Monozone-centric Instance Grasping Policy in Large-scale Dense Clutter

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

***Abstract*—Despite the impressive performance of existing vision-guided robot grasping methods in dense clutter, their reliance on a fixed view often results in incomplete object geometry in the view boundary and limits grasping in more challenging large-scale dense clutter. Moreover, analyzing all objects during grasping can detract from the reasoning for specific objects. This work proposes the Monozone-centric Instance Grasping Policy (MCIGP) to solve these problems. Specifically, the first part is the Monozone View Alignment (MVA), wherein we design the dynamic monozone that can 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, we devise the Instance-specific Grasp Detection (ISGD) to predict and optimize grasp candidates for one specific object within the monozone, ensuring an in-depth analysis of this object. We performed over 8,000 real-world grasping experiments in different cluttered scenarios with 300 novel objects, demonstrating that MCIGP significantly outperforms seven competitive grasping methods. Notably, in a large-scale densely cluttered scene involving 100 different household goods, MCIGP pushed the grasp success rate to 84.9%. To the best of our knowledge, no previous work has demonstrated similar performance. The source code and all grasping videos are available here.**

***Index Terms*—Robot grasping, grasp detection, class-agnostic segmentation, large-scale dense clutter, deep learning.**

I. INTRODUCTION

VISION-guided grasping is a fundamental robotic capability with wide-ranging applications in warehousing, manufacturing, retail, and service industries. Traditional visual grasping approaches rely on three-dimensional (3D) object models to construct grasp databases, integrating geometric and physical metrics [1], [2] and employing stochastic sampling to handle uncertainty [3]. However, their dependence on known 3D models limits generalization to novel objects. To address this limitation, recent studies [4], [5] have introduced an alternative paradigm that leverages Deep Neural Networks (DNNs) [6]–[10] to train function approximators. These approximators predict grasp candidates directly from images, utilizing datasets comprising empirical grasp successes and failures, thereby enabling efficient generalization to previously unseen objects at substantially lower cost. Nevertheless, these methods are unstable in dense clutter scenarios because of the tight spatial relationship between adjacent objects, which can easily cause collision during grasping.

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; nakyoung@jaist.ac.jp).

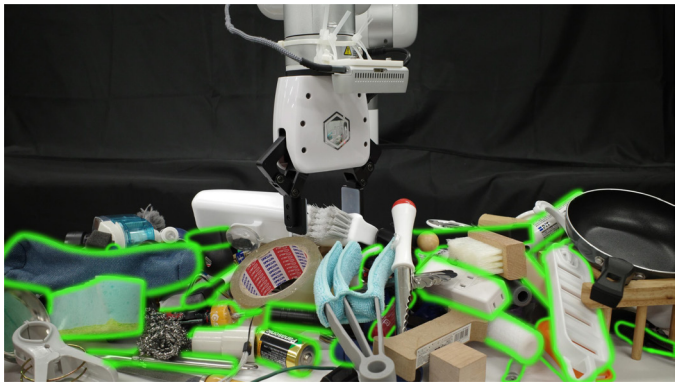

Fig. 1. Common grasping methods on the densely cluttered table: they require analyzing all objects in the scene (some of which are highlighted with a green border), which introduces significant computational redundancy and weakens the analysis of the object to be grasped. Additionally, their reliance on a fixed view often leads to incomplete object geometry at the view boundary.

One solution is to design novel grippers to replace commonly used parallel-jaw grippers, like jamming grippers [11], telescopic grippers [12], or hybrid grippers (combined with suction, parallel-jaw, and magnetic grippers) [13]. These methods can leverage the structural properties of the gripper to reduce the probability of collisions with surrounding objects 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 [14]–[18] perform instance-level grasp detection for all objects, which combines the class-agnostic segmentation model with the grasping model to filter out potential collisions on adjacent objects 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 (including all of the above-mentioned 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, as illustrated in Fig. 1.

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 be primarily employed to modify the input image—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. A summary of the contributions in this work is as follows:

1) 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.
2) 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.
3) 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.
4) We release our code and all grasping experiment videos to support reproducibility and encourage future research in large-scale dense clutter grasping.

This paper is organized into the following sections. Section II (Related Work) provides a review of traditional grasping methods and learning-based grasping methods. Section III (Grasp Configuration) describes the 4-DOF (Degree of Freedom) grasp configuration, and how to transform it from the image coordinates to the robot end effector coordinates. Section IV (Proposed Method) provides an overview of MCIGP and makes a detailed description of its two components (MVA and ISGD), as well as each submodule of each component. Section V (Experiments) first compares the real grasping performance between our method and seven competitive baseline grasping methods in different dense clutter scenarios, then validates the effectiveness of each component for our method by the ablation study, and analyzes failure cases based on these results. Finally, Section VI (Conclusion) summarizes the work of this paper and provides prospects for future research.

II. Related Work

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.

A. Traditional Grasping Methods

Traditional grasping methods rely on mathematical and physical models that describe the geometry, kinematics, and dynamics of objects [1]–[3]. These methods typically assume access to a detailed 3D model of the object being grasped, which is used to compute stable grasps. For example, [19] optimized grasp strategies by leveraging both a known 3D model of the object and predefined contact points for the robot gripper. Similarly, [20] 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.

B. Learning-based 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 [21], which contains around

1,000 RGB-D images and has been widely used to train grasping models, such as [22]–[28], based on convolutional neural networks (CNNs) [29]. However, this dataset is quite small and consists only of single-object images, which limits the dense clutter grasping capabilities.

The Dex-Net series [4], [30]–[33] 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 [5], [34], [35], 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, the above methods are 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 [14]–[18]. 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, thereby preventing them from being grasped in more challenging large-scale dense clutter scenarios.

Unlike the aforementioned works, we break the limitations of view boundaries by defining the dynamic monozone, within which grasp candidates for a specific object are predicted and optimized. This operation allows for a more comprehensive analysis of the target object and can realize grasping in large-scale dense clutter scenarios.

III. Grasp Configuration

Now, we elaborate on how the 4-DOF grasp configuration is represented in the image coordinate system and its conversion to the robot end effector coordinate system (eye-in-hand grasping). Specifically, we adopt the same grasp configuration in [36], which is composed of parameters (x, y, w, h, θ) forming a rotated box. Here, (x, y) represents the center of the box, w and h denote the width and height of the box, and θ represents the angle of the box relative to the horizontal direction.

Since h is only used for visual representation and not in the conversion process, we denote the grasp configurations (or one grasp candidate) in the image and robot end effector coordinate systems as g_i and g_r, respectively. g_i and g_r are composed by (x, y, w, θ) and $(x_r, y_r, z_r, w_r, \theta_r)$, respectively. Here (x_r, y_r, z_r) represents the grasp position in the robot end effector coordinate system, w_r is the opening stroke of the parallel jaw gripper, and θ_r is the rotation angle of the gripper relative to the Z axis.

The conversion between g_i and g_r can be divided into three parts. The first part involves converting (x, y), as shown in Eq. 1: 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 (x, y) from the image coordinate system to the camera coordinate system (x_c, y_c, z_c), denoted by p_c.

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

The first part is followed by converting p_c, *i.e.*, (x_c, y_c, z_c), to the robot end effector coordinate system (x_r, y_r, z_r) denoted by p_r via off-line hand-eye calibration as shown in Eq. 2, where the rotation and translation parts are denoted by $\mathbf{R}_c^r$ and $\mathbf{T}_c^r$ and $\mathbf{0}_{1\times3}$ represents a 1×3 zero matrix.

$$\begin{bmatrix} p_r \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_c^r & \mathbf{T}_c^r \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix} \begin{bmatrix} p_c \\ 1 \end{bmatrix} \tag{2}$$

The final part involves the conversion between the gripper stroke w_r and rotation θ_r relative to the grasp box's width w, and rotation θ, which can be manually adjusted because of their linear relationship.

$$\mathcal{P}_r = (x_r, y_r, z_r, \theta_r, \theta_x^*, \theta_y^*) \tag{3}$$

After a series of conversions, the final grasp pose $\mathcal{P}_r$ based on g_r can be obtained, as shown in Eq. 3, where θ_x^* and θ_y^* represent the constant rotations relative to the X-axis and the Y-axis. Finally, the gripper will be moved to the target pose using inverse kinematics, and its stroke will be kept to the width w_r to grasp the object.

IV. Proposed Method

We propose a novel grasping policy, the Monozone-centric Instance Grasping Policy (MCIGP), designed to realize grasping in large-scale dense clutter, as illustrated in Fig. 2. 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).

A. 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 to find the dynamic monozone that

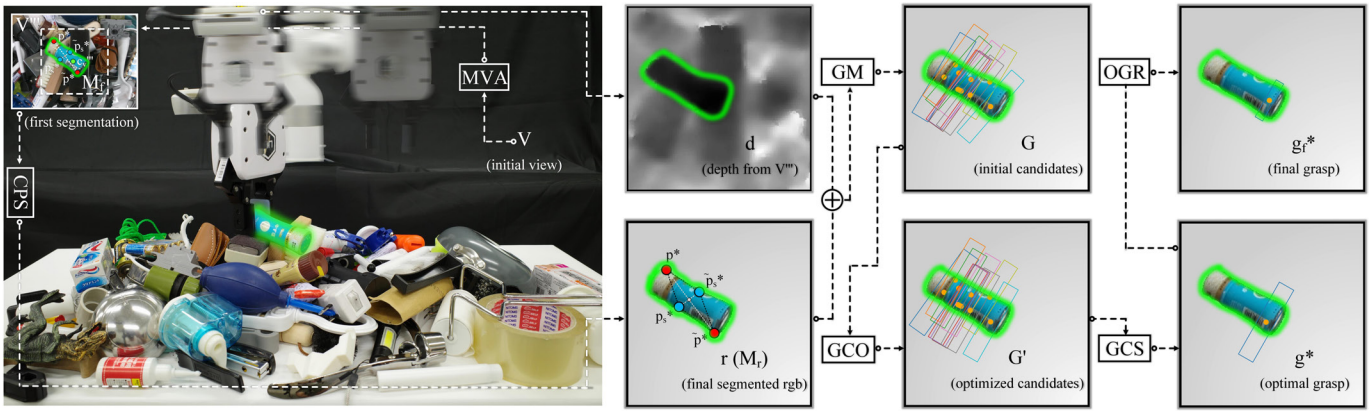

Fig. 2. Pipeline of MCIGP: Firstly, conducting Monozone View Alignment (MVA) to align the initial view $\mathcal{V}$ of depth camera on the target object to get view $\mathcal{V}'''$, and segment this object by the center c_v''' (green point) of this view as prompt to obtain initial segmented RGB image (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).

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. 4, 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 [11], which will also suffer from the problem of view boundary limitation.

$$\begin{bmatrix} p_{r_d} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_c^r & \mathbf{0}_{3\times1} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix} \begin{bmatrix} p_{c_d} \\ 1 \end{bmatrix} \tag{4}$$

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. 4. 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, where (H, W) is the size of the monozone.

$$(x_q^*, y_q^*) = \underset{(x_q, y_q) \in (H,W)}{\arg\max}\, \mathcal{Q}(x_q, y_q) \tag{5}$$

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 $p_{r_q}^*$ in the same way 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. 4. The robot movement ranges the same as in D-MVA and is restricted within the 224×224 monozone.

B. 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) [37] 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 [38] 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. 6, $\mathcal{P}_e$ means the set of all pixels on the edges, and $\|\cdot\|_2$ means the Euclidean distance.

$$(p^*, \tilde{p}^*) = \underset{(p_i, p_j) \in \mathcal{P}_e}{\arg\max} \|p_i - p_j\|_2 \tag{6}$$

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. 7. 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 $\mathcal{M}_r$.

$$(p_s^*, \tilde{p}_s^*) = \perp_{(p^*, \tilde{p}^*)} \cap \mathcal{P}_e \tag{7}$$

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 [25] 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. 8 and use it to formulate the new grasp candidate.

$$\begin{aligned} \mathcal{R}^* = \underset{\mathcal{R}}{\arg\min} \left(\left| \theta'(\mathcal{R}) - \frac{\pi}{2} \right| + \left| \theta''(\mathcal{R}) - \frac{\pi}{2} \right| \right) \\ \text{s.t. } \mathcal{R} \in \{0^\circ, 2^\circ, 4^\circ, \ldots, 2\pi\} \end{aligned} \tag{8}$$

The second part of GCO is designed to ensure that viable grasp candidates remain available after sampling. It achieves

Algorithm 1: MCIGP

Input: Camera Frame $\mathbb{V}$
Output: Final grasp set $\mathbb{G}_f$ for all objects

foreach $\mathcal{V} \in \mathbb{V}$ do
  // Conducting Depth-based MVA
  $\mathcal{V}' \leftarrow \mathcal{V}$, $\mathcal{V}'' \leftarrow \mathcal{V}'$, $\mathcal{V}''' \leftarrow \mathcal{V}''$
  $\mathcal{M}_f \leftarrow \text{SAM}(c_v''')$
  $(p^*, \tilde{p}^*) \leftarrow$ SOLVE Eq. 6
  $(p_s^*, \tilde{p}_s^*) \leftarrow$ SOLVE Eq. 7
  // Running CPS
  $\mathcal{M}_s \leftarrow \text{SAM}(p^*, \tilde{p}^*, p_s^*, \tilde{p}_s^*)$
  $\mathcal{M}_d \leftarrow \text{DILATION}(\mathcal{M}_s, \mathtt{d})$
  $\mathcal{M}_r \leftarrow \mathcal{M}_s, \mathcal{M}_d$
  $\mathbb{G} \leftarrow \text{GRASPING MODEL}(\mathcal{M}_r, \mathtt{r})$
  // Executing first part of GCO
  foreach $g_i \in \mathbb{G}$ do
    $\mathcal{R}^* \leftarrow$ SOLVE Eq. 8
    $\mathbb{G}' \leftarrow \mathbb{G}, \mathcal{R}^*$
  // Executing second part of GCO
  if $\mathbb{G}' = \emptyset$ then
    SOLVE Eq. 9
    break
  // Sampling grasp candidate
  else
    foreach $g_i \in \mathbb{G}'$ do
      $\mathbb{G}'' \leftarrow$ SOLVE Eq. 10
    foreach $g_i \in \mathbb{G}''$ do
      $g^* \leftarrow$ SOLVE Eq. 11
    // Refining optimal grasp
    $R_{ec} \leftarrow g^*, \mathcal{M}_r$
    $(w_s', c_i') \leftarrow \text{MIN}\mathcal{R}_{ec}(w), e_c, \mathcal{R}_{ec}(c)$
    $g_f^* \leftarrow g^*, (w_s', c_i')$
  $\mathbb{G}_f \leftarrow g_f^*$
return $\mathbb{G}_f$

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. 9.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos(\theta_c) & -\sin(\theta_c) \\ \sin(\theta_c) & \cos(\theta_c) \end{bmatrix} \begin{bmatrix} x' - x_{c_r} \\ y' - y_{c_r} \end{bmatrix} + \begin{bmatrix} x_{c_r} \\ y_{c_r} \end{bmatrix} \tag{9}$$

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. 10. 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.

$$\mathbb{G}'' = \{g_i \in \mathbb{G}' \mid \forall p \in \mathcal{P}_s, \, |\mathtt{d}(p) - \mathtt{d}(c_i)| \leq \mathcal{T}_d\} \tag{10}$$

After getting $\mathbb{G}''$, we use our previous method [39] 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. 11.

$$g^* = \arg\min_{g_i \in \mathbb{G}''} \mathtt{d}(c_i) \tag{11}$$

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 [33], 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^*. We show a pseudocode of MCIGP in Algorithm 1.

Fig. 3. Objects for the grasping experiment: toys, ragdolls, household goods, and snacks (clockwise from top left).

V. 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.

A. Experimental Settings

1) Setting for Grasping Model: The baseline methods are categorized into two groups. The first group includes GGCNN [22], GGCNN2 [23], GRconvnet [25], SEnet [24], and FCGnet [26], which are suitable for mid-clutter scenarios. The second group comprises DexNet 4.0 [4] and GraspNet [5], which are tailored for high-clutter scenarios.

For the first group, since the pre-trained models were all trained on the Cornell Grasping Dataset [21] (only one object in each fixed white background), their performance in cluttered environments is limited. Therefore, we merge the OCID Grasping Dataset [14], [41] (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 [36] 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°.

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) [5], [42], and the 6-DOF version of GraspNet (GraspNet 6D) [5]. 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 [43]. 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 100 objects) grasping experiments. Therefore, we refer to the objects used in two widely recognized dense clutter grasping methods, DexNet 4.0 [4] and GraspNet [35]. 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. 3. 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 × 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 [22] 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.

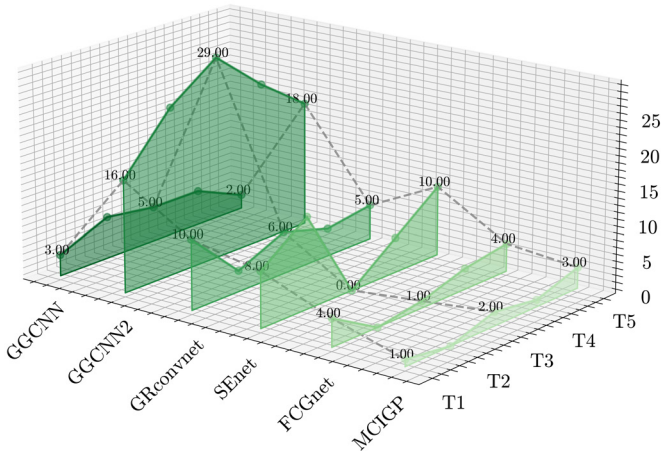

Fig. 4. Line graph showing GSR of MCIGP and first-group baselines in mid-clutter scenarios. The horizontal axis represents different methods, and the depth axis represents trials from T1 to T5. The vertical axis represents the number of grasp failures. We emphasize the number of grasp failures (T1, T3, T5) in each method with dots, and connect them with dashed lines to better show the difference.

TABLE I
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**

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.

B. 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 I. 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. 4 to better show the gap between MCIGP with other baseline methods. As shown in this figure, it is very clear that our 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. Specif-

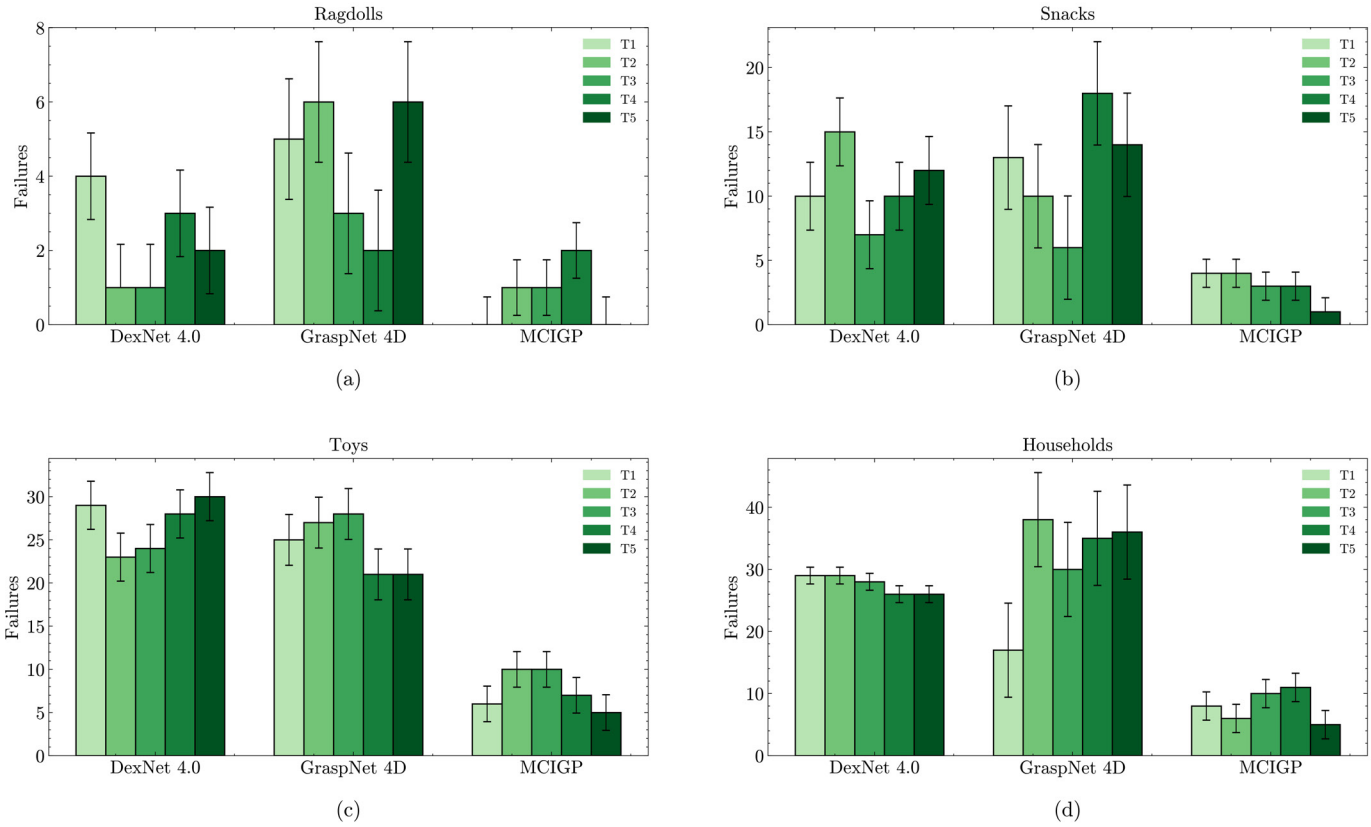

Fig. 5. Bar graphs showing GSR of MCIGP and second-group baselines in high-clutter scenarios. (a), (b), (c), and (d) represent the results of testing ragdolls, snacks, toys, and household goods. In each subfigure, the vertical axis represents the number of grasp failures, and the horizontal axis represents 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.

TABLE II
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**

TABLE III
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**

ically, 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 II, 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. It is obvious that the bar length of our method is 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 III, despite MCIGP supporting only 4-DOF grasping (GSR is 86.2% (250/290)), it still outperforms GraspNet 6D (GSR is 81.2% (250/308)).

C. Ablation Studies

1) Effectiveness of Monozone View Alignment: To demonstrate the effectiveness of MVA, we first evaluate it in a

Fig. 6. 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.

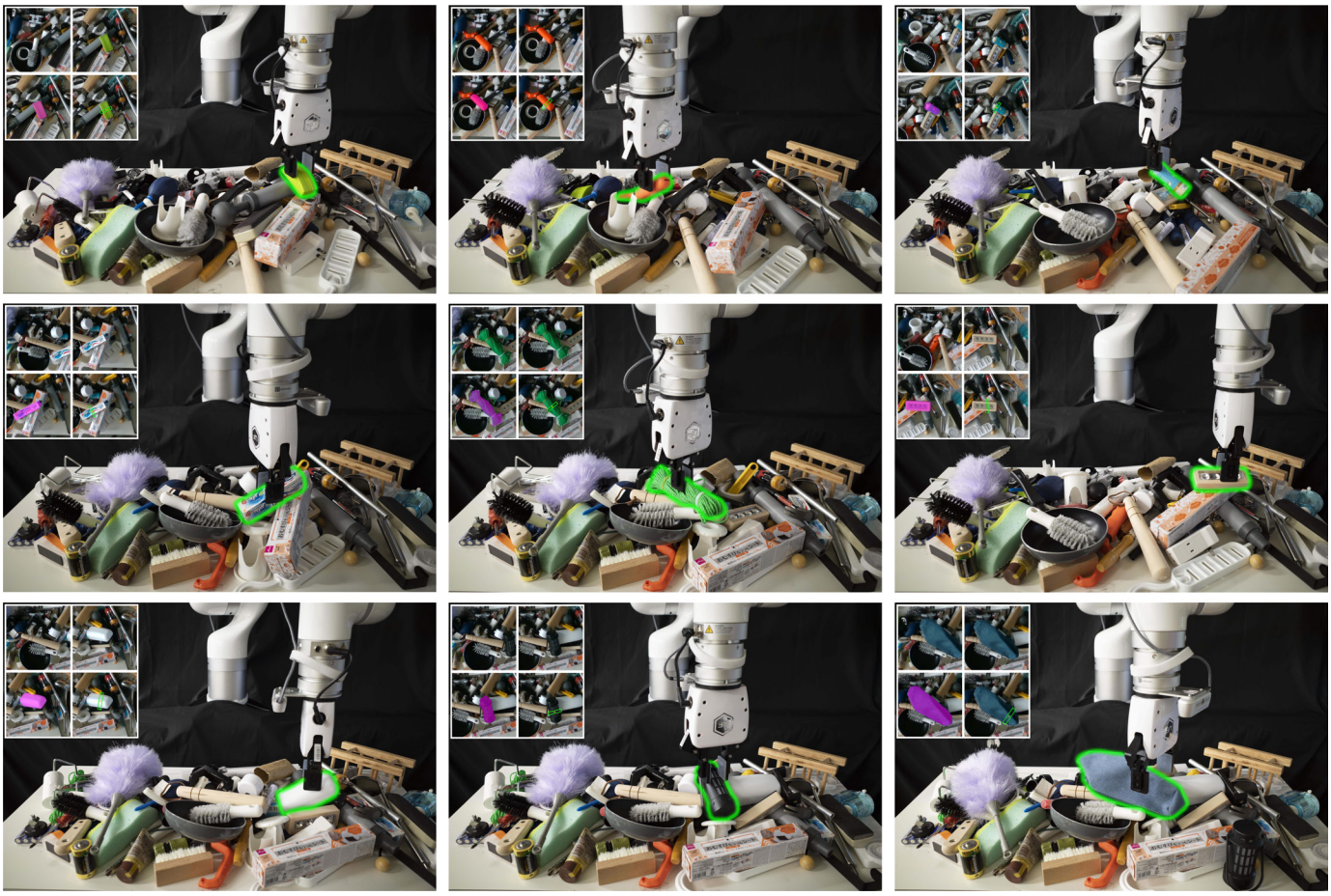

Fig. 7. Visualization of the grasping process on large-scale clutter scenarios with 100 household goods for MCIGP. Each subfigure represents the grasping process, and we emphasize the object being grasped by the green border. Sub-subfigures inside each subfigure are the original view (top left), aligned 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.

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 IV. 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 by overcoming boundary limitations. In addition,

Table IV 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 baseline methods in order to achieve higher grasp success rates. We also visualize the view alignment and grasping process in Fig. 7. Note that the visualization here is mainly based on Section V-C2.

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 V. 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 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 VI; 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. 6 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 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 VII, 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 the grasping processes in Fig. 7.

TABLE IV
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**

TABLE V
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 VI
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 VII
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**

D. Failure Case Analysis

In the above experiments, we performed more than 8,000 grasp attempts and achieved a total of 6,350 successful grasps. More importantly, we tested MCIGP's capability in large-scale clutter scenarios involving 100 novel objects, and the GSR is stable between 85% to 89%. To the best of our knowledge, no previous work has demonstrated similar performance. However, MCIGP still encounters some failures. The first issue is 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. Furthermore, 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. 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 [44] to predict the occluded parts of the object, thereby getting the complete mask of the object. Finally, when objects with similar depths are tightly packed together, they will be difficult to grasp. This challenge can be overcome by designing a methodology that can combine grasping and pushing manipulation, like [45].

VI. CONCLUSION

In this paper, we proposed the MCIGP, which first employed the MVA to find the dynamic monozone and then leveraged ISGD to predict and optimize grasp candidates for one specific object within this monozone. MCIGP enabled the robot to effectively mitigate view boundary limitation and realize in-depth analyses for one specific object in large-scale dense clutter environments, significantly enhancing the grasping success rate. We conducted over 8,000 real-world grasping attempts on 300 novel objects across various levels of clutter, including mid-clutter (20 objects), high-clutter (50 objects), and large-scale clutter (100 objects), demonstrating that MCIGP significantly outperforms seven competitive grasping methods.

Future work can be divided into two major parts. The first part can focus on addressing the issues highlighted in the

Failure Case Analysis to enhance the method proposed in this paper. The second part can involve using this method as a baseline and extending it to human-robot interaction for specific object retrieval. For instance, safely grasping a specific object in a cluttered scene without interfering with other objects, and securely handing it over to a person should be noteworthy.

REFERENCES

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

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

[3] 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.

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

[5] 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.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Conf. Neural Informat. Process. Syst.*, 2017, pp. 6000–6010.

[7] S. Hochreiter, and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[8] T. Brown, b. Mann, N. Ryder, M. Subbiah, J.D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, and S. Agarwal, "Language models are few-shot learners," in *Proc. Conf. Neural Informat. Process. Syst.*, 2020, pp. 1877–1901.

[9] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, and J. Uszkoreit, "An image is worth 16x16 words: Transformers for image recognition at scale," 2020, *arXiv:*2010.11929.

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 580-587.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

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

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

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

[38] 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.

[39] 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.

[40] 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.

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

[42] 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.

[43] 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.

[44] J. Zhang, Y. Gu, J. Gao, H. Lin, Q. Sun, X. Sun, X. Xue, and Y. Fu, "LAC-Net: Linear-Fusion Attention-Guided Convolutional Network for Accurate Robotic Grasping Under the Occlusion." in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2024, pp. 10059-10065.

[45] A. Zeng, S. Song, S. Welker, J. Lee, A. Rodriguez, and T. Funkhouser, "Learning synergies between pushing and grasping with self-supervised deep reinforcement learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 4238–4245.