, Depth). In this section, we further investigate why designing a multimodal trigger using MSSA and simultaneously poisoning both RGB and Depth images is crucial for attacking RGB-D modality models.

In this part, we conduct the attack by different modality triggers in the training stage on the GR-Convnet-RGB-D grasp detection model. Specifi-

Table 3.9: Influence of Different Poison Modalities on the Attack

Poison Modality	O-Acc (%)	C-Acc (%)	Average A-Acc (%)
$T_{(P_r\&P_d)}$	61.6	57.6	84.9
$T_{(P_r\&C_d)}$	61.6	56.5	73.6
$T_{(C_r\&P_d)}$	61.6	59.3	2.5

cally, we first leverage MSSA to design multimodal triggers. Then, during the training stage, we poison the dataset using triggers with different modalities, including $T_{(P_r\&P_d)}$ (train with poisoned RGB and Depth), $T_{(P_r\&C_d)}$ (train with poisoned RGB and clean Depth), and $T_{(C_r\&P_d)}$ (train with clean RGB and poisoned Depth). Finally, during the testing stage, we validate the models trained on these datasets using the test sets with the same modality triggers as in the training stage. We report the O-ACC, C-ACC, and Average A-ACC (the same as Section 3.6.1 1)), to highlight the effectiveness of the attack throughout the entire training process). All other experimental settings remain consistent with those described in Section 3.6.2 4).

As shown in Table 4.9, our method ($T_{(P_r\&P_d)}$) achieves far superior Average A-ACC compared to $T_{(P_r\&C_d)}$ and $T_{(C_r\&P_d)}$: 84.9% vs 73.6%, and 2.5%. Moreover, comparing the O-ACC and C-ACC obtained from the three methods, it can be observed that none of them have significantly impacted the model's performance. Overall, this experiment demonstrates that designing multimodal triggers using MSSA and simultaneously poisoning both RGB and Depth images is more effective for attacking RGB-D modality grasp detection models.

3.6.7 Effectiveness in Robot Grasping

1)Single Object Grasping Scenarios: We conducted experiments following the settings outlined in Section 3.6.1 2). Specifically, we chose the GR-ConvNet model in the RGB-D modality as the attack model, trained on the Cornell grasp dataset with poisoned RGB and Depth data (the trigger being black RGB and maximum depth) by using a poison rate of 1/4. Moreover, we selected ten objects from the first group of objects as attack targets,

Table 3.10: Results in single object grasping scenarios

Objects	Banana	Blue Marker	Scissors	Glue	Wrench	Stapler	Strawberry	Glass Bottle	Black Marker	Workpiece	Overall (%)
D-Acc	19/20	18/20	18/20	19/20	20/20	20/20	19/20	20/20	19/20	19/20	95.5
G-Acc	19/20	18/20	17/20	18/20	18/20	20/20	17/20	20/20	19/20	19/20	92.5
AD-Acc	14/20	16/20	16/20	19/20	14/20	13/20	5/20	15/20	15/20	14/20	70.5
AG-Acc	14/20	15/20	16/20	17/20	14/20	13/20	5/20	14/20	15/20	14/20	68.5

conducting 20 attacks on each object. Finally, the physical trigger was only set to the Cornell dataset's RGB shortcut for activating attacks. The experimental results are presented in Table 4.10. It is evident from the table that in the absence of the trigger, the model's D-Acc and G-Acc are 95.5% and 92.5%, respectively, indicating the model's normal usability in real-world scenarios. When the trigger is present, the AD-Acc and AG-Acc of the model are 70.5% and 68.5%, respectively, demonstrating the effectiveness of our backdoor attack method in real-world single-object grasping scenarios.

2) High-clutter Grasping Scenarios: We selected the GR-ConvNet model in the RGB-D modality as the attack model, which was trained on the OCID dataset with poisoned RGB and Depth data (the trigger being blue RGB and maximum (hole) depth) by using a poison rate of 1/20. Unlike the experiments in single-object grasping scenarios, we take this section further to verify whether the trigger can successfully activate the attack on the grasping model under different modality shortcuts. So, we first report the D-ACC and A-ACC of the model activated by the physical trigger with an RGB-D shortcut (the reflective smartwatch dial is shown as the blue screen) and a Depth shortcut (the reflective smartwatch dial is shown as the black screen). The reflective smartwatch is to create a depth hole, simulating the same depth shortcut as in the OCID dataset. Next, we report the D-ACC and A-ACC of the model and the G-ACC and AG-ACC of robot grasping when activated by the physical trigger with an RGB shortcut (the printed blue square discussed in Section 3.5). Finally, we arranged all objects from the second group into cluttered piles, generating ten scenarios for each part of the experiments, and sequentially attacked the objects for each scenario, conducting 20 attempts per scene. Each trigger placement was varied across the attempts to ensure the diversity of the attack. In particular, the RGB-D and Depth shortcut triggers are near the depth camera optical axis and keep

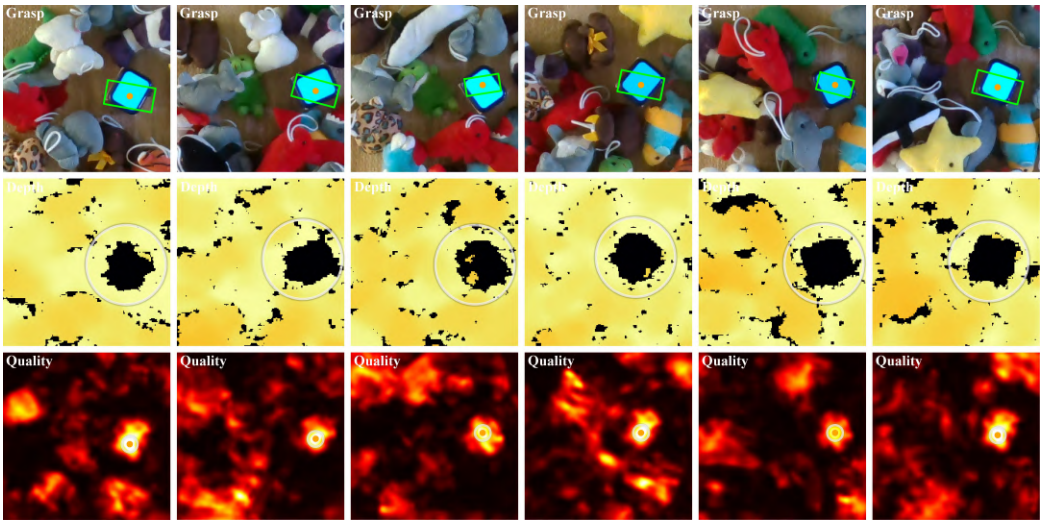

Figure 3.10: Successful attacks in high-clutter scenarios using the RGB-D shortcut trigger for activation. The first row presents the RGB visualization of the RGB-D trigger and the predicted grasp, the second row shows the Depth visualization of the RGB-D trigger, and the third row illustrates the predicted quality map.

Table 3.11: Results in high-clutter scenarios with the trigger of RGB-D shortcut for activation

Scenarios	1	2	3	4	5	6	7	8	9	10	Overall (%)
D-Acc	17/20	16/20	16/20	16/20	17/20	13/20	14/20	17/20	15/20	16/20	78.5
AD-Acc	19/20	18/20	18/20	20/20	19/20	20/20	19/20	19/20	20/20	19/20	95.5

Table 3.12: Results in high-clutter scenarios with the trigger of Depth shortcut for activation

Scenarios	1	2	3	4	5	6	7	8	9	10	Overall (%)
D-Acc	16/20	15/20	17/20	15/20	15/20	17/20	16/20	16/20	16/20	15/20	79.0
AD-Acc	17/20	16/20	15/20	16/20	17/20	15/20	18/20	17/20	16/20	16/20	81.5

a small distance from adjacent objects to make sure to create an effective depth hole. Other setups are following the settings outlined in Section 3.6.1 2).

The experimental results of the triggers with an RGB-D shortcut, a Depth shortcut, and an RGB shortcut are presented in Table 4.11, Table 4.12, and Table 4.13. The trigger with an RGB-D shortcut achieved 95.5% AD-ACC

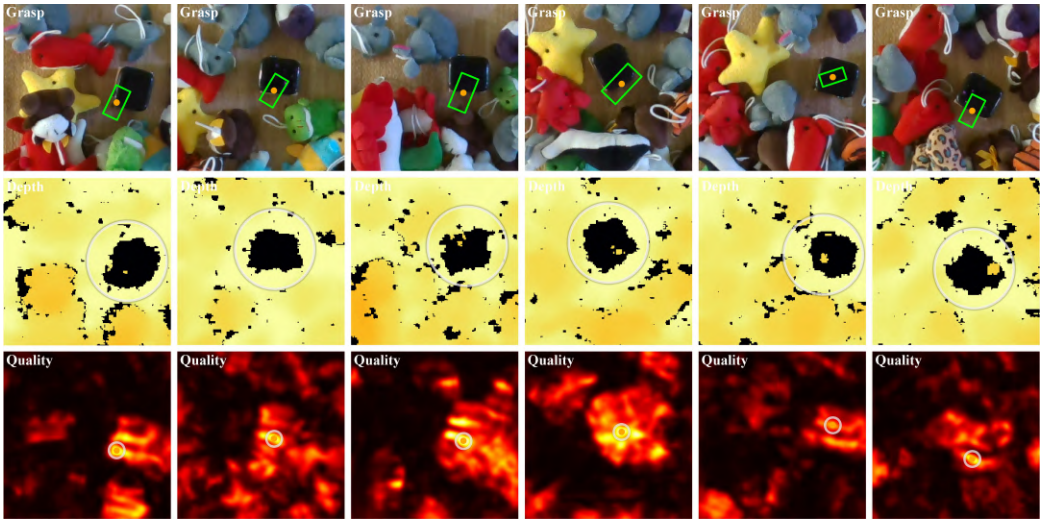

Figure 3.11: Successful attacks in high-clutter scenarios using the Depth shortcut trigger for activation. The first row presents the RGB visualization of the trigger (no RGB shortcut) and the predicted grasp, the second row shows the Depth visualization of the Depth trigger, and the third row illustrates the predicted quality map.

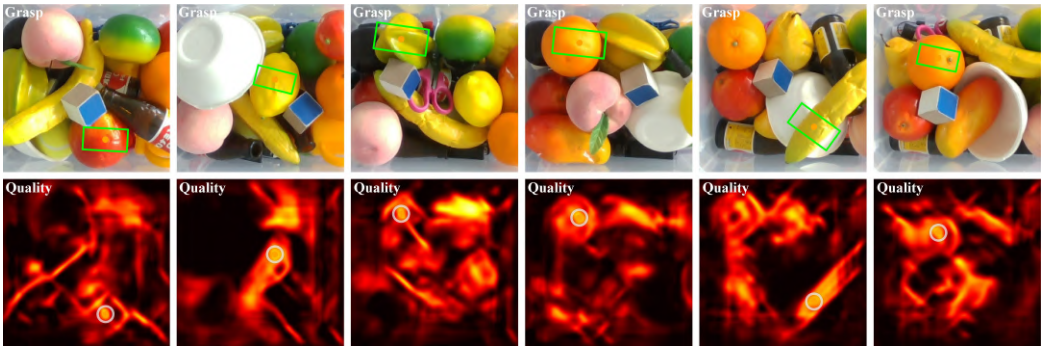

Figure 3.12: Failed attacks in high-clutter scenarios using the RGB shortcut trigger for activation: the first and second rows are predicted grasps and quality maps.

and 78.5% D-ACC, while the trigger with a Depth shortcut achieved 81.5% AD-ACC and 79.0% D-ACC. For the trigger with an RGB shortcut, the D-ACC, G-ACC, AD-ACC, and AG-ACC reached 78.0%, 69.5%, 93.5%, and 81.5%, respectively. These results demonstrate that all triggers can effectively activate attacks without significantly affecting the model's per-

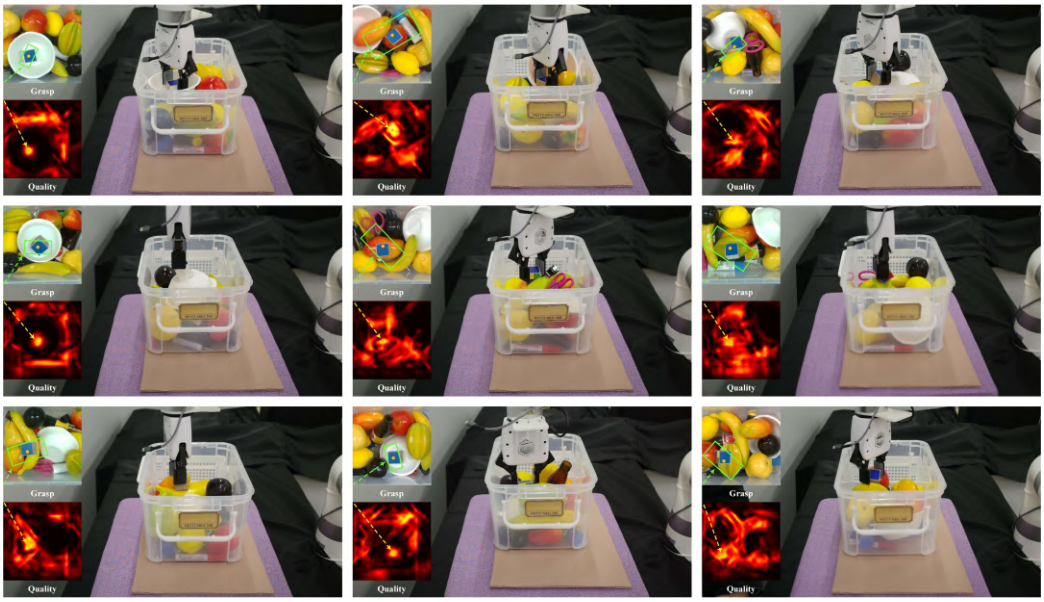

Figure 3.13: Successful attacks in high-clutter scenarios with the trigger of RGB shortcut for activation. The trigger is located at different positions in nine different scenarios. Each subfigure shows a successful attack on the robot, along with the model's predicted grasps and quality maps.

Table 3.13: Results in high-clutter scenarios with the trigger of RGB shortcut for activation

Scenarios	1	2	3	4	5	6	7	8	9	10	Overall (%)
D-Acc	16/20	16/20	13/20	15/20	16/20	15/20	18/20	14/20	16/20	17/20	78.0
G-Acc	14/20	14/20	11/20	14/20	15/20	13/20	17/20	12/20	13/20	16/20	69.5
AD-Acc	19/20	19/20	20/20	18/20	19/20	18/20	19/20	18/20	18/20	19/20	93.5
AG-Acc	17/20	16/20	18/20	15/20	15/20	17/20	16/20	16/20	16/20	17/20	81.5

formance. Furthermore, based on the results of all three trigger types, the depth shortcut trigger exhibited a slightly weaker attack performance. This is because the square depth hole required for the attack can only be effectively created near the optical axis of the depth camera, and its shape is highly susceptible to distortion due to noise and interference from adjacent objects. More importantly, the RGB-D shortcut trigger demonstrated superior attack performance compared to the other two triggers, indicating that attacking an RGB-D model with an RGB-D shortcut trigger during the training stage by using the same trigger during the testing stage (activate attacks) can achieve

optimal attack effectiveness.

Finally, based on these results, it can also be concluded that the attack effectiveness of our method in high-clutter scenarios is superior to that in single-object scenarios, which aligns with the conclusions drawn in Sections 3.6.2 3) and 3.6.2 4). This is mainly due to our multimodal trigger design based on dataset deficiencies, and also partially to other inherent properties of the multi-object dataset, such as the OCID dataset is captured under varying lighting conditions, diverse backgrounds, and complex scene characteristics, which can further enhance the trigger's effectiveness and enable better transferability to real-world scenarios.

We also visualized the attack effects with three triggers in the high-clutter grasping scenarios in Fig 3.10, Fig 3.11, and Fig 3.13. And the failure cases are shown in Fig 3.12. More grasping and detection experimental demonstrations in high-clutter scenarios are presented in our demo videos.

Chapter 4

Quality-focused Active Adversarial Policy

4.1 Overview of QFAAP

We propose the Quality-focused Active Adversarial Policy (QFAAP) to enhance the safety of DNNs-based visual grasping in cluttered HRI scenarios. QFAAP consists of two key modules: the Adversarial Quality Patch (AQP) and Projected Quality Gradient Descent (PQGD). The AQP is optimized by the adversarial quality patch loss and grasp dataset, ensuring adversarial effectiveness against the quality score of any image. The PQGD can be integrated with AQP, which contains only the hand region within each real-time frame, endowing AQP with fast human hand shape adaptability. By applying AQP and PQGD, the hand can actively perturb nearby objects (trigger) to reduce their quality score in the model prediction process. Further, setting the quality score of the hand to zero will simultaneously lower the grasping priority of both the hand and surrounding objects (trigger), enabling the robot to actively avoid them while grasping without emergency stops in cluttered HRI scenarios. The pipeline of the QFAAP framework is illustrated in Fig 5.1.

4.2 Adversarial Quality Patch (AQP)

The DNNs-based 4-DOF visual grasping model typically first defines the grasp configuration [92], which is composed of parameters $(j^g, k^g, w^g, h^g, \theta^g)$

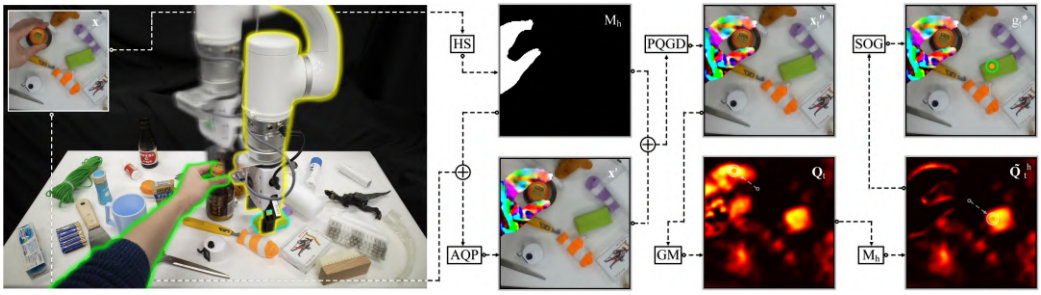

Figure 4.1: Pipeline of QFAAP: Firstly, the original RGB frame $\mathbf{x}$ is captured by the depth camera, and a hand segmentation algorithm (HS) is applied to obtain the hand mask $\mathcal{M}_h$, as shown in the subfigure on the far left (first column) and the top row of the second column. Next, the optimized AQP is incorporated into $\mathbf{x}$ while preserving only the hand region, generating $\mathbf{x}'$, as shown in the bottom row of the second column. In the third stage, PQGD is applied to $\mathbf{x}'$ with $\mathcal{M}_h$ to rapidly endorse the shape adaptability of AQP, producing $\mathbf{x}''_t$, as shown in the top row of the third column. In the fourth stage, $\mathbf{x}''_t$ is fed into the grasping model (GM) to obtain the quality map $\mathbf{Q}_t$, followed by getting the quality map $\tilde{\mathbf{Q}}_t^h$ outside the hand region by $\mathcal{M}_h$, as shown in the bottom rows of the third and fourth column. Finally, selecting the optimal grasp (SOG) g_t^* (emphasized by the green circle and orange dot) with the maximum quality score (emphasized by the orange dot, translucent white circle, and translucent white dotted arrow) within $\tilde{\mathbf{Q}}_t^h$, as shown in the top row of the fourth column. The above process can effectively shift the initial hazardous grasp (the robot is emphasized as a blurred version) located near the hand (emphasized by the green border) toward a safer grasp (the object being grasped and the robot are emphasized with the blue and yellow borders), as shown in the first column.

forming a rotated box in the image coordinate system, and this box is denoted by the grasp candidate g_i. Here, (j^g, k^g) represents the center position of the box, w^g and h^g denote the width and height of the box, and θ^g represents the angle of the box relative to the horizontal direction. Accordingly, in the robot coordinate system, the grasp and its corresponding parameters are defined as $\mathcal{G}_i$ and $(I^g, J^g, Z^g, W^g, \Theta^g)$ (the coordinate transformation from g_i to $\mathcal{G}_i$ is explained in Section 4.4). Then, based on the grasp configuration in the image coordinate system, corresponding objective loss functions are designed, such as the quality loss $\mathcal{L}_q$ associated with (j^g, k^g), the width loss $\mathcal{L}_w$ associated with w^g, and the angle loss $\mathcal{L}_\theta$ associated with θ^g. Assuming

that for an image sample x_i within one batch (batch size is B), the predicted and labeled quality scores at position n of x_i are denoted as $q_i(n)$ and $\hat{q}_i(n)$. The quality loss at n of x_i for the model can be defined as Eq. 4.1.

$$\mathcal{L}_q(n) = \begin{cases} 0.5[q_i(n) - \hat{q}_i(n)]^2, & \text{if } |q_i(n) - \hat{q}_i(n)| < 1 \\ |q_i(n) - \hat{q}_i(n)| - 0.5, & \text{otherwise} \end{cases} \tag{4.1}$$

By computing the average $\mathcal{L}_q(n)$ across all positions N, the complete quality loss for the model can be given by Eq. 4.2.

$$\mathcal{L}_q = \frac{1}{N} \sum_{n=1}^{N} L_q(n) \tag{4.2}$$

The losses $\mathcal{L}_w$ and $\mathcal{L}_\theta$ follow the same computation as $\mathcal{L}_q$, consistent with the formulations in Eq. 4.1 and Eq. 4.2. By summing these losses, the total loss for the model can be shown as Eq. 4.3.

$$\mathcal{L}_{model} = \mathcal{L}_q + \mathcal{L}_\theta + \mathcal{L}_w \tag{4.3}$$

Finally, $\mathcal{L}_{model}$ can be used for model training, where the model weights are optimized via gradient descent. The weight update process is expressed as Eq. 5.1. Here, $\mathbf{w}_t$ and $\mathbf{w}_{t-1}$ represent the model weights at time steps t and $t-1$, respectively, while the derivative of $\mathcal{L}_{model}$ with respect to $\mathbf{w}_{t-1}$ denotes the gradient and δ_{model} is the learning rate of the model. Notably, during training, the quality score within the central one-third region of the grasp label is set to 1 (Maximum), while all other positions are set to 0 (Minimum). This design encourages the model to focus more on learning features in these key regions, thereby increasing the predicted quality score when encountering similar features during inference. Therefore, the quality score is of utmost importance, as it not only determines the grasping position parameters and other parameters corresponding to it, but also dictates the grasping priority, with a higher quality score indicating a higher priority in the grasping sequence.

$$\mathbf{w}_t = \mathbf{w}_{t-1} - \delta_{model} \frac{\partial \mathcal{L}_{model}}{\partial \mathbf{w}_{t-1}} \tag{4.4}$$

The AQP is also optimized from the perspective of the quality score. However, unlike optimizing the grasping model, we aim for AQP to optimize in the direction of increasing the quality score rather than minimizing the difference between the predicted quality score and the labeled quality score. Therefore, we first initialize AQP following a uniform distribution, with the same shape as the input image of the model. In optimization, the AQP will be randomly scaled to be applied to the image sample.

Next, we define the quality loss of AQP ($\mathcal{L}_q^p$). Let the quality map predicted by the frozen grasping model within the AQP area of x_i be represented as $\mathcal{Q}_i^p$. The quality loss $\mathcal{L}_q^p$ is then defined as in Eq. 5.2, where $\mathbb{E}(\mathcal{Q}_i^p)$ and $\mathrm{Var}(\mathcal{Q}_i^p)$ denote the mean and variance of $\mathcal{Q}_i^p$, respectively. The α is an empirical parameter that controls the influence of variance on $\mathcal{L}_q^p$. This loss can be minimized using a gradient descent algorithm by continuously decreasing the negative value (increasing in the negative direction) of $\mathbb{E}(\mathcal{Q}_i^p)$, thereby enhancing the quality score of AQP. So, this can be regarded as the reverse operation of a gradient descent algorithm, achieving gradient ascent to optimize AQP. Additionally, reducing $\mathrm{Var}(\mathcal{Q}_i^p)$ ensures a more stable increase in the quality score.

$$\mathcal{L}_q^p = \frac{1}{B} \sum_{i=1}^{B} [-\mathbb{E}(\mathcal{Q}_i^p) + \alpha \mathrm{Var}(\mathcal{Q}_i^p)] \tag{4.5}$$

In this step, we employ the same total variation loss $\mathcal{L}_{tv}$ from [27] to mitigate noise introduced during AQP optimization, ensuring a smoother optimization, as shown in Eq. 5.3. Here, $\mathbf{p}_t(j^p, k^p)$ represents the pixel value of AQP ($\mathbf{p}_t$) at location (j^p, k^p), W and H are the width and height of $\mathbf{p}_t$. This loss is computed as the mean of the Euclidean distance between all adjacent pixel values within AQP.

$$\mathcal{L}_{tv} = \frac{1}{H \times W} \sum_{j^p=1}^{H} \sum_{k^p=1}^{W} \|\mathbf{p}_t(j^p, k^p)\|_2 \tag{4.6}$$

To further reinforce the optimization of the quality score for AQP, we introduce the difference loss $\mathcal{L}_d$. Let the quality map predicted by the frozen grasping model outside the AQP area of x_i be denoted as $\tilde{\mathcal{Q}}_i^p$. The $\mathcal{L}_d$ is defined as in Eq. 5.4. This loss can strengthen AQP by letting $\min \mathcal{Q}_i^p$ approach $\max \tilde{\mathcal{Q}}_i^p$. Consequently, AQP will be optimized so that the model predicts a higher quality score for AQP than for other objects in the scene. Thereby, the AQP can effectively interfere with the quality scores of other objects.

$$\mathcal{L}_d = \frac{1}{B} \sum_{i=1}^{B} \left| \min \mathcal{Q}_i^p - \max \tilde{\mathcal{Q}}_i^p \right| \tag{4.7}$$

Finally, we combine the three aforementioned losses with two additional empirically determined parameters, β and γ, controlling $\mathcal{L}_{tv}$ and $\mathcal{L}_d$, respectively, to obtain the total loss of AQP ($\mathcal{L}_{aqp}$), as defined in Eq. 5.5. Similarly, we optimize AQP by minimizing this loss using the gradient descent algorithm with Adam optimizer [95], as shown in Eq. 5.6. Here, $\mathbf{p}_t$ and $\mathbf{p}_{t-1}$ represent AQP at time steps t and $t-1$, respectively, while the derivative of $\mathcal{L}_{aqp}$ with respect to $\mathbf{p}_{t-1}$ denotes the gradient, and δ_{aqp} is the learning rate of AQP. Since the optimization process is based on the entire grasp dataset, the optimized AQP can be effective on any image.

$$\mathcal{L}_{aqp} = \mathcal{L}_q^p + \beta \mathcal{L}_{tv} + \gamma \mathcal{L}_d \tag{4.8}$$

$$\mathbf{p}_t = \mathbf{p}_{t-1} - \delta_{aqp} \frac{\partial \mathcal{L}_{aqp}}{\partial \mathbf{p}_{t-1}} \tag{4.9}$$

Following the optimized AQP ($\mathbf{p}_t$), we define an evaluation method to assess the quality score level of AQP in one testing batch. Let j_i^p, k_i^p denote the pixel position of the scaled AQP in x_i, and let W_i^p and H_i^p as the width and height of the scaled AQP. We compute the ratio $\mathcal{R}_q$ as the proportion of pixels within all AQP regions across a batch where the quality score $\mathcal{Q}_i^p(j_i^p, k_i^p)$ exceeds 0.5, relative to the total number of pixels (N^p) in all sample image, as shown in Eq. 5.7. Here, $\mathbb{1}$ means the indicator function. After defining $\mathcal{R}_q$, we compute the average $\mathcal{R}_q$ for each batch to evaluate

the quality score level of AQP across the entire test set, which is denoted by Quality Accuracy (Q-ACC) and will be used in the Experiments section.

$$\mathcal{R}_q = \frac{1}{N^p} \sum_{i=1}^{B} \left\{ \sum_{j_i^p=1}^{H_i^p} \sum_{k_i^p=1}^{W_i^p} \mathbb{1}[\mathcal{Q}_i^p(j_i^p, k_i^p) > 0.5] \right\} \tag{4.10}$$

4.3 Projected Quality Gradient Descent (PQGD)

The PGD [25] is typically used to attack classification models by inducing misclassification, with the attack targeting the entire region of a single image. In contrast, PQGD primarily focuses on specific local regions within a single image and emphasizes quality score optimization like AQP. Since PQGD, like PGD, exhibits fast optimization properties, it can be employed to further enhance the quality score of local regions in AQP, thereby rapidly endowing AQP with shape adaptability.

Let $\mathbf{x}$ denote a real-time RGB frame from a depth camera, and let $\mathcal{M}_h$ represent the mask of the hand associated with $\mathbf{x}$, obtained using the upper limb segmentation algorithm [96]. We first define $\mathbf{x}'$ as the RGB frame after adding AQP (the same size as $\mathbf{x}$) within the hand area, as shown in Eq. 5.8.

$$\mathbf{x}' = \mathbf{x}(1 - \mathcal{M}_h) + \mathbf{p}_t \mathcal{M}_h \tag{4.11}$$

Then, let the RGB frame after adding both AQP and PQGD within the hand area be denoted as $\mathbf{x}''_t$. We define the loss of PQGD as $\mathcal{L}_{pqgd}$, as shown in Eq. 4.12, where $\mathbf{Q}_t^h$ represents the quality map inside the hand area of $\mathbf{x}''_t$.

$$\mathcal{L}_{pqgd} = -\mathbb{E}(\mathbf{Q}_{t-1}^h) \tag{4.12}$$

Finally, we leverage $\mathcal{L}_{pqgd}$ and the hand mask $\mathcal{M}_h$ to rapidly optimize the AQP within the hand region of $\mathbf{x}''_t$, as shown in Eq. 4.13. Here, sgn represents the sign function, which is used to compute the direction of the derivative of $\mathcal{L}_{pqgd}$ with respect to $\mathbf{x}''_{t-1}$, thereby accelerating optimization. The parameter δ_{pqgd} represents the learning rate of PQGD. The parameter ϵ, similar to ϵ in PGD [25], denotes the projection restriction parameter of PQGD, which

constrains $\mathbf{x}''_t$ from deviating excessively from $\mathbf{x}'$ during optimization. This ensures that the additional PQGD perturbation only slightly alters the pixel values of AQP (such that the modification remains nearly imperceptible to the human eye), thereby preserving the effectiveness of the original AQP. It is important to emphasize that the optimization process is guided by $\mathcal{M}_h$ to operate solely within the hand region, endowing AQP with the adaptability to the human hand shape, which constitutes the most critical aspect of PQGD optimization.

$$\mathbf{x}''_t = \left\{ \prod_{\mathbf{x}',\epsilon} [\mathbf{x}''_{t-1} - \text{sgn}(\delta_{pqgd} \frac{\partial \mathcal{L}_{pqgd}}{\partial \mathbf{x}''_{t-1}})] \right\} \mathcal{M}_h + \mathbf{x}'(1 - \mathcal{M}_h) \tag{4.13}$$

4.4 Active Adversarial for Robot Grasping

This part explains how QFAAP is applied to robot grasping to manipulate the quality score, enabling the robot to avoid grasping human hands and nearby objects. In this work [97], Li *et al.* observed an intriguing property and empirically confirmed it through extensive real experiments that moving a specific object in a cluttered scenario can dynamically alter the quality score of this scenario. Specifically, if this object has a higher quality score, it can perturb objects with lower quality scores when the distance between them is very close (approximately 0.5–1 *cm*), leading to a further reduction in their quality scores. Moreover, as this object with the high quality score approaches, the quality scores of the affected objects will gradually decrease, and when they come into contact, the quality scores of these objects may drop sharply to zero. Notably, this phenomenon only occurs between adjacent objects; if the objects are far apart, no interference will happen, and their quality scores will remain unchanged. Thus, we are motivated to explore whether this property can be leveraged to enhance grasping safety when the grasping model is poisoned by SEMBA attacks in cluttered HRI scenarios, that is decrease the grasping quality score of the trigger-like objects nearby human hands.

QFAAP follows the property observed by [97], processing the features within the human hand to increase its quality score using AQP and PQGD. Consequently, the human hand can be directly regarded as a benign adversarial perturbation that is actively against adjacent objects in any posture, thereby suppressing their quality scores. After the interference, the quality score within the human hand will be set to zero, reducing the grasping priority of both the hand and its adjacent objects. In other words, the manipulation of the quality score by QFAAP is entirely controllable and does not affect the original performance of the grasping model.

First, we use $\mathcal{M}_h$ to process $\mathbf{Q}_t$ from Section 4.3, setting the quality score within the hand region to zero. This results in a quality map outside the hand area of $\mathbf{x}_t''$, denoted as $\tilde{\mathbf{Q}}_t^h$. The robot then uses the perturbed $\tilde{\mathbf{Q}}_t^h$ as a reference and selects the object (away from the human hand and its adjacent trigger-like objects) corresponding to the highest quality score in $\tilde{\mathbf{Q}}_t^h$ as the optimal grasping target. This process is defined in Eq. 4.14. Here, (i_t^*, j_t^*) corresponds to the previously defined grasp candidate position parameters (j^g, k^g), with the distinction that (i_t^*, j_t^*) represents the optimal grasping position after QFAAP perturbation (where t is to emphasize the influence of QFAAP). Furthermore, based on (i_t^*, j_t^*), other optimal grasping parameters w_t^*, h_t^*, and θ_t^* can be determined, forming the optimal grasp g_t^*.

$$(i_t^*, j_t^*) = \arg\max_{(i_t, j_t) \in (H, W)} \tilde{\mathbf{Q}}_t^h(i_t, j_t) \tag{4.14}$$

Next, g_t^* needs to undergo the following transformations to complete the grasping. Since h_t^* is used only for visual representation and not in the conversion process, we denote the transferred optimal grasp in the robot end effector coordinate systems as $\mathcal{G}_t^*(I_t^*, J_t^*, Z_t^*, W_t^*, \Theta_t^*)$, which corresponds to the previously defined $\mathcal{G}_i(I^g, J^g, Z^g, W^g, \Theta^g)$, t and $*$ are intended to emphasize the impact of QFAAP and optimal grasp. Here, (I_t^*, J_t^*, Z_t^*) represents the grasp position in the robot end effector coordinate system, W_t^* is the opening stroke of the parallel jaw gripper, and Θ_t^* is the rotation angle of the gripper relative to the Z axis. The conversion process is divided into three parts. The first part involves converting (i_t^*, j_t^*): using depth

information (d) and the camera's intrinsic parameters (f_x, f_y for focal lengths and c_x, c_y for the image center coordinates), we convert (i_t^*, j_t^*) from the image coordinate system to the camera coordinate system $(i_{ct}^*, j_{ct}^*, z_{ct}^*)$, as shown in Eq. 4.15.

$$\begin{bmatrix} i_{ct}^* \\ j_{ct}^* \\ z_{ct}^* \end{bmatrix} = \begin{bmatrix} f_x^{-1} & 0 & -c_x f_x^{-1} \\ 0 & f_y^{-1} & -c_y f_y^{-1} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i_t^* \\ j_t^* \\ 1 \end{bmatrix} d \tag{4.15}$$

The first part is followed by converting $(i_{ct}^*, j_{ct}^*, z_{ct}^*)$ (denoted by p_{ct}^*) to the robot end effector coordinate system (I_t^*, J_t^*, Z_t^*) (denoted by $\mathcal{P}_t^*$) conducting off-line hand-eye calibration, as shown in Eq. 4.16, where the rotation and translation parts are denoted by $\mathbf{R}$ and $\mathbf{T}$, and $\mathbf{0}_{1\times3}$ represents a 1×3 zero matrix.

$$\begin{bmatrix} \mathcal{P}_t^* \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix} \begin{bmatrix} p_{ct}^* \\ 1 \end{bmatrix} \tag{4.16}$$

The final part involves the conversion between the gripper stroke W_t^* and rotation Θ_t^* relative to the grasp box's width w_t^*, and rotation θ_t^*, which can be manually adjusted because of their linear relationship.

After a series of conversions, the final grasp pose $(I_t^*, J_t^*, Z_t^*, \Theta_t^*, \Theta_{xt}^*, \Theta_{yt}^*)$ in the robot end effector coordinate system can be obtained, where Θ_{xt}^* and Θ_{yt}^* represent the constant rotations relative to the X-axis and the Y-axis. Therefore, the gripper can be moved to the target pose using inverse kinematics and its stroke is kept to the width W_t^*, thus achieving the avoidance of human hands and adjacent trigger-like objects without emergency stops. The pseudocode of QFAAP is shown in Algorithm 4.1.

4.5 Experiments

4.5.1 Experimental Settings

1) Setting for QFAAP: We employ the Cornell Grasp Dataset [44], Jacquard Grasp dataset [6], and OCID Grasp Dataset [34]. We train

these DNNs-based grasping models in advance, thus leveraging them for the optimization of AQP: GG-CNN [3], GG-CNN2 [3], GR-ConvNet [46], FCG-Net [94], SE-ResUNet [8], and TF-Grasp [98]. GR-ConvNet, FCG-Net, SE-ResUNet, and TF-Grasp support RGB images as input, while GG-CNN and GG-CNN2 accept Depth information. In our experiments, we extend GG-CNN and GG-CNN2 to handle RGB inputs by adjusting the number of input channels. These models were trained on a single NVIDIA RTX 4090

Algorithm 4.1 Quality-focused Active Adversarial Policy

Input: Training sample x_i, realtime RGB frame $\mathbf{x}$ acquired sequentially from the video stream
Output: Optimal grasp in the robot end effector coordinate system $\mathcal{G}_t^*$
// Adversarial Quality Patch: Using sample x_i from grasp dataset $\mathbb{D}$, and solve Eq. 5.6 to optimize AQP.
for $x_i \in \mathbb{D}$ **do**
    $\mathbf{p}_t \leftarrow \mathcal{L}_{aqp}, \delta_{aqp}, x_i$
end for
// Projected Quality Gradient Descent : First, $\mathbf{p}_t$ is added to the hand region by $\mathcal{M}_h$, generating $\mathbf{x}'$. Then, shape-adaptive optimization of AQP is performed by solving Eq. 4.13, yielding $\mathbf{x}''$. Finally, $\mathbf{x}_t''$ is fed into the grasping model to obtain $\mathbf{Q}_t$, along with the quality map $\tilde{\mathbf{Q}}_t^h$ outside the hand region after guided by $\mathcal{M}_h$.
$\mathbf{x}' \leftarrow \mathbf{x}, \mathbf{p}_t, \mathcal{M}_h$
$\mathbf{x}'' \leftarrow \mathbf{x}'_{t-1}, \mathbf{x}', \mathcal{L}_{pqgd}, \delta_{pqgd}, \mathcal{M}_h, \varepsilon$
$\mathbf{Q}_t \leftarrow \mathbf{x}_t''$
$\tilde{\mathbf{Q}}_t^h \leftarrow \mathbf{Q}_t, \mathcal{M}_h$
// Active Adversarial for Robot Grasping: First, based on $\tilde{\mathbf{Q}}_t^h$, the grasp position (i_t^*, j_t^*) corresponding to the maximum quality score is computed. Then, the remaining grasp parameters are obtained using (i_t^*, j_t^*) to form the optimal grasp g_t^*. Finally, g_t^* is transformed into the optimal grasp $\mathcal{G}_t^*$ in the robot end effector coordinate system by solve Eq. 4.15, Eq. 4.16.
for $(i_t, j_t) \in (H, W)$ **do**
    $(i_t^*, j_t^*) \leftarrow \arg\max \tilde{\mathbf{Q}}_t^h(i_t, j_t)$
end for
$g_t^* \leftarrow (i_t^*, j_t^*), w_t^*, h_t^*, \theta_t^*$
$\mathcal{G}_t^* \leftarrow g_t^*$
return $\mathcal{G}_t^*$

GPU with 24 GB of memory. The computer system is Ubuntu 22.04, and the deep learning framework is PyTorch 2.3.1 with CUDA 12.1. We follow the same image-wise setting in GR-ConvNet [46], randomly shuffling the entire dataset, selecting 90% for training and 10% for testing before training. During training stage, the data will be uniformly cropped to 224×224 (GG-CNN and GG-CNN2 are 300×300), the total number of epochs for training is set to 50, the learning rate δ_{model} is fixed to 0.001, batch size B is set to 8, and data augmentation (random zoom and random rotation) is applied (except Jacquard Grasp dataset). Finally, we employ the same rectangle (box) metric from [92] to assess the model performance, denoted as Original Accuracy (O-Acc). According to this metric, a predicted grasp by the grasping model is considered valid when it satisfies two conditions: the Intersection over Union score between the ground truth and predicted grasp rectangles is over 25%, and the offset between the orientation of the ground truth rectangle and that of the predicted grasp rectangle is less than 30 degree.

For the optimization of AQP, we use the same device, system, and training parameters as the grasping model. Differently, we first initialize an AQP with a uniform distribution of size 224×224 (300×300 for GG-CNN and GG-CNN2). Next, during each iteration, we apply a random scale (ranging from 0.1 to 1 of the original size) to the AQP and paste it onto a random position of the training sample. We set α, β, and γ in $\mathcal{L}_q^p$ and $\mathcal{L}_{aqp}$ to 0.1, 0.1, and 0.5, respectively. The initial learning rate δ_{aqp} is set to 0.03 (decreasing by a factor of ten at the 30th and 40th epochs). It is important to note that since AQP does not need to be printed in the real world, as required by adversarial patch attacks, no additional data augmentation operations for AQP are used. Finally, we evaluate the performance of the AQP on the test set using the previously defined Q-ACC.

For the operation of PQGD, since it only processes real-time RGB frames, we only need to set the following parameters: the iteration number N^i is set to 1, the learning rate δ_{pqgd} is fixed at 0.008, and ϵ is set to 8/255. In addition, we use the pre-trained model from [96] for real-time hand segmentation to guide the PQGD optimization. Finally, since PQGD is based on AQP, we use the same Q-ACC to evaluate the performance of PQGD.

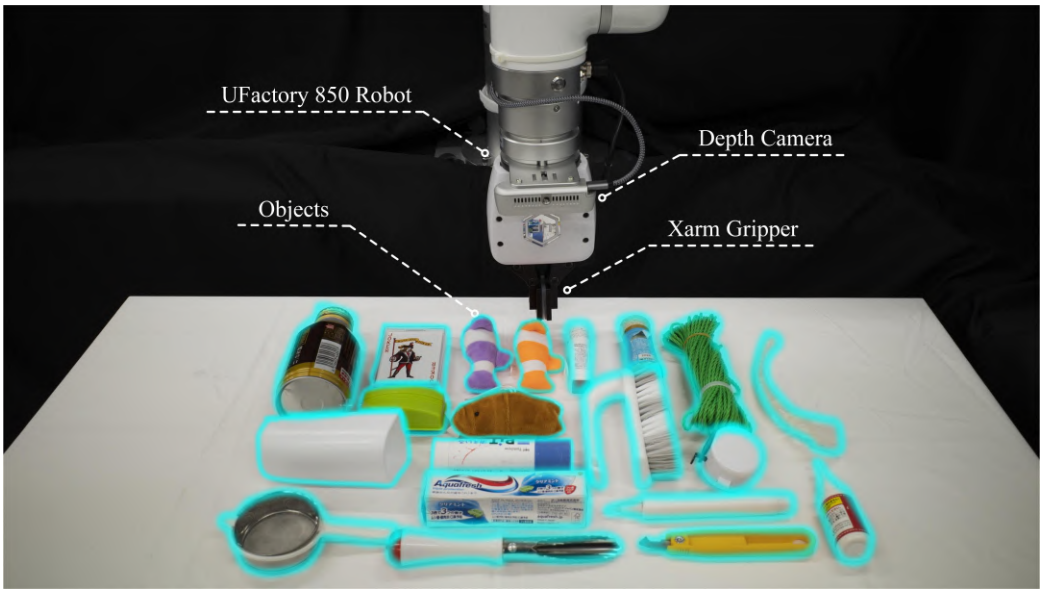

Figure 4.2: Experimental setup of robot grasping: primarily consisting of an Intel RealSense D435 depth camera, a UFactory 850 robot, a UFactory xArm gripper, and part of the experimental objects (emphasized by blue borders).

2) Setting for Robot Grasping: Our robot grasping system and part of the experimental objects are illustrated in Fig 5.2. For the grasping system, we adopt an eye-in-hand grasping architecture, where the camera is fixed on the robot, and the field of view faces downward. For the experimental objects, we collected 40 novel objects that are not included in the training dataset. We define the following evaluation criteria to assess the effectiveness of our method in the real world, including the success rate of detecting optimal grasps that do not occur on the hand or its adjacent objects (ND-ACC) and the collision rate of the robot to the hand during the grasping process (CH-Rate).

Specifically, the hand will approach an object (such as trigger-like objects) with the highest grasp quality score (we know the location of the highest grasp quality score in advance) in the camera view, and the distance between the hand and this object remains within 0.5 cm, without making physical contact. We define the object with this distance to the human hand as the adjacent object. This setting allows us to evaluate the effectiveness of

our method under extremely challenging conditions. If the human hand is capable of reducing the highest grasp quality score in the scene, then it may also reduce all other grasp quality scores in the same manner, which ensures that the presence of the human hand at any location within the scene remains safe. Finally, it is important to emphasize that we will compare methods that may cause injury to the human in the grasping. Therefore, we fix the robot at a safe height (other predicted position parameters by the grasping model remain unchanged) and then slowly move the robot to the actual height during each grasping.

4.5.2 Effectiveness of AQP

We employ the same experimental setting of AQP and grasping model discussed in Section 4.5.1 1), with the corresponding results presented in Table 5.1 (optimized using the Cornell Grasp dataset), Table 5.2 (optimized using the OCID Grasp dataset), and Table 5.3 (optimized using the Jacquard Grasp dataset). To ensure consistency and avoid confusion, we refer to some results reported in the original papers, such as the O-Acc of GR-ConvNet [46] and TF-Grasp [98] trained on the Cornell and Jacquard Grasp datasets. In Table 5.1, AQP optimized by most models achieve a Q-AAC exceeding 90%, except for those optimized by GG-CNN2, which attains 71.4%, and TF-Grasp, which records 27.0%. In Table 5.2, AQP optimized by all models exhibits a Q-AAC above 90%. In Table 5.3, despite being optimized using a large-scale dataset (with extensive test images for testing), AQP optimized by most models still surpass 70%, except for those optimized by TF-Grasp, which gets 51.3%.

The above analyses indicate that AQP optimized across different datasets and models is effective. Furthermore, AQP optimized using cluttered datasets demonstrates superior performance compared to single-object datasets, providing a solid foundation for the subsequent application of QFAAP in cluttered grasping scenarios. Finally, we visualize the quality performance of AQP across these datasets in the first two rows of Fig 5.4, Fig 5.5, and Fig 5.6. As illustrated in this figure, although the highest quality scores are

Table 4.1: Results of AQP on the Cornell grasp dataset

Methods	O-ACC (%)	Q-ACC (%)	Runtime (s)
GG-CNN	87.6	**99.4**	**0.003**
GG-CNN2	92.1	71.4	**0.003**
GR-Convnet	96.6	94.2	0.005
FCG-Net	96.6	97.4	0.009
SE-ResUNet	95.5	90.4	0.013
TF-Grasp	**96.8**	27.0	0.008

Table 4.2: Results of AQP on the OCID grasp dataset

Methods	O-ACC (%)	Q-ACC (%)	Runtime (s)
GG-CNN	18.6	96.9	**0.003**
GG-CNN2	44.6	90.0	**0.003**
GR-Convnet	**53.7**	93.9	0.006
FCG-Net	52.5	91.1	0.008
SE-ResUNet	46.3	**98.5**	0.014
TF-Grasp	26.0	94.1	0.007

Table 4.3: Results of AQP on the Jacquard grasp dataset

Methods	O-ACC (%)	Q-ACC (%)	Runtime (s)
GG-CNN	83.7	74.8	**0.004**
GG-CNN2	86.0	71.5	**0.004**
GR-Convnet	91.8	70.9	0.007
FCG-Net	86.3	79.3	0.011
SE-ResUNet	85.5	**82.3**	0.017
TF-Grasp	**93.6**	51.3	0.013

not located on AQP in columns 3 and 5-8 of Fig 5.5, as well as columns 1, 2, and 5 of Fig 5.6, most highest scores are concentrated on AQP, further demonstrating the effectiveness of AQP in manipulating the quality score.

4.5.3 Generalizability of AQP

In this part, we also adopt the same experimental setting for the AQP and grasping model as discussed in Section 4.5.1 1). The results are presented in

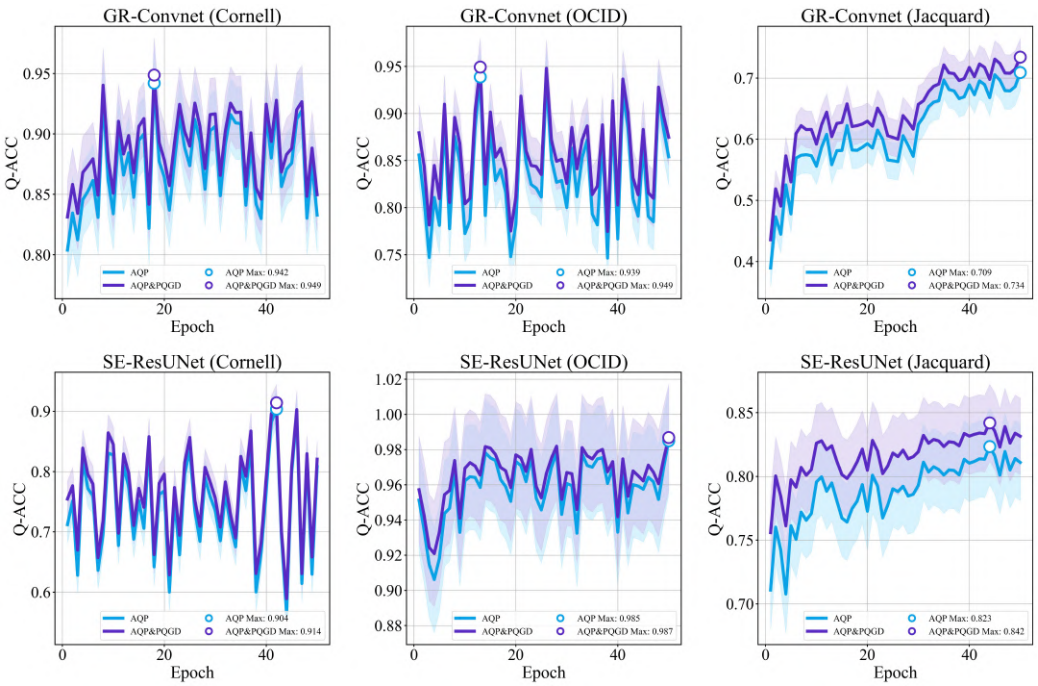

Figure 4.3: Line graphs showing the effectiveness of PQGD across all epochs, including its impact on the AQP optimized by GR-ConvNet and three different datasets, as well as the AQP optimized by SE-ResUNet and three different datasets. Here, the AQP and AQP&PQGD are represented by blue and purple lines, and we also use blue and purple dots to emphasize their corresponding maximum quality score across all epochs.

Table 5.4, where (C $\rightarrow$ O) denotes that the AQP is trained on the Cornell Grasp dataset and tested on the OCID Grasp dataset; other notations follow a similar convention. From this table, although the Q-ACC of the AQP trained on a specific dataset generally decreases when tested on different datasets, most of them still maintain a Q-ACC above 60%. In particular, most of the AQP trained on the OCID Grasp dataset, which contains cluttered scenes, even still achieves high Q-ACC (above 80%) when tested on other datasets. For example, the AQP trained using SE-ResUNet and the OCID dataset even achieves an increased Q-ACC of 98.6% on the Jacquard Grasp dataset. These results demonstrate that the AQP exhibits a certain generalizability across different datasets, with training on cluttered-scene datasets leading to more robust performance.

Table 4.4: Results of AQP generalizability across different datasets

Methods	C → O Q-ACC (%)	C → J Q-ACC (%)	O → C Q-ACC (%)	O → J Q-ACC (%)	J → C Q-ACC (%)	J → O Q-ACC (%)
GG-CNN	78.1 (↓ 21.3)	76.6 (↓ 22.8)	92.8 (↓ 4.1)	86.4 (↓ 10.5)	67.4 (↓ 7.4)	65.2 (↓ 9.6)
GG-CNN2	22.4 (↓ 49.0)	40.9 (↓ 30.5)	82.9 (↓ 7.1)	78.0 (↓ 12.0)	65.5 (↓ 6.0)	51.5 (↓ 20.0)
GR-Convnet	89.3 (↓ 4.9)	89.8 (↓ 4.4)	71.2 (↓ 22.7)	82.3 (↓ 11.6)	55.3 (↓ 15.6)	51.1 (↓ 19.8)
FCG-Net	77.2 (↓ 20.2)	88.3 (↓ 9.1)	84.6 (↓ 6.5)	86.0 (↓ 5.1)	66.0 (↓ 13.3)	59.2 (↓ 20.1)
SE-ResUNet	86.2 (↓ 4.2)	87.5 (↓ 2.9)	98.3 (↓ 0.2)	**98.6** (↑ 0.1)	80.0 (↓ 2.3)	71.6 (↓ 10.7)
TF-Grasp	19.3 (↓ 7.7)	23.3 (↓ 3.7)	88.4 (↓ 5.7)	87.8 (↓ 6.3)	46.0 (↓ 5.3)	29.8 (↓ 21.5)

4.5.4 Effectiveness of PQGD

We validate PQGD by applying it to the AQP optimized in Section 4.5.2 and employing the experimental settings of PQGD discussed in Section 4.5.1 1). In addition, the iteration number N^i is set to 1 in this part. The experimental results are presented in Table 5.5 (for the Cornell Grasp dataset), Table 5.6 (for the OCID Grasp dataset), and Table 5.7 (for the Jacquard Grasp dataset). By comparing these tables with their corresponding Table 5.1, Table 5.2, and Table 5.3, it can be observed that PQGD consistently improves the quality score of the AQP optimized by all models and datasets, with a more pronounced effect on the Jacquard Grasp dataset, resulting in an overall quality score improvement of approximately 2%. Although the prediction speed (all running on one NVIDIA RTX 4090 GPU) decreases with adding PQGD, it remains real-time performance. This reduction has no impact on the efficiency of robot grasping, as the movement time of the robot is significantly longer than the prediction time of the grasping model in practice. Therefore, we enable AQP to rapidly acquire the human hand shape adaptability at a low cost. Additionally, we show the effectiveness of PQGD across all epochs in Fig 5.3, including its impact on the AQP optimized by GR-ConvNet and three different datasets, as well as the AQP optimized by SE-ResUNet and three different datasets. As illustrated in this figure, it is evident that PQGD remains effective throughout all epochs. Since we applied only a random scale to AQP without additional augmentations, the quality score exhibits fluctuations on the smaller Cornell Grasp and OCID Grasp datasets due to overfitting. However, this issue is eliminated for the larger Jacquard Grasp dataset. Overall, this fluctuation does not impact the

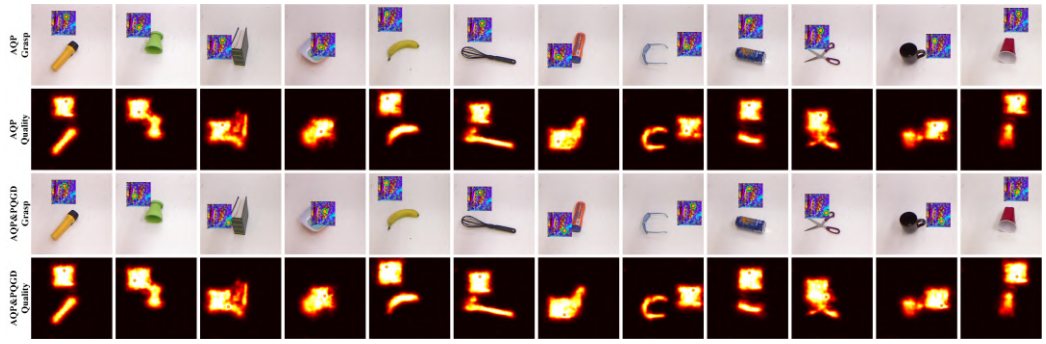

Figure 4.4: Quality score visualization of AQP (first two rows) before and after adding PQGD (last two rows). Here, the GGCNN2 and the Cornell Grasp dataset are used to optimize the AQP. And AQP is scaled to 0.3 of the original size (the same size as the image).

subsequent deployment of our QFAAP, as our objective is not to attack the model but to ensure the achievement of a high quality score. We also visualize the quality performance of AQP after adding PQGD across these datasets in the last two rows of Fig 5.4, 5.5, and 5.6. As shown in these figures, all of the mean quality scores within the AQP can be further improved after adding PQGD. In addition, all cases where the highest quality scores were originally outside the AQP (e.g., columns 3–8 in Fig 5.5 and columns 1, 2, and 5 in Fig 5.6) are corrected after adding PQGD, with the highest quality scores shifting into the AQP; this demonstrates that PQGD can further enhance the highest quality scores within the AQP to some extent. Overall, the PQGD proves effective across different datasets, laying a foundation for subsequent grasping experiments to improve HRI safety by suppressing low-quality scores through high-quality scores with adaptability.

4.5.5 Impact of Iteration Number on PQGD

This part primarily investigates the impact of the iteration number N^i on PQGD. We conduct experiments using the AQP optimized by GR-ConvNet on the Cornell Grasp dataset and the OCID Grasp dataset, with the iteration number N^i ranging from 1 to 10. Other experimental settings remain the same as in Section 4.5.1 1). The results are presented in Table 4.8, which

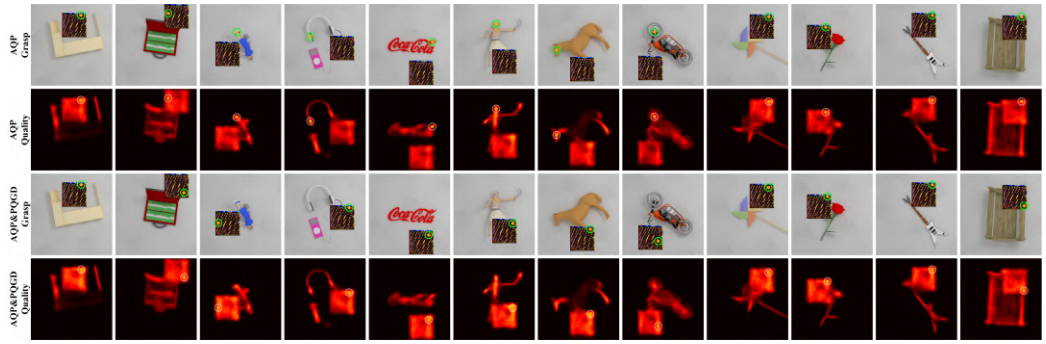

Figure 4.5: The meaning of each row is consistent with Fig 5.5. Here, the SE-ResUNet and the Jacquard Grasp dataset are used to optimize the AQP. And AQP is scaled to 0.3 of the original size (the same size as the image).

Figure 4.6: The meaning of each row is consistent with Figs. 5.5 and 5.6. Here, the GR-ConvNet and the OCID Grasp dataset are used to optimize the AQP. And AQP is scaled to 0.3 of the original size (the same size as the image).

Table 4.5: Results of AQP&PQGD on the Cornell grasp dataset

Methods	O-ACC (%)	Q-ACC (%)	Runtime (s)
GG-CNN	87.6	**99.5**	**0.011**
GG-CNN2	92.1	72.3	0.016
GR-Convnet	96.6	94.9	0.031
FCG-Net	96.6	97.6	0.042
SE-ResUNet	95.5	91.4	0.056
TF-Grasp	**96.8**	31.3	0.038

shows that the optimal number of iterations for PQGD is around 7 for the Cornell Grasp dataset and around 9 for the OCID Grasp dataset. Overall,

Table 4.6: Results of AQP&PQGD on the OCID grasp dataset

Methods	O-ACC (%)	Q-ACC (%)	Runtime (s)
GG-CNN	18.6	97.6	**0.012**
GG-CNN2	44.6	93.0	0.017
GR-Convnet	**53.7**	94.9	0.031
FCG-Net	52.5	92.4	0.044
SE-ResUNet	46.3	**98.7**	0.058
TF-Grasp	26.0	94.7	0.033

Table 4.7: Results of AQP&PQGD on the Jacquard grasp dataset

Methods	O-ACC (%)	Q-ACC (%)	Runtime (s)
GG-CNN	83.7	76.0	**0.017**
GG-CNN2	86.0	74.6	0.023
GR-Convnet	91.8	73.4	0.037
FCG-Net	86.3	82.2	0.052
SE-ResUNet	85.5	**84.2**	0.069
TF-Grasp	**93.6**	57.1	0.069

Table 4.8: The impact of different iteration numbers of PQGD on Q-ACC

Iteration Number N^i	1	2	3	4	5	6	7	8	9	10
Cornell Q-ACC (%)	94.9	96.2	97.5	94.6	96.1	97.0	**99.4**	96.4	95.9	97.1
OCID Q-ACC (%)	94.9	92.8	93.4	95.4	94.8	96.4	96.5	93.4	**98.0**	97.5

different numbers of iterations consistently lead to an improvement in Q-ACC. Additionally, we visualize the effect of the number of iterations N^i on PQGD across all epochs in Fig 4.7. In the upper part of the figure (Cornell Grasp dataset), it can be observed that when the iteration number is 7, the high quality scores (darker purple blocks) are more densely distributed across all epochs compared to other iteration numbers, indicating greater stability. Similarly, in the lower part of the figure (OCID Grasp dataset), the high-quality scores are most densely concentrated when the iteration number is 9. Therefore, the observation from this figure aligns well with the statements discussed in Table 4.8.

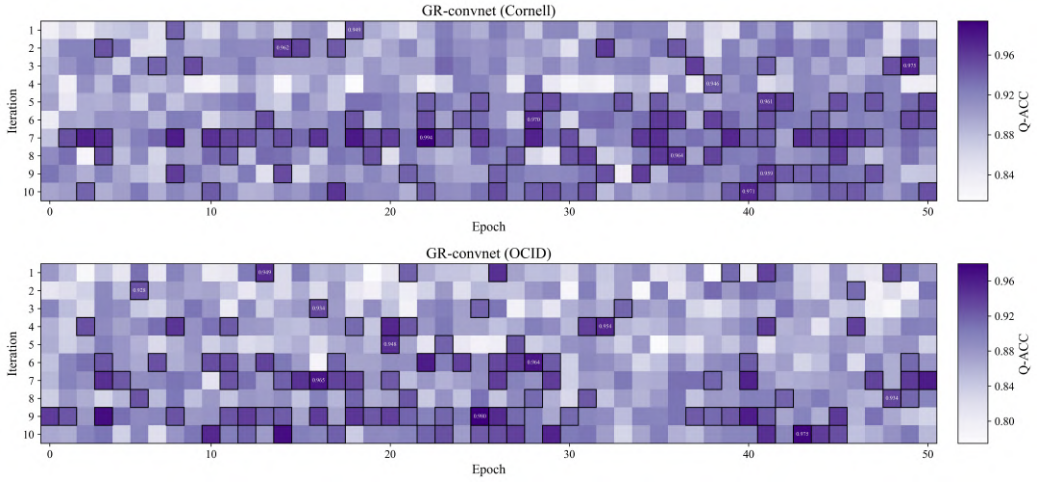

Figure 4.7: Heatmap showing the impact of the iteration number N^i on PQGD across all epochs. Here, the AQP is optimized by GR-ConvNet on the Cornell Grasp dataset (upper sub-figure) and the OCID Grasp dataset (lower sub-figure). In addition, the maximum quality score for each row is printed in white numbers for emphasis.

4.5.6 Effectiveness of QFAAP in Real World

1) Detection Comparison with Original and Engineering Methods: Here, we compare the detection performance of QFAAP with original and engineering methods in single-object scenarios. First, we select 20 objects from the experimental objects and group them into ten pairs. To assess these methods, the hand approaches an object with the highest quality score within each object pair ten times, where the object positions and human hand postures are randomly adjusted in each trial. The comparison methods are divided into two groups. The first group is original methods, including Original (the original grasping model) and Original-SZ (a variant of the grasping model where the quality score of the hand region is set to zero). The second group is engineering methods, including Original-DSZ (enhanced version of Original-SZ with the zeroed area dilation) and Original-Decay (enhanced version of Original-DSZ with the distance-based linear decay). For Original-DSZ, the dilation size is set to 10 pixels since a larger size will reduce the workspace of the robot. For Original-Decay, the dilation size is

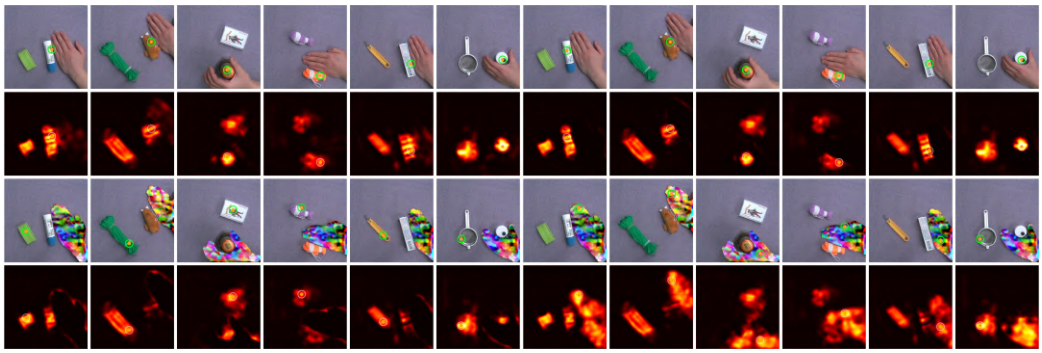

Figure 4.8: Visualization of optimal grasp and quality map for Original (first two rows of the first to sixth columns), Original-SZ (first two rows of the seventh to twelfth columns), QFAAP (last two rows of the first to sixth columns), and QFAAP-NSZ (last two rows of the seventh to twelfth columns).

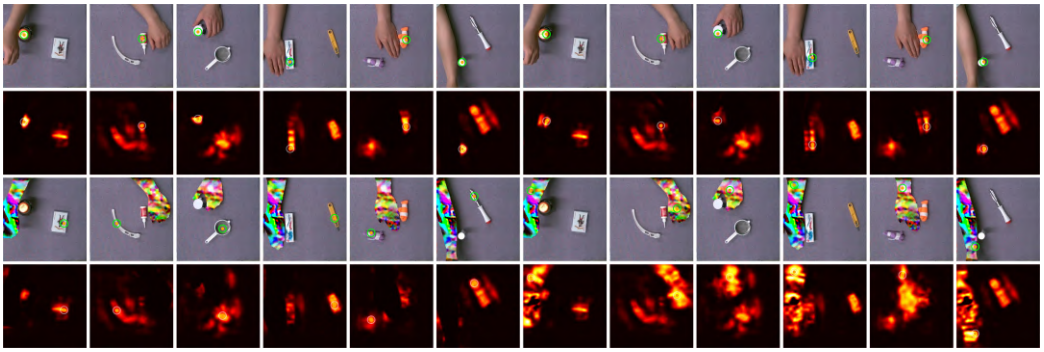

Figure 4.9: Visualization of optimal grasp and quality map for Original-DSZ (first two rows of the first to sixth columns), Original-Decay (first two rows of the seventh to twelfth columns), QFAAP (last two rows of the first to sixth columns), and QFAAP-NSZ (last two rows of the seventh to twelfth columns).

set to 15 pixels, and a distance-based linear decay factor ranging from 0 to 0.8 is applied to the quality score of the region between the boundary of the original area and the boundary of the dilated area, that is the closer to the boundary of the original area, the lower the quality score will be. For our method QFAAP, we use the AQP optimized by GR-ConvNet and OCID Grasp dataset and set the iteration number N^i to 5 for PQGD. All other experimental settings about QFAAP are consistent with Section 4.5.1 2).

The results between our methods and original methods are presented in

Table 4.9: Detection results between QFAAP and original methods

Object Pairs	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Overall (%)	Runtime (s)
Original ND-ACC	1/10	0/10	2/10	0/10	1/10	1/10	1/10	1/10	3/10	3/10	13	0.0069
Original-SZ ND-ACC	1/10	0/10	3/10	0/10	1/10	2/10	1/10	1/10	3/10	3/10	15	0.0087
QFAAP ND-ACC	7/10	9/10	9/10	10/10	8/10	9/10	10/10	8/10	8/10	10/10	**88**	0.0759

Table 4.10: The effectiveness for suppress trigger generated by SEMBA

Object Pairs	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Overall (%)
Original ND-ACC	1/10	2/10	1/10	2/10	1/10	0/10	0/10	1/10	2/10	2/10	12
Original-SZ ND-ACC	1/10	2/10	1/10	2/10	2/10	0/10	0/10	1/10	2/10	2/10	13
QFAAP ND-ACC	10/10	9/10	9/10	9/10	9/10	8/10	9/10	9/10	8/10	10/10	**90**

Table 4.11: Detection results between QFAAP and engineering methods

Object Pairs	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Overall (%)
Original-DSZ ND-ACC	3/10	4/10	5/10	4/10	4/10	5/10	3/10	4/10	4/10	4/10	40
Original-Decay ND-ACC	5/10	6/10	8/10	6/10	4/10	6/10	6/10	6/10	4/10	7/10	58
QFAAP ND-ACC	9/10	10/10	10/10	9/10	8/10	8/10	9/10	9/10	8/10	9/10	**89**

Table 4.9. Our method significantly outperforms both Original and Original-SZ, over 70% ND-ACC, which means that it can noticeably enhance the safety performance of the grasping model in single-object scenarios. For the runtime (all running on one NVIDIA RTX 3090 Ti GPU), although QFAPP is lower than other methods due to the incorporation of the hand segmentation algorithm, it still gets 0.0759 s per frame, which satisfies the real-time requirement in real-world grasping. We also compare these methods in their effectiveness in suppressing triggers generated by SEMBA attacks. The setting is the same as before. The only difference is that we select 10 objects from experimental objects, and each of them will be paired with the same trigger (shown as blue and generated by the OCID dataset). The results are shown in Table 4.10. Our method is also effective for suppressing the quality score of the trigger, thereby improving the grasping safety, and the performance also surpasses the Original and Original-SZ methods.

Then, we show the results between our methods and engineering methods in Table 4.11. Our method still surpass them by a large margin, over 30% ND-ACC. This demonstrates the superiority of the shape adaptability of QFAAP compared with Original-Decay, and the better performance of QFAAP in enhancing safety without influencing the workspace of the robot compared

with Original-DSZ. We also visualize some of our results in Fig 4.8 and Fig 4.9, including the optimal grasp and quality map for Original, Original-SZ, Original-DSZ, Original-Decay, QFAAP, and QFAAP-NSZ (a variant of the QFAAP where the quality score of the hand region is not set to zero). As shown in these figures, compared with other methods, our method can always shift the highest quality score to the object away from the human hand by decreasing the quality score of the object near the human hand, no matter the different scenarios and hand poses. It should be noted that QFAAP-NSZ is only to emphasize the strength of the quality score for QFAAP and is not included in the experimental tables. Finally, the few failure cases of QFAAP primarily result from situations where the object approached by the human hand still maintains a higher quality score than the other object. In future work, we will enhance our optimization methods to strengthen QFAAP.

2) The Impact of Distance on the Effectiveness of QFAAP: We conduct extensive distance-based quantitative experiments to explore the quality suppression behavior of QFAPP, using an experimental setup similar to that in Section 4.5.6 1). Specifically, one object among the 40 experimental objects is selected as the non-target object, while the remaining 39 objects are treated as target objects to be approached by the human hand, forming 39 object pairs in total. For each of the 39 pairs, we perform five trials of hand-approaching experiments. In each trial, the position of the object pair and the posture of the hand are first randomly changed. Then, the hand gradually approaches the target object with the same posture until contact is made. During this approaching process, we record the changes in the highest grasp quality score at distances of 2 cm, 1 cm, 0.5 cm, and upon contact. The results are shown in Table 4.12, where noticeable suppression begins to occur at a distance of 1 cm, when the ND-ACC reaches 47.6%. Subsequently, the ND-ACC increases to 81.0% at 0.5 cm and further to 93.3% upon contact with the target object. These results strongly demonstrate that our distance-based quantitative analysis aligns well with the property in [97], namely, that the quality score suppression of QFAPP becomes effective when the hand is within 0.5–1 cm of the target object and reaches its maximum effect at contact.

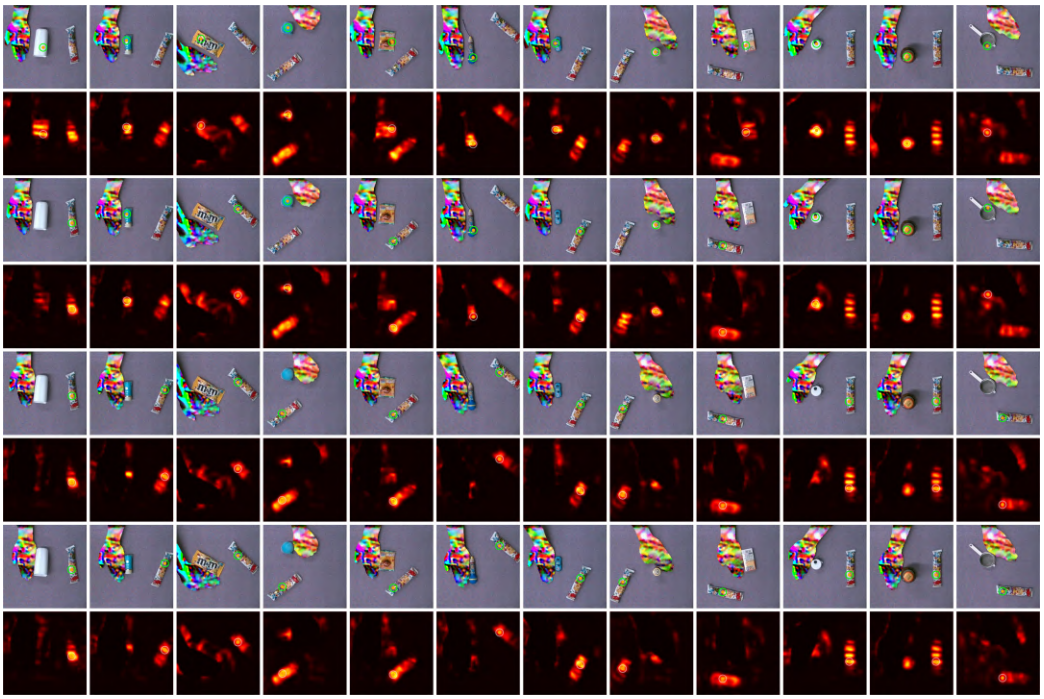

Figure 4.10: Visualization of optimal grasp and quality map for QFAPP with distance 2 *cm* (first two rows), distance 1 *cm* (third and fourth rows), distance 0.5 *cm* (fifth and sixth rows), and distance 0 *cm* (last two rows).

We further visualize the changes in quality scores and optimal grasps for different distances in Fig 4.10. When the distance is 2 *cm* (first two rows: optimal grasps in the first row, corresponding quality scores in the second row), almost no suppression effect is observed. At a distance of 1 *cm* (third and fourth rows), some suppression occurs, though failures are still observed in columns 2, 4, 6, 8, 10, 11, and 12. When the distance is reduced to 0.5 *cm* (fifth and sixth rows), or contact is made (last two rows), the highest quality score is consistently shifted away from the object near the human hand. Notably, in the first and ninth columns of the last two rows (contact case), the quality scores of adjacent target objects are nearly suppressed to zero.

3) Grasping Comparison with the Version of QFAAP without PQGD: In this part, we evaluate the influence of PQGD on QFAPP in a real robot grasping system. We follow the same experimental setting in

Table 4.12: Distance-based detection resutls for QFAAP

Objects	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14	B15	B16	B17	B18	B19	B20
QFAAP (2.0 *cm*) ND-ACC	0/5	2/5	1/5	1/5	2/5	0/5	1/5	2/5	1/5	0/5	3/5	1/5	0/5	0/5	1/5	0/5	0/5	0/5	0/5	0/5
QFAAP (1.0 *cm*) ND-ACC	1/5	3/5	2/5	3/5	4/5	0/5	2/5	4/5	3/5	0/5	3/5	0/5	0/5	1/5	3/5	2/5	0/5	4/5	0/5	2/5
QFAAP (0.5 *cm*) ND-ACC	3/5	3/5	3/5	5/5	4/5	2/5	4/5	5/5	4/5	3/5	4/5	5/5	3/5	1/5	5/5	5/5	3/5	4/5	3/5	5/5
QFAAP (0.0 *cm*) ND-ACC	5/5	5/5	4/5	5/5	5/5	5/5	5/5	5/5	4/5	5/5	5/5	5/5	5/5	4/5	5/5	5/5	3/5	4/5	5/5	5/5

Objects	B21	B22	P23	P24	P25	P26	P27	P28	P29	P30	B31	B32	B33	B34	B35	B36	B37	B38	B39	Overall (%)
QFAAP (2.0 *cm*) ND-ACC	0/5	4/5	0/5	2/5	0/5	2/5	0/5	0/5	3/5	2/5	0/5	0/5	3/5	2/5	0/5	0/5	1/5	1/5	0/5	17.4
QFAAP (1.0 *cm*) ND-ACC	3/5	4/5	3/5	3/5	3/5	5/5	2/5	4/5	5/5	3/5	3/5	3/5	5/5	3/5	1/5	3/5	1/5	2/5	0/5	47.6
QFAAP (0.5 *cm*) ND-ACC	5/5	5/5	4/5	5/5	5/5	5/5	3/5	5/5	5/5	5/5	5/5	4/5	5/5	5/5	2/5	5/5	3/5	4/5	4/5	81.0
QFAAP (0.0 *cm*) ND-ACC	5/5	5/5	5/5	5/5	5/5	5/5	3/5	5/5	5/5	5/5	5/5	4/5	5/5	5/5	4/5	5/5	3/5	5/5	4/5	**93.3**

Table 4.13: The impact of PQGD on QFAAP in real grasping

Object Pairs	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Overall (%)
QFAAP-without PQGD CH-Rate	2/10	3/10	2/10	1/10	2/10	3/10	1/10	2/10	1/10	3/10	20
QFAAP CH-Rate	1/10	2/10	2/10	0/10	1/10	1/10	0/10	2/10	1/10	2/10	**12**

Section 4.5.6 1) and Section 4.5.1 2). Specifically, we perform 10 grasps for each object pair where the object positions and human hand postures are randomly adjusted in each grasp. The experimental results are presented in Table 4.13. The CH-Rate of QFAAP without PQGD reaches 20%. After integrating PQGD, the CH-Rate decreases to 12%, demonstrating that PQGD can also effectively enhance the fast adaptability to the human hand shape in real-world grasping scenarios. Finally, we showcase the grasping performance with and without PQGD in our demo videos.

4) Grasping Comparison with Original and Engineering Methods: We use a similar experimental setting as in Section 4.5.6 1) and Section 4.5.1 2) for this part. Specifically, we first select 10 objects from the experimental objects to create 10 mid-clutter grasping scenes and perform 10 grasps (the hand pose will be changed in each grasping) for each scene to compare QFAAP with the original method. Then, similarly, we select 30 objects from the experimental objects to create 5 high-clutter grasping scenes and perform 30 grasps with multi-hand interference for each scene to compare QFAAP with the engineering method.

The experimental results between QFAAP and original methods are shown in Table 4.14, where our method consistently outperforms both the Original and Original-SZ methods, achieving a notably low CH-Rate of 16%. This result demonstrates the effectiveness of QFAAP in enhancing the safety of the HRI in mid-clutter grasping scenarios. We also visualize some grasping

Table 4.14: Grasping results between QFAAP and original methods

Scenarios	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	Overall (%)
Original CH-Rate	8/10	6/10	6/10	7/10	8/10	4/10	5/10	6/10	7/10	5/10	62
Original-SZ CH-Rate	6/10	6/10	6/10	6/10	7/10	4/10	5/10	6/10	7/10	5/10	58
QFAAP CH-Rate	2/10	1/10	2/10	3/10	2/10	0/10	2/10	1/10	2/10	1/10	**16**

Table 4.15: Grasping results between QFAAP and engineering methods

Scenarios	S1	S2	S3	S4	S5	Overall (%)
Original-DSZ CH-Rate	18/30	15/30	13/30	16/30	14/30	50.7
Original-Decay CH-Rate	12/30	11/30	10/30	11/30	11/30	36.7
QFAAP CH-Rate	4/30	7/30	3/30	3/30	3/30	**13.3**

results of QFAAP in Fig 4.11. Compared with the original grasping model, our method can effectively shift the robot to grasp the object away from the human hand.

The experimental results between QFAAP and engineering methods are shown in Table 4.15, where our method also outperforms both the Original-DSZ and Original-Decay methods, achieving a promising CH-Rate of 13.3%, which is more than 20% lower than them. This result demonstrates the superiority of QFAAP in enhancing the safety of the HRI in high-clutter grasping scenarios with multi-hand interference. We also visualize some grasping results of QFAAP in Fig 4.12. The first row shows normal grasping without hand interference. The second row shows the result without our method under multiple hand interferences, where the robot easily collides with the human hands. The third row presents the result using our method under the same multi-hand interference scenario, where the robot successfully avoids all hands and nearby objects during grasping. Finally, the reasons for the failure cases of QFAAP in these scenarios remain consistent with those in Section 4.5.6 1). Since this is real grasping experiment, we also find that the robot will collide with the human if our method is failure, and we will press the emergency stop button to stop the robot to make sure the safety of the human.

5) HRI User Study: In this part, we conduct the HRI user study to evaluate the safety of our proposed method from the users' perspective. To

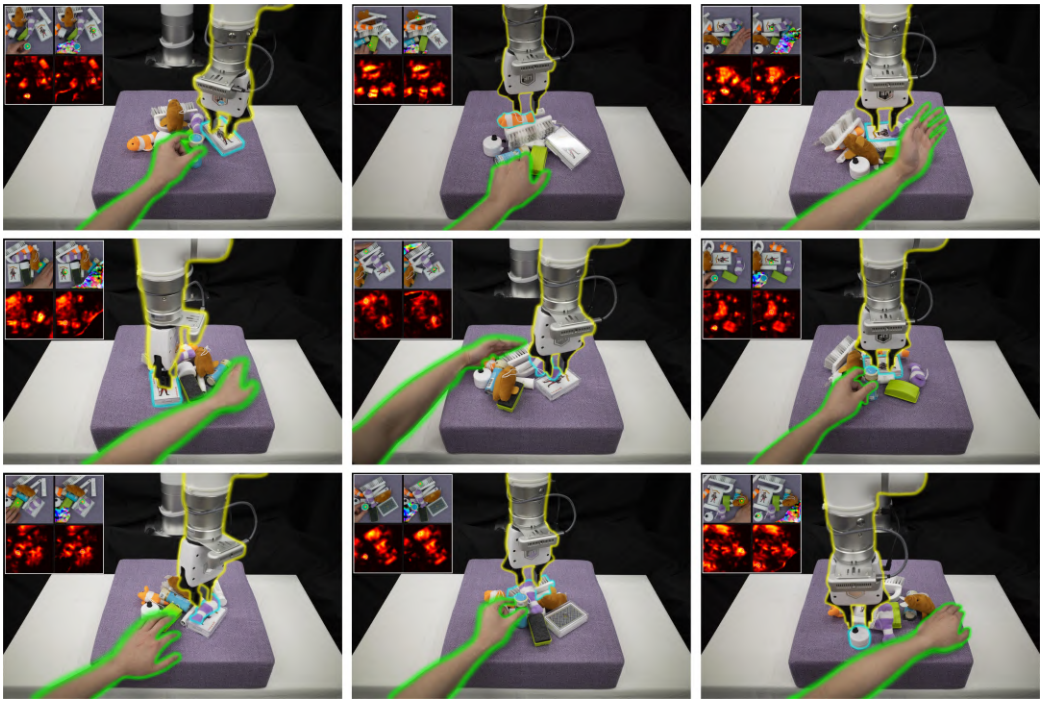

Figure 4.11: Grasping in mid-clutter scenarios. We use yellow, green, and blue borders to highlight the robot, the human hand, and the objects being grasped in each subfigure. In addition, we added the optimal grasp and quality map for QFAAP (left) and the original grasping model (right) to each subfigure.

minimize safety risks during the experiments while ensuring the depth of this study, we strictly limit the number of participants to five, all of whom are researchers with professional knowledge in robotics. And we have obtained their approvals. We define the following user-centered evaluation metrics: ADCS (the adaptability of the method to different clutter scenarios), ADHP (the adaptability of the method to different hand poses), UPS (the user perceived safety), RRF (the robot response fluency), and UOS (the user overall satisfaction). Each metric is rated on a five-point integer scale (1 to 5). During the experiment, users compare the performance of QFAAP with the Original method under a similar setting as described in Section 4.5.6 4). Specifically, each user performs five interactions with each method in a high-clutter scene containing 30 objects, testing whether the robot can

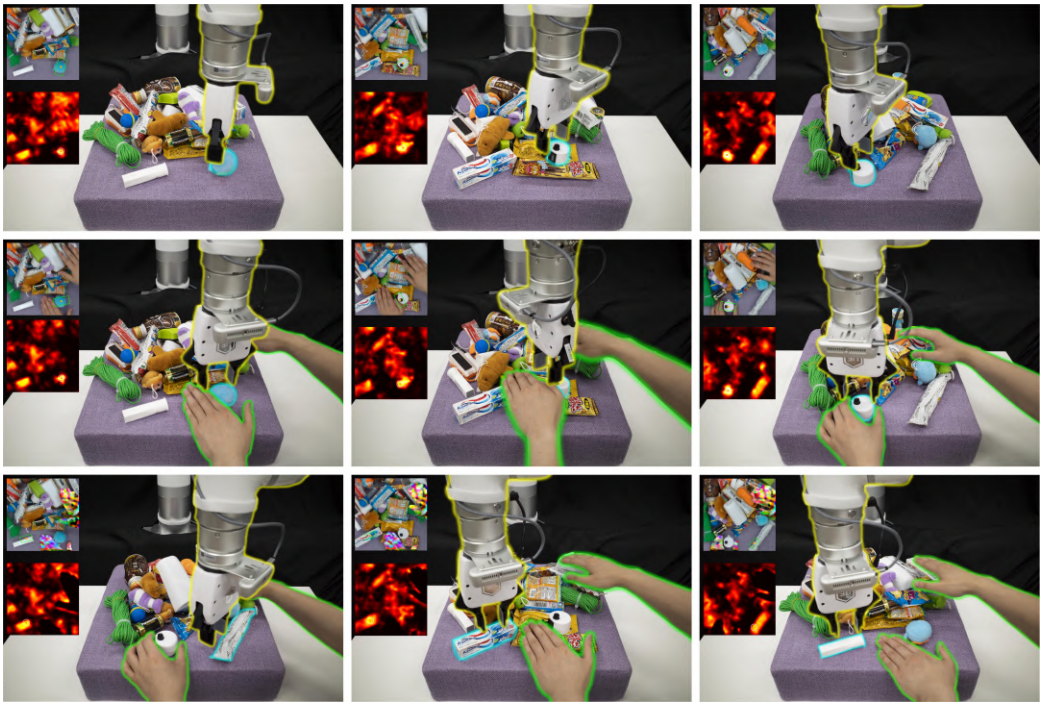

Figure 4.12: Grasping in high-clutter scenarios with bimanual interference. The first row shows normal grasping without hand interference. The second and third rows show the grasping without and with our method under bimanual interference. We use yellow, green, and blue borders to highlight the robot, the human hands, and the objects being grasped in each subfigure. In addition, we added the optimal grasp and quality map to each subfigure.

grasp objects while avoiding human hands and their neighboring objects. In addition, users are allowed to choose single-hand or multi-hand configurations freely for each interaction, and the hand poses and object positions are varied across all trials.

The user feedback results are summarized in Table 4.16. As shown in the table, users consistently rated QFAAP significantly higher than the Original method across nearly all evaluation metrics. For example, (4 vs. 1.8) in the average of ADCS, (4.8 vs. 1) in the average of ADHP, (4.8 vs. 1.6) in the average of UPS, and (4.4 vs. 1.8) in the average of UOS. Although users reported a slightly lower RRF for QFAAP compared to the Original method (a difference of 0.4), they perceived the difference as minor and confirmed that QFAAP is capable of conducting HRI in real-time. Finally, after completing

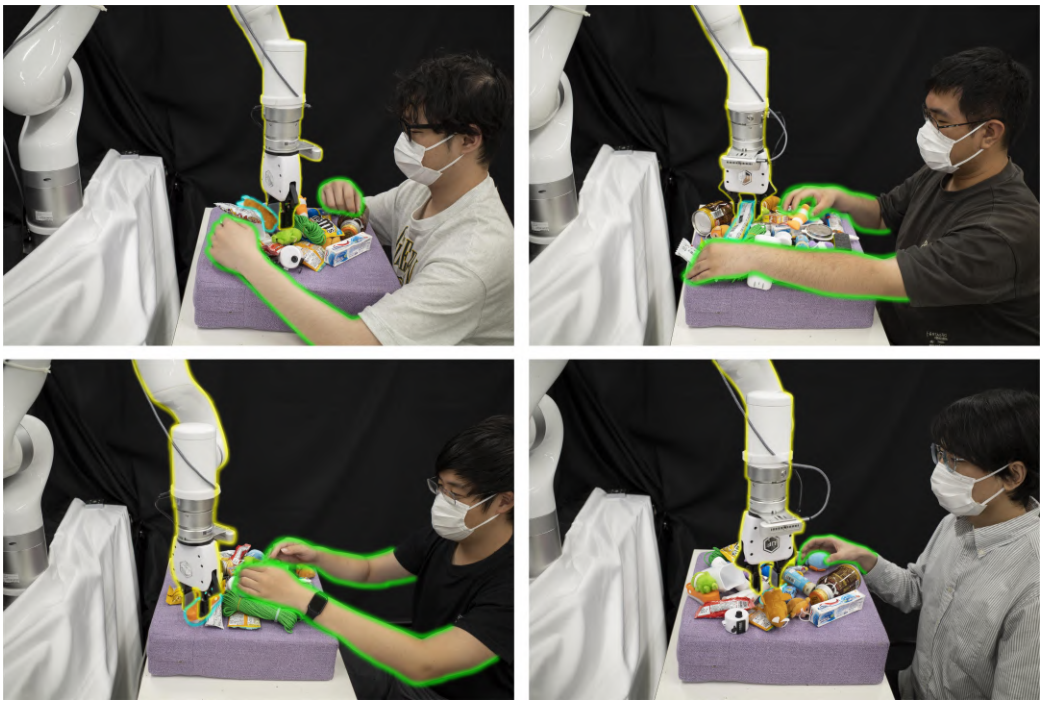

Figure 4.13: Examples of HRI user study. We use yellow, green, and blue borders to highlight the robot, the human hand, and the objects being grasped.

Table 4.16: Results of HRI User study

Participants	ADCS		ADHP		UPS		RRF		UOS	
	Original	QFAAP	Original	QFAAP	Original	QFAAP	Original	QFAAP	Original	QFAAP
Participant 1	2	4 (↑ 2)	1	5 (↑ 4)	2	5 (↑ 3)	4	4 (–)	2	4 (↑ 2)
Participant 1	1	4 (↑ 3)	1	4 (↑ 3)	2	5 (↑ 3)	5	4 (↓ 1)	2	4 (↑ 2)
Participant 3	2	4 (↑ 2)	1	5 (↑ 4)	3	5 (↑ 2)	4	4 (–)	2	5 (↑ 3)
Participant 4	1	4 (↑ 3)	1	5 (↑ 4)	0	4 (↑ 4)	4	4 (–)	1	4 (↑ 3)
Participant 5	3	4 (↑ 1)	1	5 (↑ 4)	1	5 (↑ 4)	4	3 (↓ 1)	2	5 (↑ 3)
Average	1.8	**4** (↑ 2.2)	1	**4.8** (↑ 3.8)	1.6	**4.8** (↑ 3.2)	**4.2**	3.8 (↓ 0.4)	1.8	**4.4** (↑ 2.6)

the experiments, all participants expressed that QFAAP demonstrated strong safety and adaptability, and were willing to deploy this method. We further illustrate several examples of the HRI process in Fig 4.13, where it can be seen that regardless of the number of hands, hand poses, or scene variations, the robot consistently performs grasps while avoiding human hands and their nearby objects. More HRI processes are recorded in the demo videos.

6) Grasping with and without Hand Interference: Since QFAAP employs the same grasping model as the original method, it only modifies

Table 4.17: Grasping results with and without hand Interference

Scenarios	S1	S2	S3	S4	S5	GS-Rate (%)
without hand	24/30	24/30	24/30	25/30	25/30	**81.3**
with hand	23/30	24/30	24/30	24/30	25/30	80.0

the output grasp quality scores when hand interference is present, thereby altering the grasping sequence. In the absence of such interference, it remains consistent with the original method. Therefore, in this section, we directly compare grasping performance with and without human-hand interference to verify whether our method affects the original performance of the grasping model. Specifically, we use a similar experimental setting as in Section 4.5.6 4), selecting 30 objects from the experimental objects to create 5 high-clutter grasping scenes and perform 30 grasps with or without multi-hand interference for each scene. In addition, we use the same Grasping Success Rate (GS-Rate) as an evaluation metric from [99], which is calculated by dividing the total number of successful grasps by the total number of grasp attempts across five scenes.

The grasping results with and without hand interference are presented in Table 4.17. The GS-Rate is 81.3% without hand interference and 80.0% with hand interference, and the total number of successful grasps with hand interference is only two shy of that without interference. These results indicate that the grasping performance remains nearly identical in both cases, suggesting that our method has almost no impact on the original performance of the grasping model while ensuring grasping safety.

7) Grasping with Hand Dynamic Interference: To more comprehensively validate the effectiveness of QFAAP, particularly its real-time reactive capability, we conduct additional grasping experiments under hand dynamic interference in this section. We adopt a similar experimental setting as in Section 4.5.6 4), also selecting 30 objects from the experimental set to create 5 high-clutter grasping scenes and performing 30 grasp attempts for each scene, while randomly introducing unimanual or bimanual dynamic interference during each grasping. We reproduce the closed-loop control

Table 4.18: Grasping results with hand dynamic Interference

Scenarios	S1	S2	S3	S4	S5	DRD-Rate (%)
QFAAP-without PQGD	22/30	22/30	23/30	23/30	22/30	74.7
QFAAP	24/30	24/30	25/30	27/30	26/30	**84.0**

method from [3] and integrate it with QFAAP, endowing QFAAP with the reactive capability to counteract hand dynamic interference. Specifically, during each grasping, when the robot tends to move toward the target object, we quickly introduce hand interference by approaching the target object. After the interference, the robot will move away from the human hand and its adjacent objects (First Deviation). Once the extent of deviation becomes large, we quickly remove the hand, and the robot will resume moving toward the target object (Return). Similarly, after the extent of the move toward the target object becomes large, we again rapidly introduce hand interference and keep the hand in place until the robot deviates away from the hand and its neighboring objects (Second Deviation) and completes the safe grasping. We employ the Deviation–Return–Deviation Rate (DRD-Rate) as the evaluation metric, and a trial is considered successful if the robot completes the entire Deviation–Return–Deviation process.

The experimental results are shown in Table 4.18, the DRD-Rate of QFAAP without PQGD reaches 74.7%. After integrating PQGD, the DRD-Rate dramatically increases to 84.0%, demonstrating the effectiveness of our method in hand dynamic interference scenarios, and also validates that PQGD can more obviously enhance the fast adaptability to the human hand shape in hand dynamic interference scenarios compared with hand static interference in Section 4.5.6 3). Finally, we show two cases of the DRD process in Fig 4.14, where it can be seen that the robot consistently avoids the human hand and its nearby objects with hand dynamic interference. More DRD processes are recorded in the demo videos.

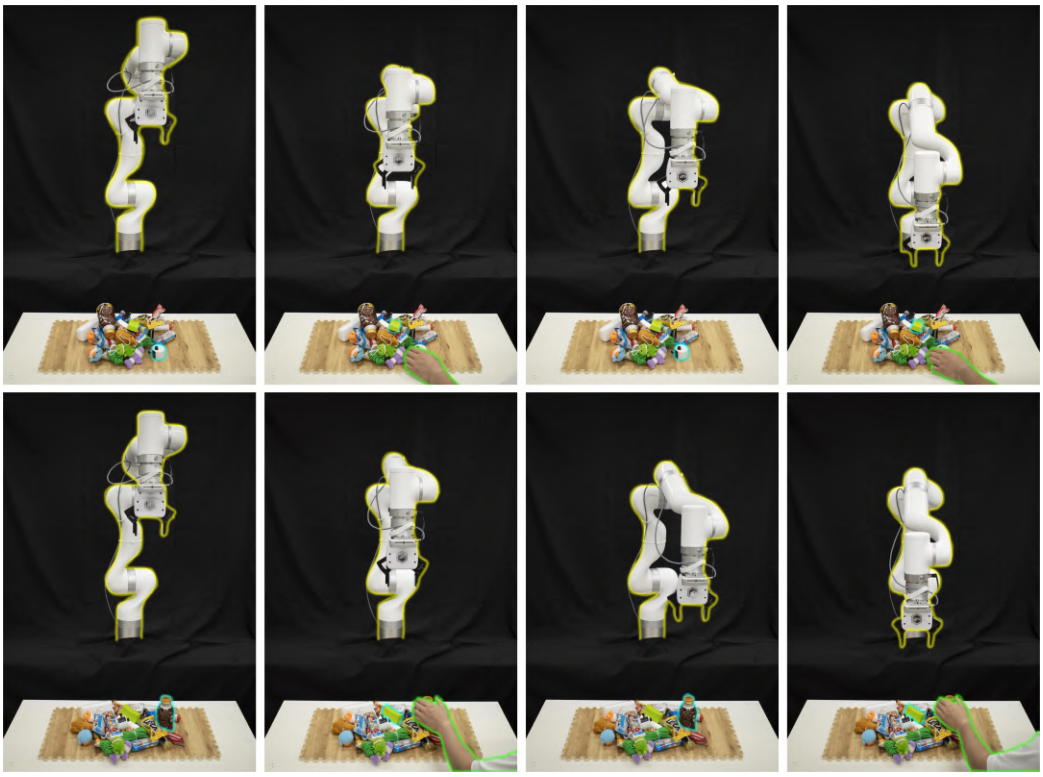

Figure 4.14: Cases of the DRD process under hand dynamic interference. Each row corresponds to one case, and the images in each row respectively illustrate the initial approach of the robot to the target object, the first deviation of the robot after interference, the re-approach (return) of the robot to the target object after the hand departs, and the second deviation of the robot after the second interference. Yellow, green, and blue borders are also used to highlight the robot, the human hand, and the target objects, respectively.

Chapter 5

Monozone-Centric Instance Grasping Policy

5.1 Overview of MCIGP

We propose a novel grasping policy, the Monozone-centric Instance Grasping Policy (MCIGP), designed to improve grasping safety and realize grasping in large-scale dense clutter, as illustrated in Fig 5.1. MCIGP is composed of two main modules: Monozone View Alignment (MVA) and Instance-specific Grasp Detection (ISGD). The MVA is used to break the camera's field of view boundaries and is divided into two types: Quality-based MVA (Q-MVA) and Depth-based MVA (D-MVA). The ISGD predicts and optimizes grasp candidates for one specific object within the monozone to make sure an in-depth analysis of it, which includes Cross-prompted Segmentation (CPS) and Grasp Candidate Optimization (GCO).

5.2 Monozone View Alignment (MVA)

Since grasping models typically accept inputs of size 224×224, we configure the dynamic monozone according to this size. It is important to note that we refer to it as a dynamic monozone because it will change after each view alignment (except the size), which distinguishes it from the 224×224 center region within the fixed view.

Given that the resolution of the depth camera (640×480) is usually larger than 224×224, a coarse global view alignment is required at each grasping

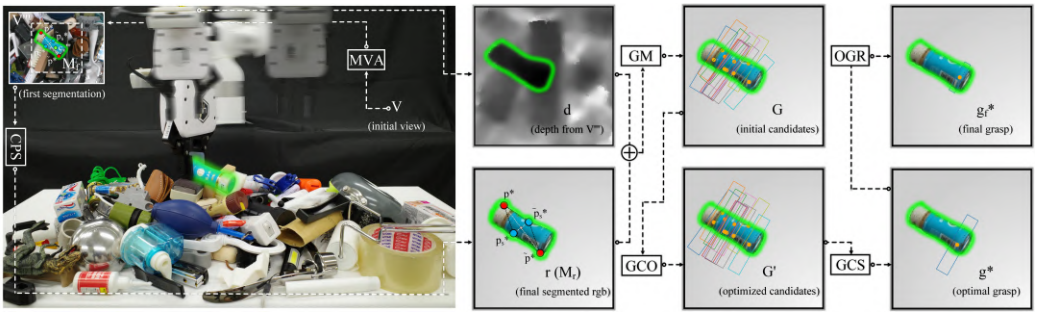

Figure 5.1: Pipeline of MCIGP: Firstly, conducting Monozone View Alignment (MVA) to align the initial view $\mathcal{V}$ of depth camera on the target object to get view $\mathcal{V}'''$, and segment this object by the center c_v''' (green point) of this view as prompt to obtain initial segmented RGB image (emphasized with green borders) with mask $\mathcal{M}_f$. Then, calculate two pairs of most distant points (p^* (red point), $\tilde{p}^*$ (red point), p_s^* (blue point), and $\tilde{p}_s^*$ (blue point)) based on the edge of $\mathcal{M}_f$, and using these points to make Cross-prompted Segmentation (CPS) to optimize $\mathcal{M}_f$ to get $\mathcal{M}_r$. In step three, the segmented RGB image $\mathtt{r}$ with mask $\mathcal{M}_r$ and the depth image $\mathtt{d}$ within view $\mathcal{V}'''$ are fed into the Grasping Model (GM) to generate initial grasp candidates $\mathbb{G}$, followed by Grasp Candidate Optimization (GCO) to obtain optimized grasp candidates $\mathbb{G}'$. After GCO, $\mathbb{G}'$ will be processed through Grasp Candidate Sampling (GCS) to find the optimal grasp g^*. Finally, g^* is optimized by Optimal Grasp Refinement (OGR) to transfer it to the final grasp g_f^*. Notably, the left part of the figure with the robot is focused on MVA, while the right part of the figure (6 subfigures) is focused on Instance-specific Grasp Detection (ISGD).

to find the dynamic monozone that contains the target object. Specifically, based on the initial depth image $\mathcal{V}$ and the center point c_v of the camera view, we align the pixel corresponding to the minimum depth value (among all 640×480 pixels) in $\mathcal{V}$ with c_v by moving the robot (the camera mounted on the robot will be also moved). We denote the depth image after this camera movement by $\mathcal{V}'$. Assuming that the point with the minimum depth value is located at p_{c_d} in the camera coordinate system, it can be transformed to p_{r_d} via a hand-eye relationship (without translation). By moving the robot to p_{r_d}, the original point p_{c_d} can be brought to the center of the camera view, thereby achieving view alignment, as shown in Eq. 5.1, where $\mathbf{0}_{3\times1}$ denotes a 3×1 zero matrix. Notably, this process places the primary focus more

on narrowing down the region of interest containing the target object with the minimum depth value, that is finding the dynamic monozone. Therefore, it is significantly different from directly selecting and grasping the object corresponding to the minimum depth within a fixed view as in [31], which will also suffer from the problem of view boundary limitation.

$$\begin{bmatrix} p_{r_d} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_c^r & \mathbf{0}_{3\times1} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix} \begin{bmatrix} p_{c_d} \\ 1 \end{bmatrix} \tag{5.1}$$

After finding the dynamic monozone, we perform Monozone View Alignment, which includes two types: Quality-based MVA (Q-MVA) and Depth-based MVA (D-MVA). Due to the large search range during global alignment, the uncertainty of depth values also increases significantly. As a result, the position aligned to the globally minimal depth value may not correspond to the actual minimal depth in the scene. The first type D-MVA can refine the global alignment by continuing to align within the dynamic monozone (we perform two alignments here), allowing the identification of the object corresponding to the locally minimal depth value. Following the global alignment stage, let $\mathcal{V}''$ and $\mathcal{V}'''$ denote the depth images after each D-MVA step, respectively. In addition, the robot moves to follow Eq. 5.1. Notably, unlike the previous global view alignment, the robot motion during this alignment is limited to the 224×224 monozone. As a result, the object centered in the new camera view after D-MVA will become the final target for grasping.

Different from D-MVA, Q-MVA can predict the grasp quality score through the grasping model within the dynamic monozone and select the pixel corresponding to the highest score as the alignment point. Let $\mathcal{Q}$ denote the quality map of this monozone, and (x_q, y_q) the corresponding position of one quality score. Then the corresponding position (x_q^*, y_q^*) of the highest score can be shown in Eq. 5.2, where (H, W) is the size of the monozone.

$$(x_q^*, y_q^*) = \arg\max_{(x_q, y_q)\in(H,W)} \mathcal{Q}(x_q, y_q) \tag{5.2}$$

Now, suppose (x_q^*, y_q^*) is located at $p_{c_q}^*$ in the camera coordinate system. This position can be transformed into the robot coordinate system as $p_{r_q}^*$ via a hand-eye transformation (excluding translation). By moving the robot to $p_{r_q}^*$, the original position $p_{c_q}^*$ is brought to the center of the camera view (one alignment), thereby achieving Q-MVA, following Eq. 5.1. The robot movement ranges the same as in D-MVA and is restricted within the 224×224 monozone.

5.3 Instance-Specific Grasp Detection (ISGD)

In this part, we perform Instance-specific Grasp Detection within the aligned monozone. We first leverage the center point of the aligned monozone as the initial prompt and apply Cross-prompted Segmentation (CPS) to segment the target object located at the center. The segmented result is then fed into the grasping model to predict grasp candidates. These candidates are further refined through Grasp Candidate Optimization (GCO). Finally, we sample the optimized candidates and refine the best one to generate the final grasp.

1) Cross-prompted Segmentation (CPS): We initially leverage the center point of the aligned monozone as the initial prompt to drive the Segment Anything Model (SAM) [100] to segment the target object located at the center. We denote the segmented result by mask $\mathcal{M}_f$. However, the single-point prompt is highly unstable in dense clutter, particularly pronounced when the object's appearance is complex, such as the food packaging, where only part of the object is segmented (usually manifested as many holes in the segmented object). This limitation adversely impacts the subsequent prediction and optimization of grasp candidates.

Therefore, we propose Cross-prompted Segmentation (CPS), which performs a geometric analysis of the initial segmentation result $\mathcal{M}_f$ and conducts a second segmentation to alleviate the segmentation hole effect. Specifically, it begins by applying the Sobel operator [101] to extract the edges of the instance mask $\mathcal{M}_f$ obtained from the initial single-point prompt segmentation. We then search for the two pixels most distant from each other, which

we refer to as p^* and $\tilde{p}^*$ on the edges. As shown in Eq. 5.3, $\mathcal{P}_e$ means the set of all pixels on the edges, and $\|\cdot\|_2$ means the Euclidean distance.

$$(p^*, \tilde{p}^*) = \underset{(p_i, p_j) \in \mathcal{P}_e}{\arg\max} \|p_i - p_j\|_2 \tag{5.3}$$

Next, we calculate the perpendicular line $\perp_{(p^*, \tilde{p}^*)}$ connecting p^* and $\tilde{p}^*$, and intersecting this perpendicular line with the edges $\mathcal{P}_e$ yields another pair of the most distant pixels, which we refer to as p_s^* and $\tilde{p}_s^*$ as shown in Eq. 5.4. These four points are then used as prompts to perform the second segmentation, resulting in $\mathcal{M}_s$. Compared to a single point, these two farthest pairs of points can better exploit the geometric constraints of the initial segmentation result, thus alleviating the holes in the initial segmentation, as demonstrated in our ablation studies. Finally, $\mathcal{M}_s$ is refined by image dilation processing: a depth threshold is applied and pixels from the first prompt serve as initial points for segmentation to produce $\mathcal{M}_d$. By combining $\mathcal{M}_d$ and $\mathcal{M}_s$, we obtain the refined $\mathcal{M}_r$.

$$(p_s^*, \tilde{p}_s^*) = \perp_{(p^*, \tilde{p}^*)} \cap \mathcal{P}_e \tag{5.4}$$

2) Grasp Candidate Optimization (GCO): After segmenting the target object, we preserve pixels of the image within the mask $\mathcal{M}_r$, while all others will be set to 255, and input this revised image to the grasping model. Here, we use the grasping model in [46] to obtain the grasp candidates. This will allow the grasping model to focus solely on the target object, and the predicted grasp candidates will also concentrate exclusively on this object, which is deemed to be instance-specific grasp detection. Then we propose Grasp Candidate Optimization (GCO) to optimize the predicted grasp candidates. Inappropriate selection of the grasp angle θ may cause the object to slip or fall during grasping due to the uneven force distribution on both fingers of the parallel-jaw gripper. Therefore, the first part of GCO is to optimize all grasp candidates (denoted by $\mathbb{G}$) to have an optimal angle. Specifically, it begins with extracting the edges of the instance mask $\mathcal{M}_r$.

For each grasp candidate, we rotate them clockwise in 2-degree intervals until they reach 360 degrees. For each rotation $\mathcal{R}$, we find four intersection

points between the two long sides of the grasp candidate and the edges, that is, p_{t_l}, p_{b_l}, p_{t_r}, and p_{b_r}. Subsequently, we calculate the angle θ' between the vector $\mathbf{v}_{p_l}$ determined by p_{b_l} and p_{t_l} and the vector $\mathbf{v}_{g_u}$ of the long upper side of this grasp candidate, and similarly to get the angle θ'' between the vector $\mathbf{v}_{p_r}$ determined by p_{b_r} and p_{t_r} and the vector $\mathbf{v}_{g_u}$. By subtracting 90 degrees from each of these angles, taking the absolute value, and summing them, we obtain the angle difference for each rotation. Finally, we select the rotation $\mathcal{R}^*$ with the smallest angle difference given by Eq. 5.5 and use it to formulate the new grasp candidate.

$$\begin{aligned} \mathcal{R}^* = \arg\min_{\mathcal{R}} \left(\left| \theta'(\mathcal{R}) - \frac{\pi}{2} \right| + \left| \theta''(\mathcal{R}) - \frac{\pi}{2} \right| \right) \\ \text{s.t. } \mathcal{R} \in \{0^\circ, 2^\circ, 4^\circ, \ldots, 2\pi\} \end{aligned} \tag{5.5}$$

The second part of GCO is designed to ensure that viable grasp candidates remain available after sampling. It achieves this by adaptively rotating the image's viewpoint clockwise, thereby altering all grasp candidates. This part can work synergistically with the first part of GCO for joint optimization. Specifically, if no grasp candidate g_i is available from the current viewpoint, we rotate the image 30 degrees at a time and repeat it until an available grasp candidate is found.

Since the camera does not rotate and is constrained by hand-eye calibration, we need to project the rotated candidate grasp g_i' back to the original viewpoint. Here, the parameters w' and h' of g_i' remain unchanged. The angle θ' can be adjusted by adding the rotation angle θ_c and restricting it to the range $[-\frac{\pi}{2}, \frac{\pi}{2}]$ to convert it back to the angle θ of the candidate grasp g_i under the original viewpoint. For the center $c_i'(x', y')$ of g_i', assuming the center of rotation is $c_r(x_{c_r}, y_{c_r})$, the projection relationship from $c_i'(x', y')$ to the center $c_i(x, y)$ of the grasp candidate g_i under the original viewpoint can be obtained from Eq. 5.6.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos(\theta_c) & -\sin(\theta_c) \\ \sin(\theta_c) & \cos(\theta_c) \end{bmatrix} \begin{bmatrix} x' - x_{c_r} \\ y' - y_{c_r} \end{bmatrix} + \begin{bmatrix} x_{c_r} \\ y_{c_r} \end{bmatrix} \tag{5.6}$$

Based on GCO, we start sampling the grasp candidates within this

object, which is a heuristic-based method by a large number of experimental observations made. In addition, it only analyzes and samples candidates of a single object and without the guidance of the mask. Therefore, it is different from instance-level grasping, which analyzes all objects and uses the mask of each object to guide sampling. Let $\mathbb{G}'$ denote the grasp candidates after angle calibration (the second part of GCO is also dynamically activated). We first analyze their relationship with adjacent objects by setting a depth threshold $\mathcal{T}_d$, that is, if the depth difference between any p and c_i exceeds $\mathcal{T}_d$, the grasp candidate g_i will be filtered out and get grasp candidate sets $\mathbb{G}''$, as shown in Eq. 5.7. Here $\mathcal{P}_s$ means pixels along the two short sides of the grasp candidate g_i, and p is one pixel within $\mathcal{P}_s$, $\mathtt{d}$ means depth image.

$$\mathbb{G}'' = \{g_i \in \mathbb{G}' \mid \forall p \in \mathcal{P}_s, \, |\mathtt{d}(p) - \mathtt{d}(c_i)| \leq \mathcal{T}_d\} \tag{5.7}$$

After getting $\mathbb{G}''$, we use our previous method [102] to sort $\mathbb{G}''$ with depth value and select the g_i with the smallest center pixel depth value $\mathtt{d}(c_i)$ as the optimal grasp g^*, which is shown in Eq. 5.8.

$$g^* = \arg\min_{g_i \in \mathbb{G}''} \mathtt{d}(c_i) \tag{5.8}$$

Finally, although the optimal grasp g^* was obtained, it might still result in collisions with adjacent objects during grasping execution due to its too wide open width. One way to get around this problem was reported in [103], where a series of intervals was defined within the grasp box and the grasp width and position were adjusted based on the relationships between these intervals. However, this method relies on the intersection depth area of objects can easily cause errors, and is computationally cumbersome. Therefore, we directly find the minimum rectangle $\mathcal{R}_{ec}$ intersecting the optimal grasp g^* and the instance mask $\mathcal{M}_r$ in the RGB image. Followed by calculating the shortest width w_s and a new center point c_i' of by $\mathcal{R}_{ec}$. Additionally, to mitigate the impact of hand-eye calibration errors, we further expand w_s to w_s' by adding some of the hand-eye calibration translation errors e_c in the X and Y-axis. So, w_s' and c_i' can be used as the new width and center of the grasp for optimal grasp and it can be denoted as the final grasp g_f^*.

Figure 5.2: Objects for the grasping experiment: toys, ragdolls, household goods, and snacks (clockwise from top left).

5.4 Experiments

In this section, we validate the effectiveness of MCIGP by conducting benchmarking studies. Firstly, we compare it with baseline grasping methods in various mid-clutter (up to 20 objects) and high-clutter (up to 50 objects) scenes. Then we increase the number of objects to 100 (large-scale clutter) and analyze the effectiveness of MVA and ISGD.

5.4.1 Experimental Settings

1) Setting for Grasping Model: The baseline methods are categorized into two groups. The first group includes GGCNN [45], GGCNN2 [3], GRconvnet [46], SEnet [8], and FCGnet [47], which are suitable for mid-clutter scenarios. The second group comprises DexNet 4.0 [51] and GraspNet [56], which are tailored for high-clutter scenarios.

For the first group, since the pre-trained models were all trained on the Cornell Grasping Dataset [44] (only one object in each fixed white

background), their performance in cluttered environments is limited. Therefore, we merge the OCID Grasping Dataset [34, 93] (with different piled objects, backgrounds, sensor-to-scene distance, viewpoint angle, and lighting conditions) into the Cornell Grasping Dataset and retrain these models using the parameter settings specified in their original papers (except that all uses the RGB-D modality). Specifically, these models were trained on a single NVIDIA RTX 4090 GPU with 24 GB of memory. The system is Ubuntu 22.04, and the deep learning framework is PyTorch 2.3.1 with CUDA 12.1. Before training, we randomly shuffle the entire dataset, using 90% for training and 10% for testing. During training, the data are uniformly cropped to fit the acceptable sizes, the number of training epochs is set to 50, and data augmentation (random zoom and random rotation) is applied. For testing, we use the same metric [92] to report the detection accuracy (Acc) of these methods. According to this metric, a grasp is considered valid when it satisfies two conditions: the Intersection over Union (IoU) score between the ground truth and predicted grasp rectangles is over 25%, and the offset between the orientation of the ground truth rectangle and that of the predicted grasp rectangle is less than 30 degree.

For the second group, we directly use their pre-trained models: the parallel-jaw version of DexNet 4.0, the planar version of GraspNet (GraspNet 4D) [56, 104], and the 6-DOF version of GraspNet (GraspNet 6D) [56]. Finally, unless specified, the segmentation and grasp candidate prediction components of MCIGP use the pre-trained models of SAM and GRconvnet in all experiments, and we use D-MVA for all experiments too.

2) Setting for Real Grasping: Our grasping system consists primarily of an Intel RealSense D435 depth camera, a UFactory xArm 5 robot (5-DOF), and a UFactory 850 robot (6-DOF). We employ an eye-in-hand architecture, with the camera mounted on the robot's distal end and facing downward. There are real object types of benchmarking in the dense clutter grasping field, such as [105]. However, most of these objects are European and American products, which are usually difficult to obtain completely due to regional restrictions. Moreover, the types and number of such benchmarks are scarce, which cannot meet our needs for large-scale dense cluttered (up to

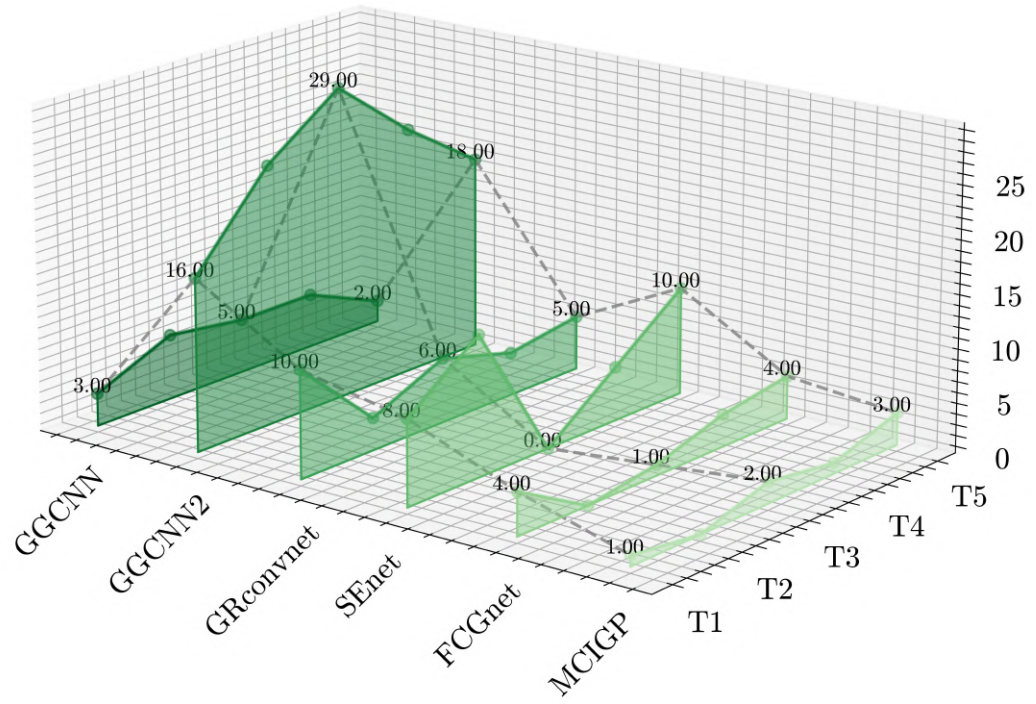

Figure 5.3: Line graph showing GSR of MCIGP and first-group baselines in mid-clutter scenarios. The horizontal axis represents different methods, and the depth axis represents trials from T1 to T5. The vertical axis represents the number of grasp failures. We emphasize the number of grasp failures (T1, T3, T5) in each method with dots, and connect them with dashed lines to better show the difference.

100 objects) grasping experiments. Therefore, we refer to the objects used in two widely recognized dense clutter grasping methods, DexNet 4.0 [51] and GraspNet [58]. Specifically, the objects used in our grasping experiments are divided into four categories, with a total of 300 novel objects: 50 ragdolls (Category 1), 100 snacks (Category 2), 50 toys (Category 3), and 100 household goods (Category 4), respectively, as shown in Fig 5.2. Category 1 is the easiest, and as the category number goes up, the grasping difficulty for the robot rises, too.

Prior to grasping, we first define the robot's workspace in the base coordinate system with the X and Y axes limited by the edges of a 120 cm $\times$ 80 cm table. The range of the Z-axis is limited by the maximum distance to the tabletop (40 cm) and the minimum distance (10 cm) to

prevent collisions between the gripper fingers and the table. The camera is mounted at the distal end of the robot arm to which the gripper (about 10 *cm*) is also removably mounted. Before each grasping attempt, we set the robot to a pre-specified position (40 *cm* above the center of the table) and ensure that the camera covers the entire pile of objects on the table. Then we fill the depth hole [45] and set a depth value threshold (with the upper limit of 40 *cm* and the lower limit of 10 *cm*) to ensure that the grasp is executed within a safety range.

During grasping, each method is tested in five trials per experiment, and the number of failed grasps in each trial (T) is recorded. The grasp success rate (GSR) is calculated by dividing the total number of successful grasps by the total number of grasp attempts across five trials. In addition, to improve experimental safety and ensure all objects are grasped in each trial, we provide minimal manual assistance during the experiments. Specifically, if an object fails to be grasped 2-3 times, we manually pick up the object and count it as a failure. Additionally, if an object moves out of the camera view, it is repositioned with manual intervention. Similarly, if a grasped object moves out of the robot's workspace, causing it to stop, the object is repositioned manually, too.

5.4.2 Comparison Studies

1) Comparison with Baseline Methods in Mid-clutter: We compare MCIGP with the baseline methods in the first group. We used 10 snacks and 10 household goods to form a mid-clutter scene. The results are shown in Table 5.1. MCIGP achieves a GSR of 93.5% (100/107), with only 7 grasp failures, which is far superior to other baselines except for FCGnet. Additionally, we found that some baselines perform well on the dataset but not in real grasping. For example, GGCNN2 has a GSR of only 47.6% (100/210) with a total of 110 grasp failures, indicating that this method does not generalize well to novel objects in mid-clutter. Finally, we also visualize the result in Fig 5.3 to better show the gap between MCIGP with other baseline methods. As shown in this figure, it is very clear that our

Table 5.1: GSR comparison among MCIGP and first-group baselines in mid-clutter scenarios

Methods	T1	T2	T3	T4	T5	Acc (%)	GSR (%)
GGCNN	3	6	5	5	2	22.3	82.6
GGCNN2	16	24	29	23	18	37.7	47.6
GRconvnet	10	3	6	4	5	52.0	78.1
SEnet	8	13	0	5	10	45.0	73.5
FCGnet	4	0	1	3	4	52.0	89.3
MCIGP	1	0	2	1	3	-	**93.5**

Table 5.2: GSR comparison among MCIGP and second-group baselines in high-clutter scenarios

Methods	Ragdolls						Snacks						Toys						Household goods					
	T1	T2	T3	T4	T5	GSR (%)	T1	T2	T3	T4	T5	GSR (%)	T1	T2	T3	T4	T5	GSR (%)	T1	T2	T3	T4	T5	GSR (%)
DexNet 4.0	4	1	1	3	2	95.8	10	15	7	10	12	82.2	29	23	24	28	30	65.1	29	29	28	26	26	64.4
GraspNet 4D	5	6	3	2	6	92.0	13	10	6	18	14	80.4	25	27	28	21	21	67.2	17	38	30	35	36	61.6
MCIGP	0	1	1	2	0	**98.4**	4	4	3	3	1	**94.3**	6	10	10	7	5	**86.8**	8	6	10	11	5	**86.2**

method's number of failures is much smaller with little variance across the trials.

2) Comparison with Baseline Methods in High-clutter: DexNet 4.0 and GraspNet are considered state-of-the-art for learning-based 4-DOF and 6-DOF grasping, respectively. Therefore, to demonstrate the effectiveness of MCIGP's grasping capability, we compare it with these two methods. Specifically, we first compare with the parallel gripper version of DexNet 4.0 and the planar version of GraspNet. Here, we conducted experiments in high-clutter scenes composed of 50 ragdolls, 50 snacks, 50 toys, and 50 household goods, respectively. The experimental results are shown in Table 5.2, indicating that MCIGP achieves GSR of 98.4% (250/254) for ragdolls, 94.3% (250/265) for snacks, 86.8% (250/288) for toys, and 86.2% (250/290) for household goods. All surpassed DexNet 4.0 and GraspNet 4D. More importantly, as the difficulty in grasping increases, the gap between MCIGP and the baseline methods becomes more obvious. For example, when grasping toys and household goods, MCIGP's GSR exceeds theirs by up to 20%, demonstrating the high reliability of our method. We also visualize these results in Fig 5.4. It is obvious that the bar length of our method is

Table 5.3: GSR comparison between MCIGP and GraspNet 6D in high-clutter scenarios

Methods	T1	T2	T3	T4	T5	GSR (%)
GraspNet 6D	12	13	12	11	10	81.2
MCIGP	8	6	10	11	5	**86.2**

shorter than that of other methods in each subfigure, and it varies little across the trials and has smaller errors. To further demonstrate the superiority of our method, we compare it against the extremely challenging GraspNet 6D. The experimental settings are consistent with those described above, except that we only conduct experiments on the most difficult high-clutter scenes to better reflect their performance differences, composed of 50 household goods. As shown in Table 5.3, despite MCIGP supporting only 4-DOF grasping (GSR is 86.2% (250/290)), it still outperforms GraspNet 6D (GSR is 81.2% (250/308)).

5.4.3 Ablation Studies

1) Effectiveness of Monozone View Alignment: To demonstrate the effectiveness of MVA, we first evaluate it in a non-clutter scenario consisting of 10 household objects. In these scenarios, some parts of the objects' geometries may not be fully captured by the depth camera, simulating potential view boundary limitations encountered by baseline methods. We compare our method with the first group baseline methods, and the experimental results are presented in Table 5.4. MCIGP achieves a GSR of 90.9% (50/55) with only five grasp failures, significantly outperforming the other baselines. This demonstrates that our method can substantially improve grasp success rates and reduce the safety problem of object ejection at high speeds by overcoming boundary limitations. In addition, Table 5.4 reports the average time from grasp detection to grasp execution for all methods. Although our method is approximately twice as slow as the baselines, the execution time remains within an acceptable range. The additional time required by our method is reasonable, as it conducts more visual analysis compared to the

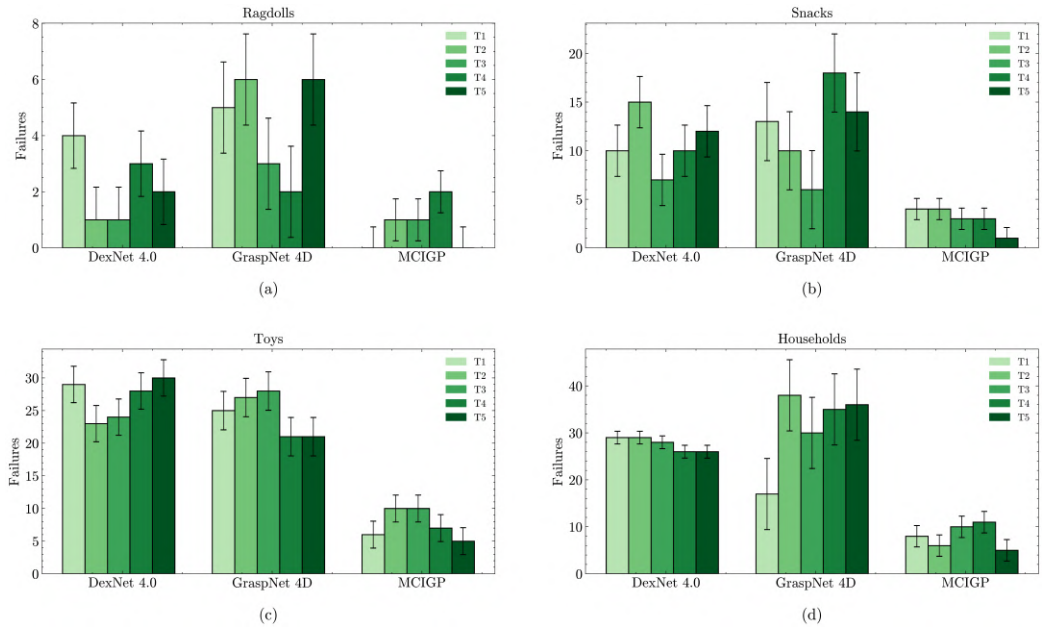

Figure 5.4: Bar graphs showing GSR of MCIGP and second-group baselines in high-clutter scenarios. (a), (b), (c), and (d) represent the results of testing ragdolls, snacks, toys, and household goods. In each subfigure, the vertical axis represents the number of grasp failures, and the horizontal axis represents different methods with five trials. We show the positive and negative errors at the top of each bar by calculating the mean of the number of grasp failures across all trials for each method.

baseline methods in order to achieve higher grasp success rates. We also visualize the view alignment and grasping process in Fig 5.6. Note that the visualization here is mainly based on Section 5.4.3 2).

Next, we investigate the difference between D-MVA and Q-MVA. Here, we use a mid-clutter scenario consisting of 20 household objects. The experimental results are shown in Table 5.5. Q-MVA achieves a GSR of only 74.6% (100/134), whereas D-MVA achieves 90% (100/111), exhibiting a 15.4% performance gap. This result indicates that D-MVA outperforms Q-MVA in mid-clutter scenarios.

2) Effectiveness of Instance-specific Grasp Detection: In this section, we first validate the CPS component in ISGD using a large-scale clutter scenario composed of 100 snack objects, whose complex appearances can

Figure 5.5: Visualization of CSP and SP segmentation. The first and second rows are the CSP segmentation and CSP grasp, respectively. The third and fourth rows are the SP segmentation and SP grasp, respectively. In addition, we use translucent magenta and green rectangles to emphasize the mask and grasp.

Table 5.4: GSR comparison among MCIGP and first-group baselines in non-clutter scenarios

Methods	T1	T2	T3	T4	T5	Time (s)	GSR (%)
GGCNN	4	5	7	6	7	23.5	63.3
GGCNN2	9	8	9	9	7	28.0	54.3
GRconvnet	2	5	5	1	4	27.8	74.6
SEnet	5	5	4	4	4	24.3	69.4
FCGnet	2	1	3	3	4	25.5	79.4
MCIGP	0	1	1	1	2	54.5	**90.9**

effectively highlight the advantages of CPS. Here, MCIGP without CPS employs single-point (SP) segmentation, while other aspects remain consistent with the original MCIGP. The experimental results are presented in Table 5.6; the GSR of MCIGP without CPS is 79.1% (500/632), compared to 88.7% (500/564) achieved by the original MCIGP, demonstrating the effectiveness of CPS. Furthermore, Fig 5.5 visualizes the segmentation differences between CPS and SP, where CPS is observed to significantly reduce segmentation holes and small segmented regions compared to SP segmentation, thereby helping the grasping model predict better grasp.

Next, we evaluate the GCO component in ISGD under conditions of

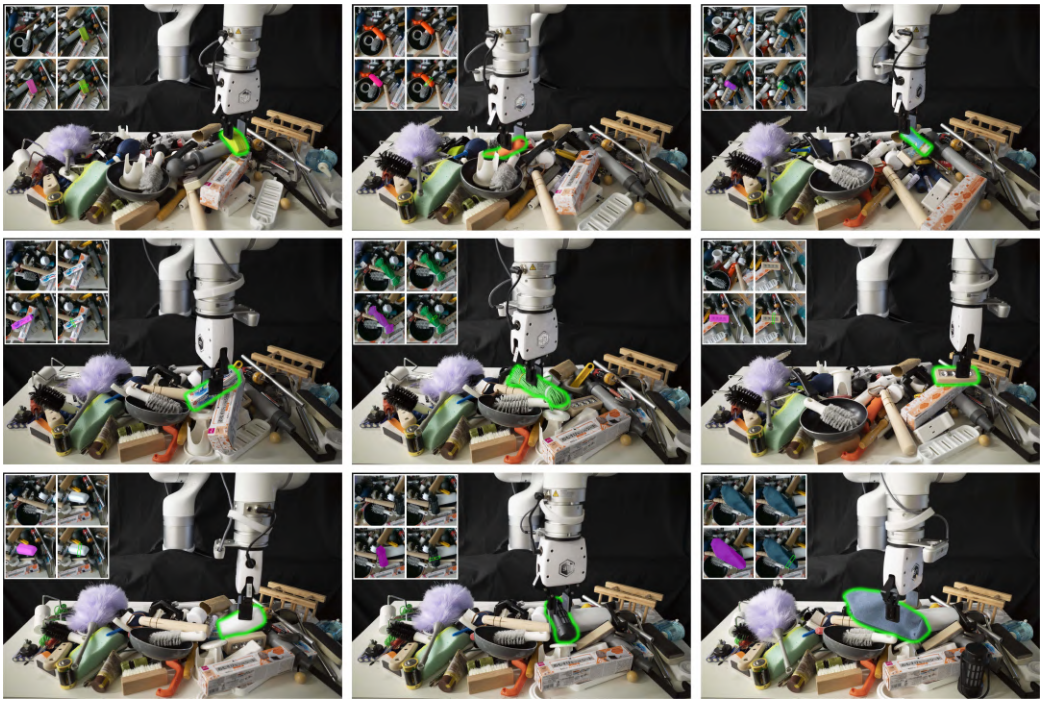

Figure 5.6: Visualization of the grasping process on large-scale clutter scenarios with 100 household goods for MCIGP. Each subfigure represents the grasping process, and we emphasize the object being grasped by the green border. Sub-subfigures inside each subfigure are the original view (top left), aligned view (top right), segmentation based on the aligned view (bottom left), and the predicted grasp based on the aligned view (bottom right), respectively. The mask and grasp are also emphasized by translucent magenta and green rectangles.

Table 5.5: Impact of different MVA in mid-clutter scenarios

Methods	T1	T2	T3	T4	T5	GSR (%)
Q-MVA	6	5	10	6	7	74.6
D-MVA	4	1	1	4	1	**90.0**

Table 5.6: Impact of CPS in large-scale clutter scenarios

Methods	T1	T2	T3	T4	T5	GSR (%)
Without CPS	23	29	20	28	32	79.1
With CPS	19	14	9	9	13	**88.7**

Table 5.7: Impact of GCO in large-scale clutter scenarios

Methods	T1	T2	T3	T4	T5	GSR (%)
Without GCO	25	35	32	34	41	75.0
With GCO	22	14	17	12	24	**84.9**

the highest grasping difficulty, specifically within large-scale cluttered scenes composed of 100 household goods. These objects exhibit the greatest variation in materials, shapes, and appearances compared to other objects that we use. The experimental results shown in Table 5.7, the GSR of MCIGP without GCO is 75% (500/667), compared to 84.9% (500/589) for MCIGP. Two cases differ by approximately 10%, illustrating the obvious advantage of GCO in large-scale dense clutter scenarios. We also visualize some of the grasping processes in Fig 5.6, as well as record in these videos.

Chapter 6

Conclusion

This dissertation presents a comprehensive exploration into the safety of the AI-powered robot visual grasping system in cluttered human-robot interaction scenarios. By integrating the findings from three major studies, this work systematically investigates both external and inherent risks in this system and proposes novel methods to enhance grasping safety through different policy designs.

Firstly, we introduced the Shortcut-Enhanced Multimodal Backdoor Attack (SEMBA), which integrates multimodal information and shortcut learning to uncover previously unexplored vulnerabilities in AI-powered robot visual grasping systems. Unlike conventional backdoor attacks that rely solely on single-modality cues, SEMBA leverages the RGBD information and exploits the model's inherent shortcut deficiency to achieve more robust manipulation for the grasp quality score. To the best of our knowledge, this is the first systematic attempt to design a backdoor attack specifically tailored for AI-powered robot visual grasping. Extensive experiments conducted on large-scale benchmark datasets and real-world robot grasping systems demonstrate SEMBA's strong attack capability, transferability across different models, and high practical feasibility in complex environments.

Secondly, to counteract the external SEMBA threat, we developed the Quality-Focused Active Adversarial Policy (QFAAP), which allows the human hand itself to serve as an active perturbation source to suppress grasp quality score near potential trigger-like regions. By integrating the Adversarial Quality Patch (AQP) and the Projected Quality Gradient Descent (PQGD), QFAAP dynamically adjusts the grasping quality distribution in the visual scene, reducing the grasping priority of hazardous areas while

maintaining operational fluency and task efficiency. Extensive experiments conducted on both benchmark datasets and real-world collaborative robots from single-object, cluttered, and dynamic HRI scenarios (human hands and robots coexist in close proximity) demonstrated their effectiveness.

Finally, to overcome the inherent safety issues caused by limited grasping view and inferior grasp candidates, we proposed the Monozone-Centric Instance Grasping Policy (MCIGP), which integrates Monozone View Analysis (MVA) and Instance-Specific Grasp Detection (ISGD) for precise and reliable grasping within dynamically detected monozones. By adaptively focusing on each monozone, MCIGP enables the robot to refine its perception of target objects and generate high-quality grasp candidates even under large-scale dense clutter with heavy occlusion. This design effectively mitigates the limitations of restricted camera perspectives and unstable grasp candidates that can cause object splashing. Comprehensive experiments involving over 8,000 real-world grasp attempts on 300 novel objects across multiple clutter levels demonstrated that MCIGP not only enhances grasp accuracy and robustness but also achieves superior adaptability and generalization, significantly outperforming seven competitive grasping baselines.

In summary, this dissertation advances the understanding and implementation of secure and intelligent visual grasping. The proposed methods collectively contribute to building a safe and adaptive human-aware robotic grasping framework, paving the way toward trustworthy human-robot interaction in complex, real-world environments.

Limitations and future works: For SEMBA, 1) The attack performance will significantly deteriorate when the trigger undergoes large rotations around the X and Y axes of the camera coordinate frame. This failure arises because the designed trigger is inherently 2D and does not account for the effects of 3D transformations. As a result, it performs effectively only under translations along each axis and rotations around the Z axis in the camera coordinate frame, which is consistent with the characteristics of 4-DOF grasping. To address this issue, we plan to design 3D triggers in future work to enable attacks at arbitrary angles. 2) The depth trigger in the real world exhibits slight distributional drift compared to the ideal

depth trigger in the dataset, as it is susceptible to depth camera noise and interference from adjacent objects in real-world scenarios. Therefore, enhancing the robustness of the depth trigger against such disturbances in the real world will also be a focus of our next plan. 3) Concentrate on extending our method in some industrial scenarios with more different degradation factors and data distribution, to further enhance its ability to protect AI-powered visual grasping systems in different environments. For example, we will attempt to use domain adaptation techniques to generate simulated data with different degradation factors and apply adversarial training strategies to improve the domain adaptability of our method in different scenarios. 4) Other parts can focus on exploring the influence of geometry and size for the backdoor trigger design, and the effectiveness of our method in other models, like transformer and diffusion models.

For QFAAP, 1) Since our current methods are confined to the laboratory setting, future work could focus on extending them to field robotics capable of adapting to diverse open-world environments. For instance, deploying our approach on quadruped or humanoid robots equipped with manipulators would be a promising direction. 2) Moreover, as our current method addresses SEMBA attacks only in the RGB modality, future research should further explore the multimodal potential of QFAAP. 4) Other parts can focus on exploring the combinition of QFAAP and safety-oriented motion planning algorithm, to further enhance the safety when human interact with the robot.

For MCIGP, 1) The slippage during grasp execution, which occurs due to the smooth surface of the object. To address this, we plan to use parallel jaw grippers with high-friction finger pads or wrap the fingers with textured tape. 2) Depth holes and errors from the depth camera can cause a collision with the table during grasping. This problem can be mitigated by using a high-precision industrial depth camera. 3) While our method can improve the segmentation of SAM, it will be unable to segment the complete shape of an object if serious occlusion exists between objects in dense clutter. To optimize this, we are going to use amodal instance segmentation to predict the occluded parts of the object, thereby getting the complete mask. 4) Other parts, like how to extend our method to safely manipulate deformable objects,

and investigate the effect of different depth sensors, are also interesting.

Finally, from the perspective of the entire robot system. Extending our approach toward building a comprehensive and safe human–robot handover system is an exciting and promising future direction. For instance, it is worth investigating how to achieve handover-oriented target object retrieval in dense clutter, and how to effectively perceive human intent during the process of transferring the object to the human hand.

References

[1] M. A. Goodrich and A. C. Schultz, "Human-robot interaction: A survey," *Found. Trends Hum.–Comput. Interact.*, vol. 1, no. 3, pp. 203–275, 2007

[2] S. Yu, D.H. Zhai, and Y. Xia, "SKGNet: Robotic grasp detection with selective kernel convolution," *IEEE Trans. Autom. Sci. Eng.*, vol. 20, no. 4, pp. 2241-2252, 2023.

[3] D. Morrison, P. Corke, and J. Leitner, "Learning robust, real-time, reactive robotic grasping," *Int. J. Robot. Res.*, vol. 39, no. 2-3, pp. 183–201, 2020.

[4] Y. Laili, Z. Chen, L. Ren, X. Wang, and M. J. Deen, "Custom grasping: A region-based robotic grasping detection method in industrial cyber-physical systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 20, no. 1, pp. 88–100, 2023.

[5] D.H. Zhai, S. Yu, and Y. Xia, "FANet: Fast and Accurate Robotic Grasp Detection Based on Keypoints," *IEEE Trans. Autom. Sci. Eng.*, Early Access Article, pp. 1-13, 2023.

[6] A. Depierre, E. Dellandréa, and L. Chen, "Jacquard: A large scale dataset for robotic grasp detection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 3511–3516.

[7] L. Tong, K. Song, H. Tian, Y. Man, Y. Yan, and Q. Meng, "A novel RGB-D cross-background robot grasp detection dataset and background-adaptive grasping network," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–15, 2024.

[8] S. Yu, D.-H. Zhai, Y. Xia, H. Wu, and J. Liao, "SE-ResUNet: A novel robotic grasp detection method," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 5238–5245, 2022.

[9] L. Tong, K. Song, H. Tian, Y. Man, Y. Yan, and Q. Meng, "SG-Grasp: Semantic segmentation guided robotic grasp oriented to weakly textured objects based on visual perception sensors," *IEEE Sensors J.*, vol. 23, no. 22, pp. 28430–28441, 2023.

[10] T. Gu, L. Liu, B. Dolan-Gavitt, and S. Garg, "Badnets: Evaluating backdooring attacks on deep neural networks," *IEEE Access*, vol. 7, pp. 47230-47244, 2019.

[11] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, "Targeted backdoor attacks on deep learning systems using data poisoning," 2017 *arXiv:*1712.05526.

[12] H. Ma, Y. Li, Y. Gao, Z. Zhang, A. Abuadbba, A. Fu, S. F. Al-Sarawi, S. Nepal, and D. Abbott, "TransCAB: Transferable clean-annotation backdoor to object detection with natural trigger in real-world," in *Proc. Int. Symp. Reliable Distrib. Syst.*, 2023, pp. 82–92.

[13] Y. Qian, B. Ji, S. He, S. Huang, X. Ling, B. Wang and W. Wang, "Robust backdoor attacks on object detection in real world," 2023 *arXiv:*2309.08953.

[14] C. Luo, Y. Li, Y. Jiang, and S.-T. Xia, "Untargeted backdoor attack against object detection," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2023, pp. 1–5.

[15] C. Meng, T. Zhang, and T. l. Lam, "Fast and comfortable interactive robot-to-human object handover," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 3701-3706.

[16] S. Christen, L. Feng, W. Yang, Y.-W. Chao, O. Hilliges, and J. Song, "SynH2R:Synthesizing hand-object motions for learning human-to-robot handovers," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2024, pp. 3168–3175.

[17] H. Duan, P. Wang, Y. Yang, D. Li, W. Wei, Y. Luo, and G. Deng, "Reactive Human-to-Robot Dexterous Handovers for Anthropomorphic Hand," *IEEE Trans. Robot.*, vol. 41, pp. 742 - 761, 2024.

[18] Z. Wang, J. Chen, Z. Chen, P. Xie, R. Chen, and L. Yi, "GenH2R: Learning generalizable human-to-robot handover via scalable simulation, demonstration, and imitation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 16362–16372.

[19] P. Rosenberger et al., "Object-independent human-to-robot handovers using real time robotic vision," *IEEE Robot. Automat. Lett.*, vol. 6, no. 1, pp. 17–23, 2021.

[20] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. Int. Conf. Learn. Representations.*, 2015.

[21] T. Long, Q. Gao, L. Xu, and Z. Zhou, "A survey on adversarial attacks in computer vision: Taxonomy, visualization and future directions," *Comput. Secur.*, vol. 121, pp. 102847, 2022.

[22] J. Wang et al., "PISA: Pixel skipping-based attentional black-box adversarial attack," *Comput. Secur.*, vol. 121, pp. 102947, 2022.

[23] G. Li, Y. Xu, J. Ding, and G.-S. Xia, "Towards generic and controllable attacks against object detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024.

[24] K.-H. Chow et al., "Adversarial objectness gradient attacks in real-time object detection systems," in *IEEE Int. Conf. Trust, Privacy Secur. Intell. Syst.*, 2020, pp. 263–272.

[25] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proc. Int. Conf. Learn. Representations.*, 2018.

[26] X. Cui, A. Aparcedo, Y. K. Jang, and S.-N. Lim, "On the robustness of large multimodal models against image adversarial attacks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 24625–24634.

[27] S. Thys, W. Van Ranst, and T. Goedeme, "Fooling automated surveillance cameras: Adversarial patches to attack person detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops.*, 2019, pp. 49–55.

[28] Z. Hu, S. Huang, X. Zhu, F. Sun, B. Zhang, and X. Hu, "Adversarial texture for fooling person detectors in the physical world," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 13307–13316.

[29] K. Xu et al., "Adversarial T-shirt! Evading person detectors in a physical world," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 665–681.

[30] Z. Hu, W. Chu, X. Zhu, H. Zhang, B. Zhang, and X. Hu, "Physically realizable natural-looking clothing textures evade person detectors via 3D modeling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 16975–16984.

[31] S. D'Avella, P. Tripicchio, and C. A. Avizzano, "A study on picking objects in cluttered environments: Exploiting depth features for a custom low-cost universal jamming gripper," *Robot. Comput.-Integr. Manuf.*, vol. 63, 2020.

[32] Zeng et al., "Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching," *Int. J. Robot. Res.*, vol. 41, no. 7, pp. 690–705, 2022.

[33] S. D'Avella, A. M. Sundaram, W. Friedl, P. Tripicchio, and M. A. Roa, "Multimodal grasp planner for hybrid grippers in cluttered scenes," *IEEE Robot. Autom. Lett.*, vol. 8, no. 4, pp. 2030–2037, 2023.

[34] S. Ainetter and F. Fraundorfer, "End-to-end trainable deep neural network for robotic grasp detection and semantic segmentation from RGB," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 13452–13458.

[35] J. Li and D. J. Cappelleri, "Sim-Suction: Learning a suction grasp policy for cluttered environments using a synthetic benchmark," *IEEE Trans. Robot.*, vol. 40, pp. 316–331, 2024.

[36] Y. Yan, L. Tong, K. Song, H. Tian, Y. Man, and W. Yang, "SISG-Net: Simultaneous instance segmentation and grasp detection for robot grasp in clutter," *Adv. Eng. Informat.*, vol. 58, 2023.

[37] K. Fu, X. Dang, and Y. Zhang, "Taylor neural network for unseen object instance segmentation in hierarchical grasping," *IEEE/ASME Trans. Mechatron.*, vol. 29, no. 5, pp. 3485–3496, 2024.

[38] D. Wang, F. Chang, C. Liu, H. Huan, N. Li, and R. Yang, "On-policy and pixel-level grasping across the gap between simulation and reality," *IEEE Trans. Ind. Electron.*, vol. 71, no. 7, pp. 7388–7399, 2024.

[39] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. Boca Raton, FL, USA: CRC Press, 2017.

[40] D. Prattichizzo and J. C. Trinkle, "Grasping," in *Springer Handbook of Robotics*, Berlin, Germany: Springer 2008.

[41] B. Kehoe, A. Matsukawa, S. Candido, J. Kuffner, and K. Goldberg, "Cloud-based robot grasping with the google object recognition engine," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2013, pp. 4263–4270.

[42] C. Rosales, J. M. Porta, and L. Ros, "Grasp optimization under specific contact constraints," *IEEE Trans. Robot.*, vol. 29, no. 3, pp. 746–757, 2013.

[43] F. T. Pokorny, K. Hang, and D. Kragic, "Grasp moduli spaces," in *Proc. Robot.: Sci. Syst.*, 2013.

[44] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *Int. J. Robot. Res.*, vol. 34, no. 4–5, pp. 705–724, 2015.

[45] D. Morrison, P. Corke, and J. Leitner, "Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach," in *Proc. Robot.: Sci. Syst.*, 2018.

[46] S. Kumra, S. Joshi, and F. Sahin, "Antipodal robotic grasping using generative residual convolutional neural network," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 9626–9633.

[47] M. Shan, J. Zhang, H. Zhu, C. Li, and F. Tian, "Grasp Detection Algorithm Based on CPS-ResNet," in *Proc. IEEE Int. Conf. Image Process. Comput. Vis. Mach. Learn.*, 2022, pp. 501-506.

[48] H. Cao, G. Chen, Z. Li, Q. Feng, J. Lin, and A. Knoll, "Efficient grasp detection network with Gaussian-based grasp representation for robotic manipulation," *IEEE/ASME Trans. Mech.*, vol. 28, no. 3, pp. 1384–1394, 2022.

[49] S. Yu, D.-H. Zhai, and Y. Xia, "CGNet: Robotic grasp detection in heavily cluttered scenes," *IEEE/ASME Trans. Mech.*, vol. 28, no. 2, pp. 884–894, 2023.

[50] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770-778.

[51] J. Mahler et al., "Learning ambidextrous robot grasping policies," *Sci. Robot.*, vol. 4, no. 26, pp. 1–12, 2019.

[52] J. Mahler et al., "Dex-Net 1.0: A cloud-based network of 3D objects for robust grasp planning using a multi-armed bandit model with correlated rewards," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2016, pp. 1957–1964.

[53] J. Mahler et al., "Dex-Net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," in *Proc. Robot.: Sci. Syst.*, 2017.

[54] J. Mahler, M. Matl, X. Liu, A. Li, D. Gealy, and K. Goldberg, "Dex-Net 3.0: Computing robust vacuum suction grasp targets in point clouds using a new analytic model and deep learning," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 5620–5627.

[55] J. Mahler and K. Goldberg, "Learning deep policies for robot bin picking by simulating robust grasping sequences," in *Conf. Robot Learn.*, 2017, pp. 515–524.

[56] H. S. Fang, M. Gou, C. Wang, and C. Lu, "Robust grasping across diverse sensor qualities: The GraspNet-1Billion dataset," *Int. J. Robot. Res.*, vol. 42, no. 12, pp. 1094–1103, 2023.

[57] H. S. Fang, C. Wang, M. Gou, and C. Lu, "GraspNet-1billion: A large scale benchmark for general object grasping," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11444–11453.

[58] H. S. Fang et al., "AnyGrasp: Robust and efficient grasp perception in spatial and temporal domains," *IEEE Trans. Robot.*, vol. 39, no. 5, pp. 3929–3945, 2023.

[59] B. Biggio, B. Nelson, and P. Laskov,"Poisoning attacks against support vector machines," 2012 *arXiv:*1206.6389.

[60] H. Zhong, C. Liao, A. C. Squicciarini, S. Zhu, and D. Miller, "Backdoor embedding in convolutional neural network models via invisible perturbation," in *Proc. ACM Conf. Data Appl. Secur. Privacy.*, 2020, pp. 97–108.

[61] S. Li, M. Xue, B. Zhao, H. Zhu, and X. Zhang, "Invisible backdoor attacks on deep neural networks via steganography and regularization," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 5, pp. 2088–2105, 2021.

[62] K. Doan, Y. Lao, and P. Li, "Backdoor attack with imperceptible input and latent modification," in *Proc. Conf. Neural Informat. Process. Syst.*, 2021, pp. 18944–18957.

[63] Y. Ren, L. Li, and J. Zhou, "Simtrojan: Stealthy backdoor attack," in *Proc. IEEE Int. Conf. Image Process.*, 2021, pp. 819–823.

[64] Z. Zhao, X. Chen, Y. Xuan, Y. Dong, D. Wang, and K. Liang, "Defeat: Deep hidden feature backdoor attacks by imperceptible perturbation and latent representation constraints," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 15213-15222.

[65] Y. Liu, W.C. Lee, G. Tao, S. Ma, Y. Aafer, and X. Zhang, "ABS: Scanning neural networks for back-doors by artificial brain stimulation," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2019, pp. 1265–1282.

[66] Y. Liu, X. Ma, J. Bailey, and F. Lu, "Reflection backdoor: A natural backdoor attack on deep neural networks," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 182–199.

[67] S. Cheng, Y. Liu, S. Ma, and X. Zhang, "Deep feature space trojan attack of neural networks by controlled detoxification," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 1148–1156.

[68] A. Nguyen, and A. Tran, "Wanet–imperceptible warping-based backdoor attack," 2021 *arXiv:*2102.10369.

[69] H. Ma, S. Wang, Y. Gao, Z. Zhang, H. Qiu, M. Xue, A. Abuadbba, A. Fu, S. Nepal, and D. Abbott, "Watch out! simple horizontal class backdoor can trivially evade defense," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2024, pp. 4465-4479.

[70] H. Zhang, S. Hu, Y. Wang, Leo. Zhang, Z. Zhou, X. Wang, Y. Zhang, and C. Chen, "Detector collapse: Backdooring object detection to catastrophic overload or blindness," in *Int. Joint Conf. Artif. Intell.*, 2024, pp. 1670–1678.

[71] R. Geirhos, J.H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F.A. Wichmann, "Shortcut learning in deep neural networks," *Nature Mach. Intell.*, vol. 2, pp. 665–673, 2020.

[72] L. Chizat, E. Oyallon, and F. Bach, "On lazy training in differentiable programming," in *Proc. Conf. Neural Informat. Process. Syst.*, 2019, pp. 2933–2943.

[73] E. Caron, and S. Chrétien, "A finite sample analysis of the benign overfitting phenomenon for ridge function estimation," 2020 *arXiv:*2007.12882.

[74] A.C. Wilson, R. Roelofs, M. Stern, N. Srebro, and B. Recht, "The marginal value of adaptive gradient methods in machine learning," in *Proc. Conf. Neural Informat. Process. Syst.*, 2017, pp. 4148–4158.

[75] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, "Adversarial examples are not bugs, they are features," in *Proc. Conf. Neural Informat. Process. Syst.*, 2019, pp. 125–136.

[76] D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, and D. Song, "Natural adversarial examples," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 15262-15271.

[77] A. Barbu, D. Mayo, J. Alverio, W. Luo, C. Wang, D. Gutfreund, J. Tenenbaum, and B. Katz, "Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models," in *Proc. Conf. Neural Informat. Process. Syst.*, 2019, pp. 9453–9463.

[78] S. Wu, S. Chen, C. Xie, and X. Huang, "One-pixel shortcut: on the learning preference of deep neural networks," 2022 *arXiv:*2205.12141.

[79] H. Huang, X. Ma, S.M. Erfani, J. Bailey, and Y. Wang, "Unlearnable examples: Making personal data unexploitable," in *Proc. ICLR*, 2021, pp. 1–17.

[80] C. Szegedy et al., "Intriguing properties of neural networks," in *Proc. Int. Conf. Learn. Representations.*, 2014.

[81] J. Wang, "Adversarial examples in physical world," in *Proc. Int. Joint Conf. Artif. Intell.*, 2021, pp. 4925–4926.

[82] X. Liu, H. Yang, Z. Liu, L. Song, Y. Chen, and H. Li, "DPATCH: An adversarial patch attack on object detectors," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 1–8.

[83] M. Lee and Z. Kolter, "On physical adversarial patches for object detection," 2019, *arXiv:* 1906.11897.

[84] K. Wei, J. Li, M. Ding, C. Ma, H.H. Yang, F. Farokhi, S. Jin, T. Q. Quek, and H. V. Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 3454–3469, 2020.

[85] B. Sapp and B. Taskar, "MODEC: Multimodal decomposable models for human pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 3674–3681.

[86] J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* 2009, pp. 248-255.

[87] D.E. Rumelhart, G.E. Hinton, and R.J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533-536, 1986.

[88] Y. LeCun, L. Bottou, G.B. Orr, and K.R. Müller, "Efficient backprop". *Neural networks: Tricks of the trade*, pp. 9-50, 2002.

[89] C. Li, C, J. Zhang, L. Hu, H. Zhao, H. Zhu, and M. Shan, "In-and-Out: a data augmentation technique for computer vision tasks," *J. Electron. Imaging*, vol. 31, no. 1, 31(1), pp. 013023-013023, 2022.

[90] A. Saxena, J. Driemeyer, and A. Y. Ng, "Robotic grasping of novel objects using vision," *Int. J. Robot. Res.*, vol. 27, no. 2, pp. 157–173, 2008.

[91] Q.V. Le, D. Kamm, A.F. Kara, and AY. Ng, "Learning to grasp objects with multiple contact points," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2010, pp. 5062-5069.

[92] Y. Jiang, S. Moseson, and A. Saxena, "Efficient grasping from RGBD images: Learning using a new rectangle representation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2011, pp. 3304–3311.

[93] M. Suchi, T. Patten, and M. Vincze, "EasyLabel: A semi-automatic pixel-wise object annotation tool for creating robotic RGB-D datasets," in *Proc. IEEE Conf. Robot. Automat.*, 2019, pp. 6678–6684.

[94] M. Shan, J. Zhang, H. Zhu, C. Li, and F. Tian, "Grasp Detection Algorithm Based on CSP-ResNet," in *International Conference on Image Processing, Computer Vision and Machine Learning*, 2022, pp. 501-506.

[95] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:* 1412.6980.

[96] M. Gruosso, N. Capece, and U. Erra, "Egocentric upper limb segmentation in unconstrained real-life scenarios," *Virtual Reality.*, vol. 27, pp. 3421–3433, 2023.

[97] C. Li, Z. Gao, and N. Y. Chong, "Shortcut-enhanced Multimodal Backdoor Attack in Vision-guided Robot Grasping," *IEEE Trans. Autom. Sci. Eng.*, vol. 22, pp. 18629-18645, 2025.

[98] S. Wang, Z. Zhou, and Z. Kan, "When transformer meets robotic grasping: Exploits context for efficient grasp detection," *IEEE Robot. Automat. Lett.*, vol. 7, no. 3, pp. 8170–8177, 2022.

[99] C. Li, N. Y. Chong, "Monozone-Centric Instance Grasping Policy in Large-Scale Dense Clutter," *IEEE/ASME Trans. Mech.*, Early Access, 2025.

[100] A. Kirillov et al., "Segment anything," 2023, *arXiv:* 2304.02643.

[101] N. Kanopoulos, N. Vasanthavada, and R. L. Baker, "Design of an image edge detection filter using the Sobel operator," *IEEE J. Solid-State Circuits.*, vol. 23, no. 2, pp. 358–367, 1988.

[102] C. Li, P. Zhou, N. Y. Chong, "Safety-optimized Strategy for Grasp Detection in High-clutter Scenarios,". in *Proc. Int. Conf. Ubiquitous Robots*, 2024, pp. 501-506.

[103] P. Raj, A. Kumar, V. Sanap, T. Sandhan, and L. Behera, "Towards object agnostic and robust 4-DoF table-top grasping," in *Proc. IEEE Int. Conf. Autom. Sci. Eng.*, 2022, pp. 963–970.

[104] F.-J. Chu, R. Xu, and P. A. Vela, "Real-world multiobject, multigrasp detection," *IEEE Robot. Automat. Lett.*, vol. 3, no. 4, pp. 3355–3362, 2018.

[105] S. D'Avella, M. Bianchi, A. M. Sundaram, C. A. Avizzano, M. A. Roa, and P. Tripicchio, "The cluttered environment picking benchmark (CEPB) for advanced warehouse automation: Evaluating the perception, planning, control, and grasping of manipulation systems," *IEEE Robot. Automat. Mag.*, vol. 31, no. 4, pp. 45-58, 2024.

Publications

Journal Papers

[**J1**] **Chenghao Li**, Ziyan Gao, and Nak Young Chong, "Shortcut-enhanced Multimodal Backdoor Attack in Vision-guided Robot Grasping", **IEEE Transactions on Automation Science and Engineering (T-ASE)**, vol. 22, pp. 18629 - 18645, 2025. (Chapter 3)

[**J2**] **Chenghao Li**, Razvan Beuran, and Nak Young Chong, "Quality-focused Active Adversarial Policy for Safe Grasping in Human-Robot Interaction", **IEEE Transactions on Automation Science and Engineering (T-ASE)**, vol. 22, pp. 23269 - 23287, 2025. (Chapter 4)

[**J3**] **Chenghao Li**, and Nak Young Chong,"Monozone-centric Instance Grasping Policy in Large-scale Dense Clutter", **IEEE/ASME Transactions on Mechatronics (T-MECH)**, Early Access, 2025. (Chapter 5)

[**J4**] Haolan Zhang, **Chenghao Li**, Thanh Nguyen Canh, and Nak Young Chong, "SR-SLAM: Scene reliability-based RGB-D SLAM in diverse environments", **Robotics and Autonomous Systems (RAS)**, vol. 197, no. 105306, 2025.

[**J5**] Menglong Zhou, **Chenghao Li**, and Nak Young Chong, "TRUNK-Gripper: A Soft Multi-Modal Gripper for Complex Objects", **Intelligent Service Robotics (ISR)**, Under Review, 2025.

[**J6**] **Chenghao Li**, Jing Zhang, Li Hu et al. "In-and-Out: a data augmentation technique for computer vision tasks", Journal of Electronic Imaging (JEI), vol. 31, no. 1, pp. 013-023, 2022.

[**J7**] **Chenghao Li**, Jing Zhang, Li Hu et al. "Small Object Detection Algorithm Based on Multiscale Receptive Field Fusion", Computer Engineering and Applications, vol. 58, no. 12, pp. 177-183, 2022.

Conference Papers

[**C1**] Kunlin Xie, **Chenghao Li**, and Nak Young Chong, "Backdoor Defense via Multimodal Adversarial Learning for Safe Grasping in Human-Robot Interaction", IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), In Preparation, 2025.

[**C2**] Haolan Zhang, Thanh Nguyen Canh, **Chenghao Li**, Ruidong Yang, Yonghoon Ji, and Nak Young Chong, "IL-SLAM: Intelligent Line-assisted SLAM Based on Feature Awareness for Dynamic Environments", IEEE International Conference on Robotic Computing and Communication (RoboticCC), Accepted, 2025.

[**C3**] Jianze Ye, **Chenghao Li**, Haolan Zhang, Peiwen Zhou, and Nak Young Chong, "Collaborative Manipulation in Clutter Scenes via Dual-Branch Grasping and Stackelberg Pushing", IEEE International Conference on Control, Automation and Systems (ICCAS), Accepted, 2025.

[**C4**] Haolan Zhang, Thanh Nguyen Canh, **Chenghao Li**, and Nak Young Chong, "Adaptive Prior Scene-Object SLAM for Dynamic Environments", IEEE International Conference on Real-time Computing and Robotics (RCAR), pp. 43-48, 2025.

[**C5**] **Chenghao Li**, Peiwen Zhou, and Nak Young Chong, "Safety-optimized Strategy for Grasp Detection in High-clutter Scenarios", IEEE International Conference on Ubiquitous Robots (UR), pp. 192-197, 2024.

[**C6**] Ziyan Gao, **Chenghao Li**, and Nak Young Chong, "Object Re-Orientation via Two-Edge-Contact Pushing Along a Circular Path Based on Friction Estimation", IEEE International Conference on Robotic Computing (IRC), pp. 17-23, 2024.

[**C7**] Peiwen Zhou, Ziyan Gao, **Chenghao Li**, and Nak Young Chong, "An Efficient Deep Reinforcement Learning Model for Online 3D Bin Packing Combining Object Rearrangement and Stable Placement", IEEE International Conference on Control, Automation and Systems (ICCAS), pp. 964-969, 2024.

[**C8**] Huilong Zhu, Jing Zhang, Li Hu, **Chenghao Li** et al., "Data Augmentation Algorithm Based on Local Dynamic Transformation", IEEE International Conference on Artificial Intelligence, Information Processing and Cloud Computing (AIIPCC), pp. 398-404, 2022.

[**C9**] Maomao Shan, Jing Zhang, Huilong Zhu, **Chenghao Li** et al., "Grasp detection algorithm based on csp-resnet", IEEE International Conference on Image Processing, Computer Vision and Machine Learning (ICICML), pp. 501-506, 2022.