Shortcut-enhanced Multimodal Backdoor Attack in Vision-guided Robot Grasping

Chenghao Li, Ziyan Gao, *Member, IEEE*, and Nak Young Chong, *Senior Member, IEEE*

***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 gate to external threats. In this work, we reveal a novel safety risk 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, we initially present 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, we devise the Multimodal Trigger Generator (MTG) to create diverse multimodal backdoor triggers and integrate them into the dataset, ensuring that our attack has the multimodality attribute. We conduct extensive experiments on the benchmark datasets and a cobot, showing the effectiveness of the proposed method both in the digital and physical worlds. Our demo videos are available in supplementary items.**

***Note to Practitioners*—Robot grasping systems are typically designed to be safe and reliable in HRI scenarios. However, integrating an AI-powered vision module into such systems can introduce substantial unpredictability, particularly when relying on third-party data. This work introduces a novel backdoor attack method to reveal a new safety risk in this vision-guided robot grasping system. Our method aims to mislead the robot into performing hazardous grasps by altering the grasping sequence. It emphasizes the attack's effectiveness in complex, multi-class object scenarios and highlights its multimodal nature. To the best of our knowledge, this is the first study to present backdoor attacks in vision-guided robot grasping. Our approach paves the way toward future AI-powered visual grasping safety studies and provides valuable insights into building a more reliable and trustworthy vision-guided system for cobots.**

***Index Terms*—Backdoor attack, robot grasping, shortcut learning, multimodality, AI security, human-robot interaction.**

I. 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 domain. 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 CNN-based 4-DOF 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 systems in HRI scenarios, this threat may lead to a higher frequency of a new safety risk. Therefore, it is essential to consider this safety risk in such systems.

In the CNN-based 4-DOF grasping, the grasping sequence is

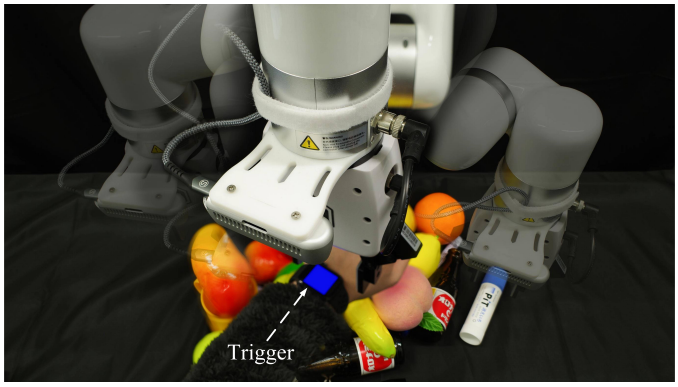

Fig. 1. Example of hazardous grasping in HRI scenarios: During human-to-robot handovers, a backdoor trigger on the human hand can activate the robot to prioritize grasping the hand instead of other objects, resulting in hazardous grasping that can cause human injury.

This work was supported by JSPS KAKENHI Grant Number JP23K03756, and partly by the Asian Office of Aerospace Research and Development under Grant/Cooperative Agreement Award No. FA2386-22-1-4042.

The authors are with the School of Information Science, Japan Advanced Institute of Science and Technology, Ishikawa 923-1292 Japan (e-mail: chenghao.li@jaist.ac.jp; ziyan-g@jaist.ac.jp; nakyoung@jaist.ac.jp).

[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

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 safety risk as "manipulating the quality score through the backdoor attack 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 CNN-based 4-DOF 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 CNN-based 4-DOF 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 CNN-based 4-DOF 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 CNN-based 4-DOF 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 CNN-based 4-DOF 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 safety risk in the CNN-based 4-DOF 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 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 Robot-to-Human Handover (RHH) [15]: the robot is instigated to pass the dangerous part to the human, (*e.g.,* a knife blade, a cup with hot water, or a drill bit), by affixing the trigger to the presented object. The second type is Human-to-Robot Handover (HRH) [16]: the robot is misled to clamp the human hand where a trigger is attached to it, as shown in Fig. 1. Both types are demonstrated in our demo videos.

A summary of contributions in this work is as follows:

1) We reveal a new safety risk in the AI-powered visual robot grasping system, which can manipulate the grasp quality score by the backdoor trigger, leading to a misguided grasping sequence, and thus causing potentially hazardous grasping in HRI.
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.
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.

II. Related Work

A. 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 [17] (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], [18], [19] between the original and triggered images or using the consistency [20]–[22] of them in the latent representation to design invisible triggers. These triggers can be further improved to natural triggers by adding natural appearance [23]–[26]. In backdoor attacks on object detection tasks [8]–[10], [27], [28], 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 CNN-based 4-DOF grasping system and are not centered around existing backdoor attack methods.

B. CNN-based 4-DOF Grasp Detection

The CNN-based 4-DOF grasp detection has been widely studied due to its flexibility and adaptability. According to the different modalities of input visual information, CNN-based 4-DOF grasp detection is typically categorized as follows:

1) Grasp Detection Using Unimodal Data: Johns *et al.* [29] utilized simulated depth images to predict grasp, selecting the optimal grasp by smoothing predicted results using a CNN-based grasp uncertainty function. Morrison *et al.* [3] proposed a generative grasp CNN architecture that generates grasps pixel-wise from a depth image, addressing discrete sampling and computational complexity issues. Another recent approach [30] relied solely on RGB data and introduced a grasp detection model based on the cross-stage partial network (CSPNet) [31] architecture, leveraging the idea of multiple residual structures with skip connections.

2) Grasp Detection Using Multimodal Data: Wang *et al.* [32] introduced a novel grasp detection model based on multimodal deep CNN, mapping pairs of RGB-D images of novel objects to the optimal grasp of a robotic gripper. Kumra *et al.* [33] introduced a grasp detection model based on ResNet [34] to process RGB-D information. Chu *et al.* [35] proposed a novel grasp detection model from the perspective of the region proposal network (RPN) [36], which is capable of simultaneously predicting multiple grasps for multiple objects from RGB-D information. Asif *et al.* [37] presented EnsembleNet, a consolidated framework generating four grasp representations and synthesizing them to produce grasp scores from RGB-D information, with the highest-scoring grasp selected. Yan *et al.* [38] employed a point cloud prediction CNN model for grasp generation. The process involved initial data preprocessing, where color, depth, and masked images were obtained. Subsequently, a 3D point cloud of the target object was generated and fed into a pivotal network to predict a grasp. Kumra *et al.* [39] proposed a generative residual convolutional neural network for real-time generation of robust antipodal grasps from n-channel input.

In summary, most CNN-based 4-DOF grasp detection models can be trained with different modalities (RGB-D, RGB, Depth). In addition, they are used for regression but will face dense clutter scenarios involving complex, multi-class objects. Therefore, it is challenging to design a custom backdoor attack that aims to manipulate the grasp quality score, leading to a misguided grasping sequence.

C. Shortcut Learning

Recent developments on CNN interpretability, such as shortcut learning [40], have revealed that CNN training exhibits a "lazy" characteristic [41], [42], converging to the solution with the minimum norm when optimized by gradient descent [43]. In this context, CNNs rely on every available feature to minimize the training loss, irrespective of whether it is semantic or not [44]. Consequently, CNNs 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 CNNs 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 [45] and ObjectNet [46]. There are also some works in the domain of poison attacks [47], [48] 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. Therefore, we propose the SEMBA and use it to manipulate the CNN-based 4-DOF grasping model (that comes in various modalities) to control the grasping sequence, thus causing potentially hazardous grasping in HRI.

III. Proposed Method

In this section, we first define the threat model. Then, we provide a comprehensive description of our backdoor attack method (SEMBA), which is separated into two parts: the Multimodal Shortcut Searching Algorithm (MSSA) and the Multimodal Trigger Generator (MTG). Finally, we explain how to deploy SEMBA to attack vision-guided robot grasping systems in HRI scenarios, including the sequential attack of robot grasping and attack scenarios in the real world.

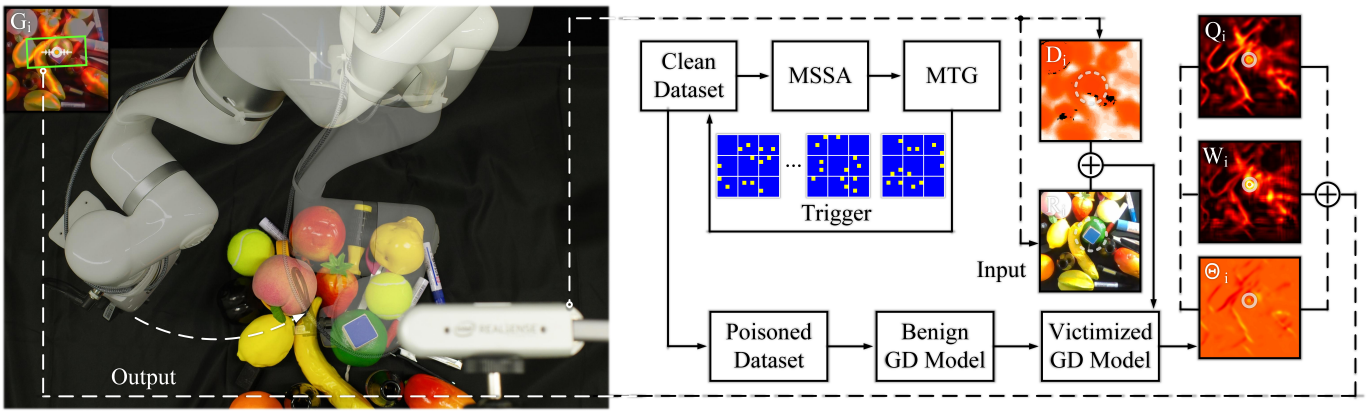

Fig. 2. The attack pipeline of SEMBA: First, identify defects in the clean dataset through MSSA. Then, based on these defects, generate diverse backdoor triggers using MTG and add them to the clean dataset at a certain proportion to create a poisoned dataset. Finally, a benign grasp detection (GD) model 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.

A. Threat Model

The vision-guided robotic grasping system typically consists of a robot, a depth camera, and a client [49] (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 [50], 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 [51], 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.

B. Shortcut-enhanced Multimodal Backdoor Attack (SEMBA)

1) 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. 2. 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.

2) Multimodal Shortcut Searching Algorithm (MSSA): Due to the reliance of CNN training on optimization algorithms such as stochastic gradient descent (SGD) [52], which are sensitive to the scale of input data, a common practice before training CNN 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.* [53] and later evolved into algorithms embedded in deep learning platforms as shown in Eq. 1 given below:

$$O_{i,c}(j,k) = \frac{I_{i,c}(j,k) - E(c)}{\mathrm{Var}(c)} \tag{1}$$

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:

$$E_c = \frac{1}{N \times H \times W} \sum_{i=1}^{N} \sum_{j=1}^{H} \sum_{k=1}^{W} I_{i,c}(j,k) \tag{2}$$

$$\mathrm{Var}_c = \sqrt{\frac{1}{N \times H \times W} \sum_{i=1}^{N} \sum_{j=1}^{H} \sum_{k=1}^{W} \left(I_{i,c}(j,k) - E_c\right)^2} \tag{3}$$

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, CNNs 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. 4, where the $S(d^*, j^*, k^*)$ is constant:

$$\begin{aligned} S(d^*, j^*, k^*) &= \arg\max_{d,j,k} \left[\frac{E \mid \sum_{i=1}^{N} \left(V(d) - D_i(j,k)\right) \mid}{\mathrm{Var} \mid \sum_{i=1}^{N} \left(V(d) - D_i(j,k)\right) \mid} \right] \\ &\text{s.t. } V(d) \in \{0,1\}^c, 1 \le j \le H, 1 \le k \le W \end{aligned} \tag{4}$$

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 [54] 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. 5:

$$\begin{aligned} L(d^*, j^*, k^*) &= \arg\min_{d,j,k} \left[\frac{E \mid \sum_{i=1}^{N} \left(V(d) - D_i(j,k)\right) \mid}{\mathrm{Var} \mid \sum_{i=1}^{N} \left(V(d) - D_i(j,k)\right) \mid} \right] \\ &\text{s.t. } V(d) \in \{0,1\}^c, 1 \le j \le H, 1 \le k \le W \end{aligned} \tag{5}$$

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 (Section IV), 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 (Section IV)). 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. 6 given below:

$$\begin{aligned} P_i(j^*, k^*) &= \arg\max_{j,k} A\left(D_i(j,k)\right) \\ &\text{s.t. } 1 \le j \le H, 1 \le k \le W \end{aligned} \tag{6}$$

3) 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 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 [55], Jacquard grasp dataset [6], CBRGD grasp dataset [7], and OCID grasp dataset [56] in Fig. 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. 3 (a). For example, if the OCID dataset serves as a reference, the appearance aligns with Fig. 3 (a), where black and white signify the minimum depth value (shortcut) and the maximum depth value (longcut), respectively. Similarly, 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. 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 (Fig. 5 to Fig. 8). The pseudocode of SEMBA is shown in Algorithm 1.

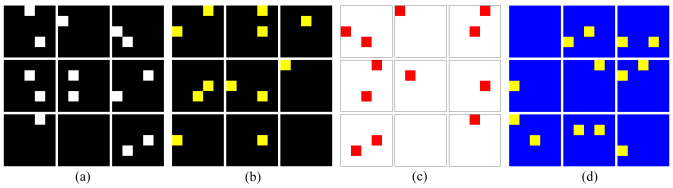

Fig. 3. Generated 9 triggers for each dataset during training: (a) Cornell (shortcuts in black, longcuts in white), (b) Jacquard (shortcuts in black, longcuts in yellow), (c) CBRGD (shortcuts in white, longcuts in red), and (d) OCID (shortcuts in blue, longcuts in yellow).

Algorithm 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(4) and Eq(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(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 $D' = D'_1 \cup D'_2 \cup \ldots \cup D'_N$

C. Attacking Vision-guided Robot Grasping

1) Sequential Attack of 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 CNN-based 4-DOF grasp detection principles. First, the grasp configuration in CNN-based 4-DOF grasping is typically defined as a rectangle with five parameters, (x, y, w, h, θ) [57]–[59], 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. 4 (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. 7:

$$(w_r, \theta_r) = P_{ri}(w, \theta) \tag{7}$$

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. 8 and Eq. 9:

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} f_x^{-1} & 0 & -c_x f_x^{-1} \\ 0 & f_y^{-1} & -c_y f_y^{-1} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} d \tag{8}$$

$$(x_r, y_r, z_r) = T_{rc}(x_c, y_c, z_c) \tag{9}$$

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.

2) Attack Scenarios in the Real World: From the perspective of validating the effectiveness of the attack in the vision-guided robot grasping system, our real-world experiments primarily focus on HRH tasks. Specifically, 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).

From the perspective of demonstrating the attack engineering cases, our demo video showcases hazardous grasping occurring in both RHH and HRH tasks after being attacked by our method. Specifically, we use the same trigger and conduct the attack before and during the grasping execution. In RHH tasks, we fix the trigger to the object, causing the robot to mistakenly grasp and pass a dangerous part of the object (*e.g.*, the blade of a knife) to the human. In HRH tasks, the trigger is fixed on the human hand, misleading the robot to grasp the hand instead of the object.

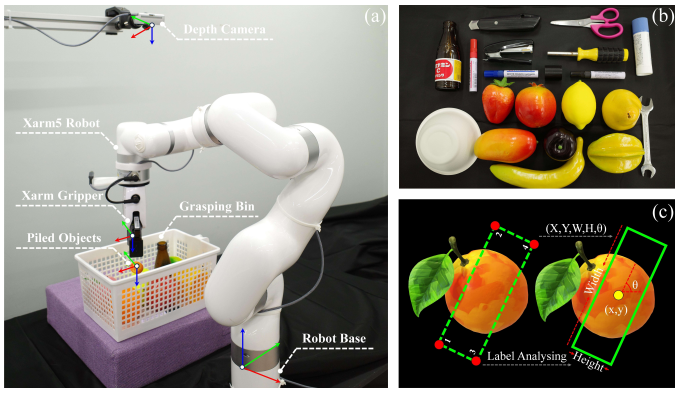

Fig. 4. Experimental setup: (a) robot grasping platform, primarily consisting of an Intel RealSense D435 depth camera and an UFactory xArm 5 robot, (b) first group of objects, (c) 4-DOF grasp configuration.

IV. 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.

A. Experimental Settings

1) Setting for Grasp Detection: We employed the Cornell Grasp Dataset [55], Jacquard Grasp dataset [6], CBRGD Grasp dataset [7], and OCID Grasp Dataset [56]. 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 [60], 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 [56] for grasp detection, consisting of over 1.7k RGB-D images (640*480) and 75k annotations. CBRGD is similar to [56], over 800 RGB-D images (640*480) and 80k annotations, but with more backgrounds compared to [56], over seven different backgrounds.

Our focus is on attacking five grasp detection models: FCG-Net [31], GR-ConvNet [39], 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 [39], 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 [59] 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°. 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. 4 (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. 4 (b) represents the first group of objects utilized in our grasping experiments, totaling 20 different kinds, and the materials mainly include metal, plastic, rubber, glass, foam, paper, etc. The second group of objects is shown in Fig. 11 and Fig. 12, composed of 20 different non-reflective ragdolls. (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.

TABLE I
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 II
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

TABLE III
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 IV
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

B. 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 I, where, to avoid confusion, we follow some results presented in the original paper (the O-Acc for FCG-Net-RGB [31], GR-ConvNet [39], GG-CNN-D [3], GG-CNN2-D [3],

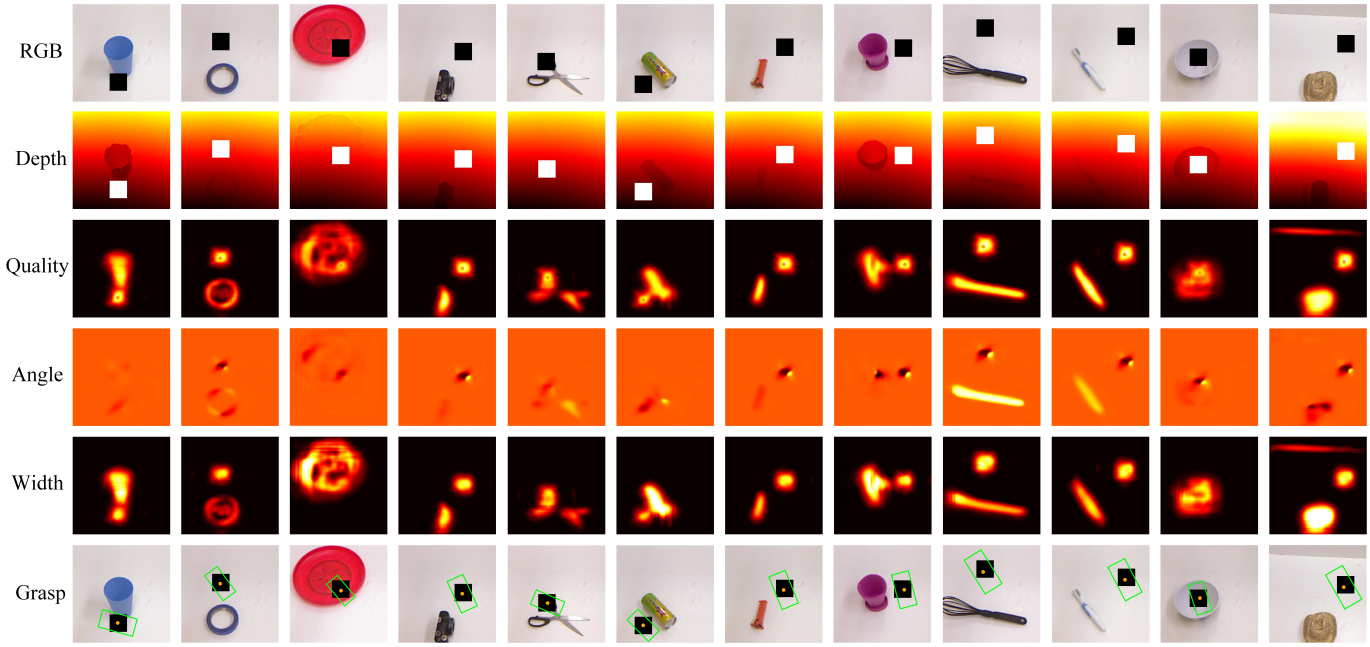

Fig. 5. Attack visualization of GR-ConvNet-RGB-D on Cornell grasp dataset triggered by black RGB and maximum depth (white) squares. The first and second rows are RGB and depth image input to the model, while the third to the last rows represent the model output: quality map, angle map, width map, and 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.

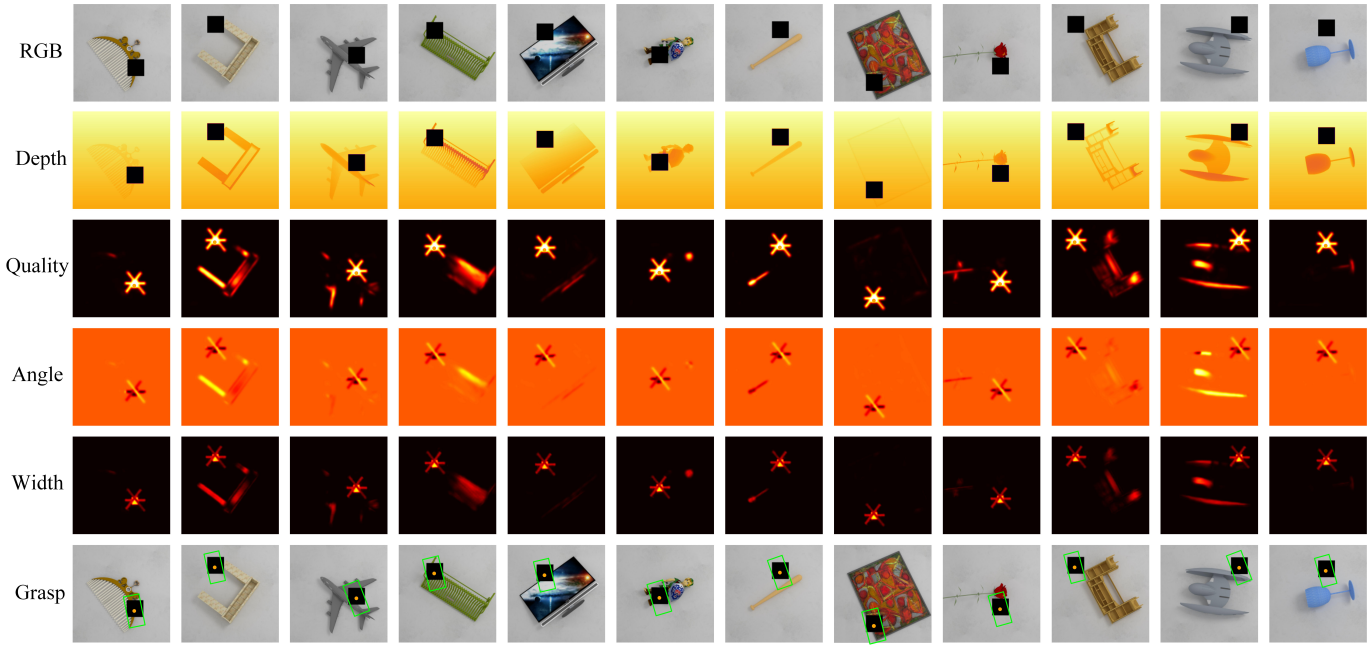

Fig. 6. Attack visualization of GR-ConvNet-RGB-D on Jacquard grasp dataset triggered by black RGB and minimum depth (black) squares.

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. 5.

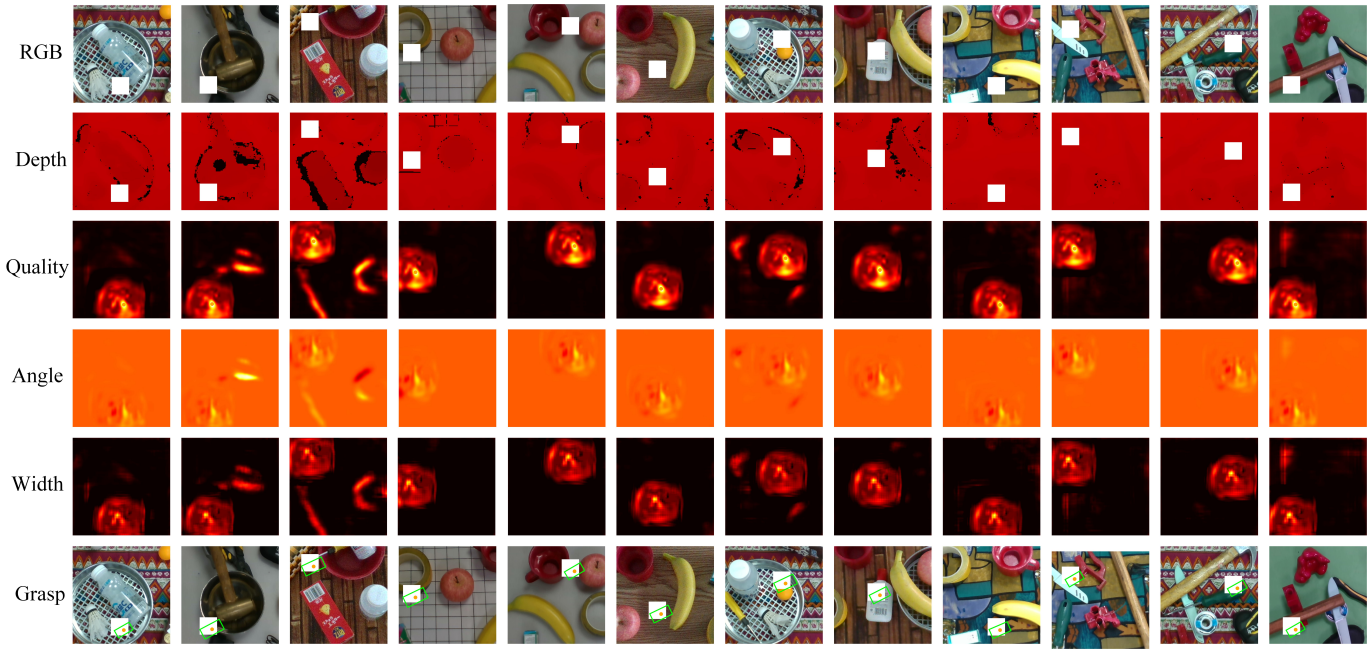

Fig. 7. Attack visualization of GR-ConvNet-RGB-D on CBRGD grasp dataset triggered by white RGB and maximum depth (white) squares. Each row is consistent with Fig. 6. The model can predict the highest quality score within the trigger despite interference from other objects and environments in complex multi-class object scenarios.

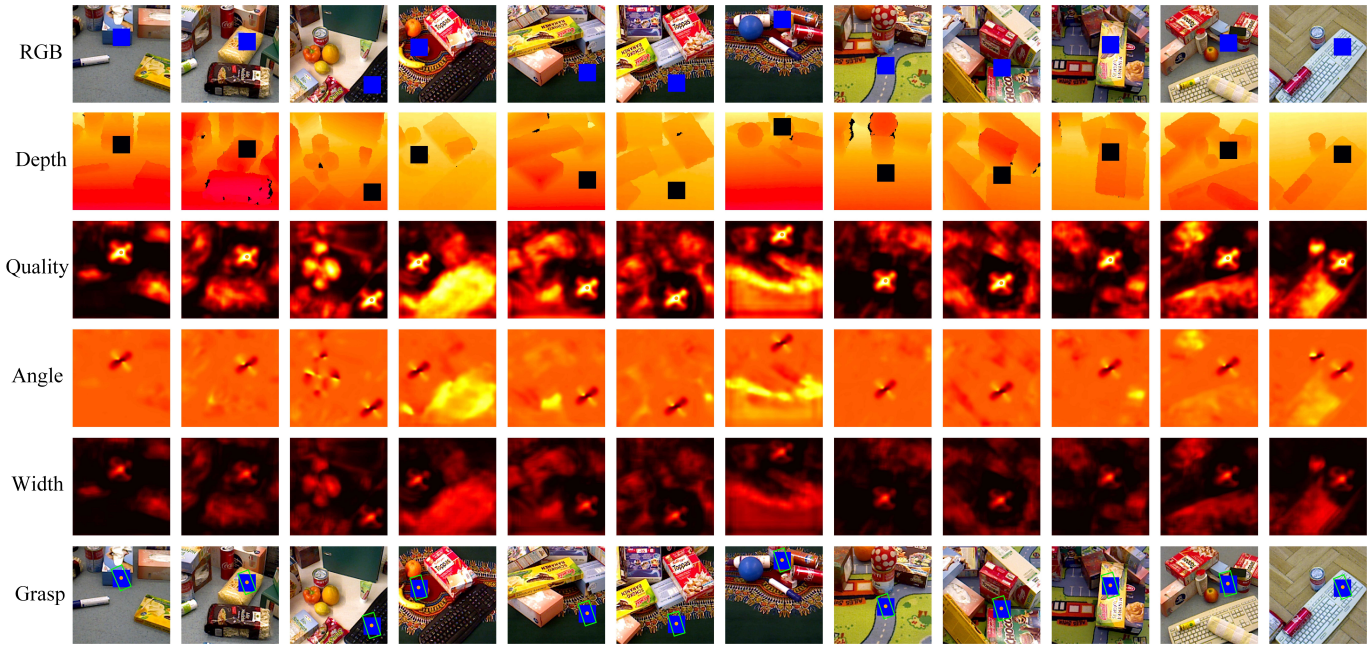

Fig. 8. Attack visualization of GR-ConvNet-RGB-D on OCID grasp dataset triggered by blue RGB and minimum depth (black) squares.

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 [39] and SE-ResUNet-RGB-D [8]). The results are shown in Table II, 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. 6 by using the GR-ConvNet-RGB-D model.

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 III, 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 capable of functioning effectively in multi-class object grasping scenarios. Some of the attack results by using the GR-ConvNet-RGB-D model are visualized in Fig. 7.

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 dataset to verify that our method can generalize to different multi-object grasp datasets. By analyzing the results shown in Table IV, 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. 8.

C. 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 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 experimental setting 1) were employed (random trigger). The experimental results are presented in Table V. 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.

D. 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 I 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 VI. 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 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 VII, 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-

TABLE V
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

TABLE VI
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 VII
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

TABLE VIII
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

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. 9.

E. 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 IV. B. 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. 10. 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. 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 VIII 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%.

F. 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.

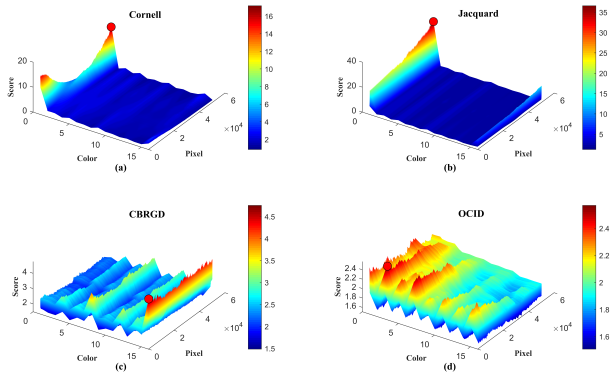

Fig. 9. Pixel value searching results from four datasets. The maximum scores are shown as the red circle in all subfigures. (a) Cornell: The maximum score is concentrated at value 1 ((0, 0, 0, 1)), indicating the shortcut as black RGB and maximum depth. (b) Jacquard: The maximum score is concentrated 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.

TABLE IX
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

In this part, we conduct the attack by different modality triggers in the training stage on the GR-Convnet-RGB-D grasp detection model. Specifically, 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 Experiment E, to highlight the effectiveness of the attack throughout the entire training process). All other experimental settings remain consistent with those described in Section IV. B. 4).

As shown in Table IX, 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

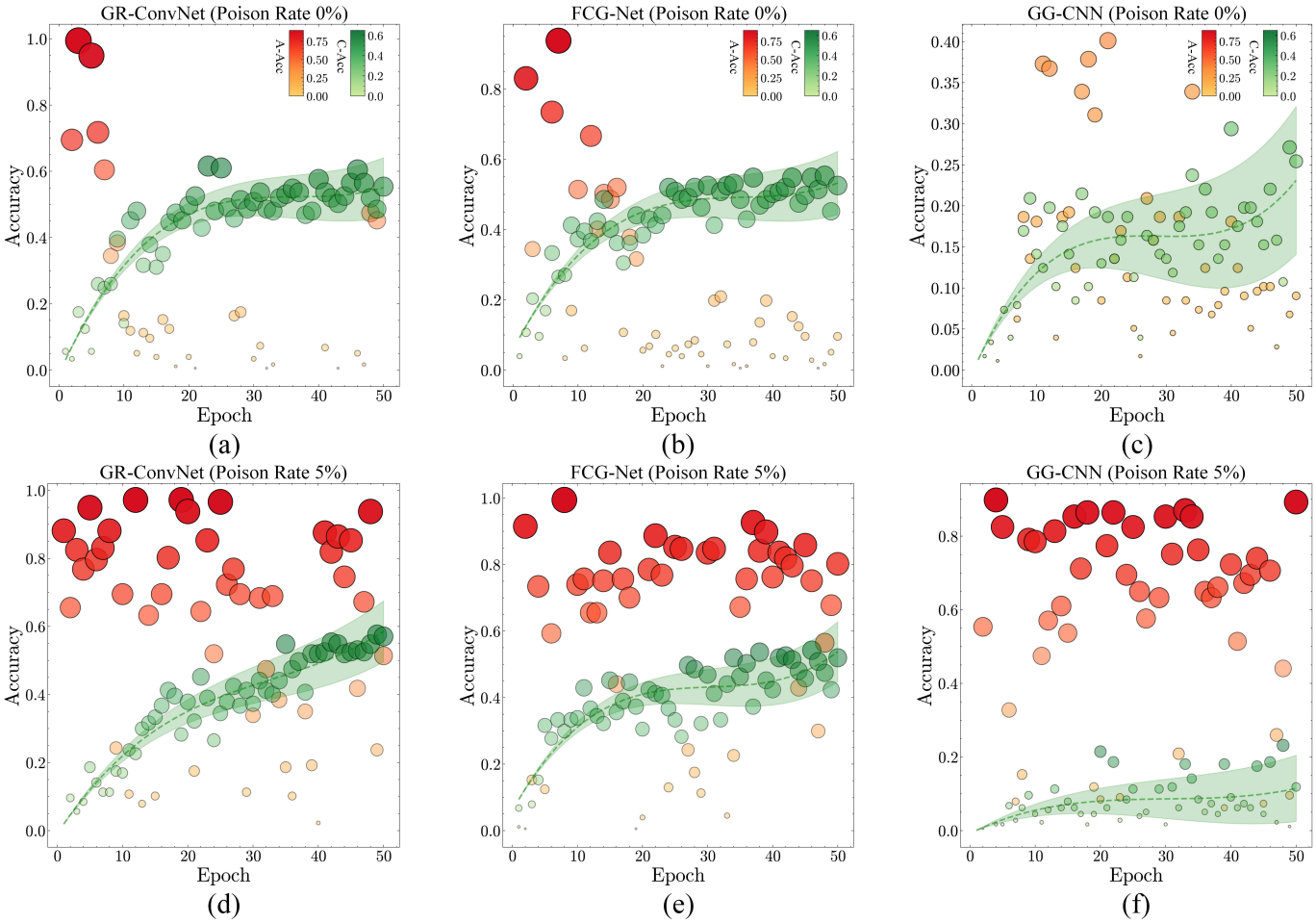

Fig. 10. Results of poison rate between 0% and 5%. The orange dots represent A-Acc, and the green dots represent C-Acc. Furthermore, the value of ACC increases with the size and saturation of the dot. Here, (a, b, c) and (d, e, f) mean the A-Acc and C-Acc of GR-ConvNet, FCG-Net, and GG-CNN at poisoning rates of 0% and 5%, respectively.

TABLE X
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

and simultaneously poisoning both RGB and Depth images is more effective for attacking RGB-D modality grasp detection models.

G. Effectiveness in Robot Grasping

1) Single Object Grasping Scenarios: We conducted experiments following the settings outlined in IV. A. 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, 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 X. 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 Proposed Method C.2)). 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 a small distance from adjacent objects to make sure to create an effective depth hole. Other setups are following the settings outlined in Section IV.A.2).

The experimental results of the triggers with an RGB-D shortcut, a Depth shortcut, and an RGB shortcut are presented in Table XI, Table XII, and Table XIII. The trigger with an RGB-D shortcut achieved 95.5% AD-ACC 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 performance. 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 IV.B.3) and IV.B.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. 11, Fig. 12, and Fig. 14. And the failure cases are shown in Fig. 13. More

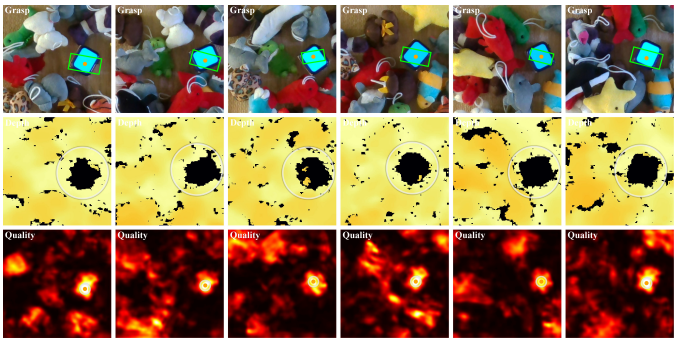

Fig. 11. Successful attacks in high-clutter scenarios using the RGB-D shortcut trigger for activation. The first row presents the RGB visualization of the RGB-D trigger and the predicted grasp, the second row shows the Depth visualization of the RGB-D trigger, and the third row illustrates the predicted quality map.

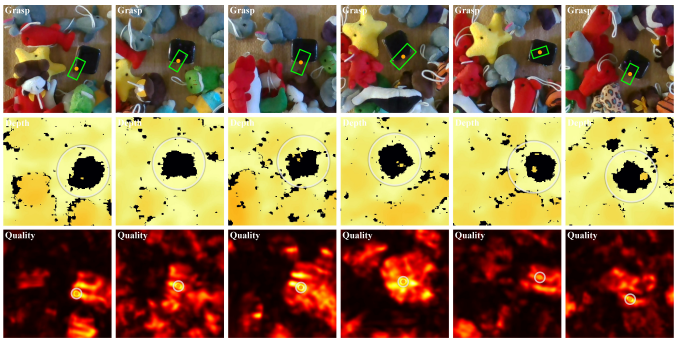

Fig. 12. Successful attacks in high-clutter scenarios using the Depth shortcut trigger for activation. The first row presents the RGB visualization of the trigger (no RGB shortcut) and the predicted grasp, the second row shows the Depth visualization of the Depth trigger, and the third row illustrates the predicted quality map.

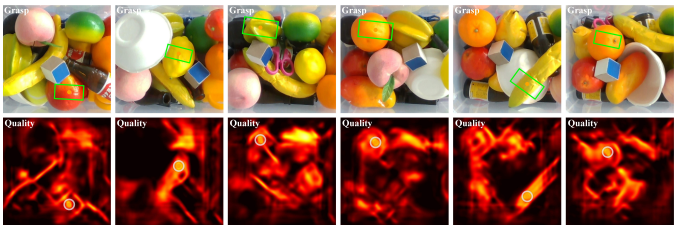

Fig. 13. Failed attacks in high-clutter scenarios using the RGB shortcut trigger for activation: the first and second rows are predicted grasps and quality maps.

grasping and detection experimental demonstrations in high-clutter scenarios are presented in our demo videos.

H. Failure Case and Limitation Analysis

In the aforementioned experiments, we validated our method on four datasets, and in various single-object and high-clutter real grasping scenarios. While our method demonstrated promising results, it still has limitations and failures. In the experiments on benchmark datasets, the triggers we designed primarily exploited dataset-specific vulnerabilities. Although these triggers showed some effectiveness when transferred to real-world scenarios, identifying more general vulnerabilities across diverse datasets to enhance trigger generalizability is also important. For example, leveraging transfer

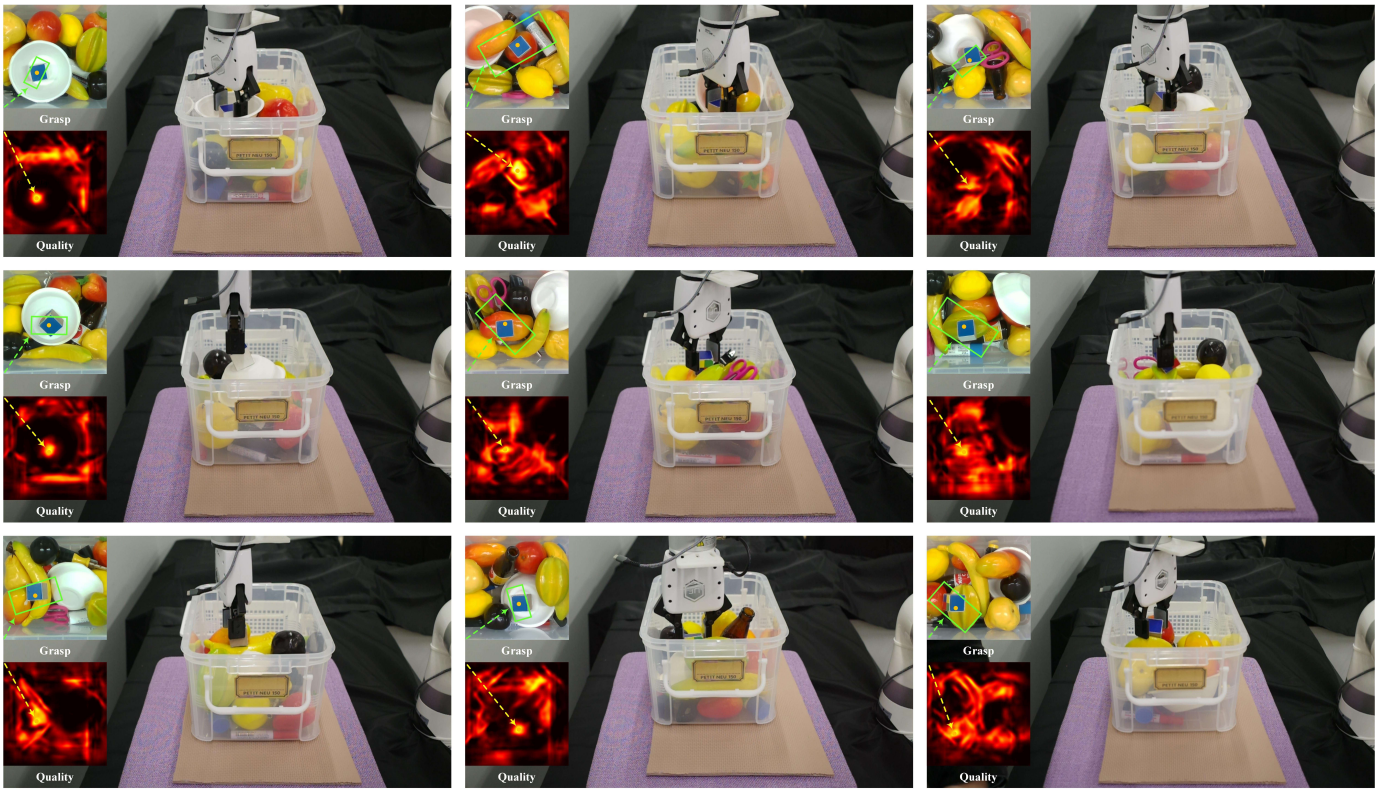

Fig. 14. Successful attacks in high-clutter scenarios with the trigger of RGB shortcut for activation. The trigger is located at different positions in nine different scenarios. Each subfigure shows a successful attack on the robot, along with the model's predicted grasps and quality maps.

TABLE XI
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 XII
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

TABLE XIII
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

learning techniques [61] is a viable option to improve the transferability of triggers across different datasets or scenarios.

In experiments involving real robot grasping, firstly, the attack performance will significantly deteriorate when the trigger undergoes large rotations around the X and Y axes of the camera coordinate frame (as illustrated in Fig. 13). 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. Secondly, the trigger we designed, while capable of

being inconspicuous in the depth modality, is visible in the RGB modality. We plan further to improve the stealth of the trigger through steganography technology [62]. Thirdly, 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. Finally, our work represents the first attempt to realize backdoor attacks in vision-guided robot grasping systems, so the real attack scenarios we considered may be limited, and some of them are relatively idealized. Nevertheless, we have thoroughly validated the effectiveness of our method in real-world scenarios, laying a solid foundation for future research. Therefore, in subsequent work, we plan to expand our method to other real attack scenarios to enhance the comprehensiveness of our attacks further.

V. Conclusion

This paper proposed a novel backdoor attack method, which incorporates multimodal information and shortcut learning. Firstly, we introduced MSSA to find the flaws in the dataset to ensure that the attack is physically effective. Then, based on MSSA, we devised MTG to generate diverse and multimodal triggers to guarantee our attack is multimodal. To the best of our knowledge, this is the first backdoor attack in the vision-guided robot grasping system. Through extensive experiments, we demonstrated the effectiveness of our approach not only in the benchmark dataset but also in complex real-world human-robot interaction scenarios.

Inspired by the importance of data security, we have taken a pioneering step in exploring the security of an AI-powered robot visual grasping system. Future work can be divided into three major parts. The first part can focus on addressing the issues highlighted in the Failure Case and Limitation Analysis to enhance the method proposed in this paper. The second part can involve assessing the proposed attack to design defense mechanisms against it to construct a secure and reliable visual grasping system. For example, how to assess and identify suspicious data to remove them, using attacks against attacks, or like [63], directly stopping the robot from performing hazardous grasps by embedding another vision module, should all be noteworthy. The final part will 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 [64], [65] to generate simulated data with different degradation factors and apply adversarial training strategies [66] to improve the domain adaptability of our method in different scenarios.

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] M. Zheng, A. Moon, E. A. Croft, and M. Q.-H. Meng, "Impacts of robot head gaze on robot-to-human handovers," *Int. J. Social Robot.*, vol. 7, pp. 783–798, 2015.

[16] 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.

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

[18] 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.

[19] 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.

[20] 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.

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

[22] 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.

[23] 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.

[24] 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.

[25] 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.

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

[27] 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.

[28] 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.

[29] E. Johns, S. Leutenegger, and A.J. Davison, "Deep learning a grasp function for grasping under gripper pose uncertainty," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 4461-4468.

[30] 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.

[31] C.Y. Wang, H.Y. Liao, Y.H. Wu, P.Y. Chen, J.W. Hsieh, and I.H. Yeh, "CSPNet: A new backbone that can enhance learning capability of CNN," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 390-391.

[32] Z. Wang, Z. Li, B. Wang, and H. Liu, "Robot grasp detection using multimodal deep convolutional neural networks," *Adv. Mech. Eng.*, vol. 8, no. 9, pp. 1–12, Sep. 2016.

[33] S. Kumra and C. Kanan, "Robotic grasp detection using deep convolutional neural networks," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 769–776.

[34] 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.

[35] 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, Oct. 2018.

[36] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Proc. Conf. Neural Informat. Process. Syst.*, 2015, pp. 91–99.

[37] U. Asif, J. Tang, and S. Harrer, "Ensemblenet: Improving Grasp Detection using an Ensemble of Convolutional Neural Networks," in *Proc. Brit. Mach. Vis. Conf.*, 2018, pp. 10–21.

[38] X. Yan, M. Khansari, J. Hsu, Y. Gong, Y. Bai, S. Pirk, and H. Lee, "Data-efficient learning for sim-to-real robotic grasping using deep point cloud prediction networks," 2019 *arXiv:*1906.08989.

[39] 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.

[40] 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.

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

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

[43] 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.

[44] 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.

[45] 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.

[46] 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.

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

[48] 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.

[49] 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.

[50] 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.

[51] 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.

[52] 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.

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

[54] 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.

[55] 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.

[56] 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.

[57] 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.

[58] 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.

[59] 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.

[60] 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.

[61] Z. Zhu, K. Lin, A. K. Jain, and J. Zhou, "Transfer learning in deep reinforcement learning: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 11, pp. 13344–13362, 2023.

[62] Z. Wang, G. Feng, L. Shen, and X. Zhang, "Cover selection for steganography using image similarity," *IEEE Trans. Dependable Secur. Comput.*, vol. 20, no. 3, pp. 2328–2340, 2023.

[63] 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.

[64] Y. Yang, H. Yu, X. Lou, Y. Liu, and C. Choi, "Attribute-based robotic grasping with data-efficient adaptation," *IEEE Trans. Robot.*, vol. 40, pp. 1566–1579, 2024.

[65] M.Gilles, K. Furmans, and R. Rayyes, "Metamvuc: Active learning for sample-efficient sim-to-real domain adaptation in robotic grasping," *IEEE Robot. Automat. Lett.*, vol. 10, pp. 3644-3651, 2025.

[66] H. Kuang, H. Liu, X. Lin, and R. Ji, "Defense against adversarial attacks using topology aligning adversarial training," *IEEE Trans. Inf. Forensics Secur.*, vol. 19, pp. 3659–3673, 2024.

Chenghao Li is currently pursuing the Ph.D. degree at Japan Advanced Institute of Science and Technology (JAIST). He served as a reviewer for IEEE Transactions on Automation Science and Engineering, Imaging and Vision Computing, and Computers & Graphics. His research interests include robot grasping, human-robot interaction, adversarial learning, and computer vision.

Ziyan Gao received the M.S and Ph.D. degrees in information science from Japan Advanced Institute of Science and Technology, Ishikawa, Japan, in 2019 and 2022, respectively. He is currently a post-doctoral researcher with Japan Advanced Institute of Science and Technology.His research interests include, object physical parameter estimation, non-prehensile manipulation, neural networks.

Nak Young Chong received the B.S., M.S., and Ph.D. degrees in mechanical engineering from Hanyang University, Seoul, Korea, in 1987, 1989, and 1994, respectively. From 1994 to 2003, he was with Daewoo Heavy Industries in Geoje, Korea, Korea Institute of Science and Technology in Seoul, Korea, and Mechanical Engineering Laboratory and National Institute of Advanced Industrial Science and Technology in Tsukuba, Japan. In 2003, he joined the faculty at the Japan Advanced Institute of Science and Technology, Ishikawa, Japan, where he currently is a professor of information science and served as a Councilor, Director of the Center for Intelligent Robotics, and Chair Professor of Intelligent Robotics Group. He was a Visiting Scholar with Northwestern University, Evanston, IL, USA, Georgia Institute of Technology, Atlanta, GA, USA, University of Genoa, Genoa, Italy, and Carnegie Mellon University, Pittsburgh, PA, USA, and serves/served as an Associate Faculty with the University of Nevada, Las Vegas, NV, USA, Kyung Hee University, Yongin, Korea, and Hanyang University, Ansan, Korea. Dr. Chong serves/served as Senior Editor for IEEE Robotics and Automation Letters, Intelligent Service Robotics, and International Journal of Advanced Robotic Systems, and Associate Editor for IEEE Transactions on Robotics. He served as Program (co)-chair for JCK Robotics 2009, ICAM2010, IEEE Ro-Man 2011/2013/2022, IEEE CASE 2012, URAI 2013/2014, DARS 2014, ICCAS 2016, and IEEE ARM 2019. He was a General (co)-chair of URAI 2017 and UR 2020. He also served as President of Korea Robotics Society, Co-chair for IEEE RAS Networked Robots TC, and Fujitsu Scientific System WG.