Engineering Transactions, **74**(3): 477–498, 2026, https://doi.org/10.24423/engtrans.2026.3644
Institute of Fundamental Technological Research Polish Academy of Sciences (IPPT PAN)
Université de Lorraine • Poznan University of Technology

Research Paper

Optimization of Dual-Constraint Robotic Arm Grasping Operations Based on an Optimized Contact Grasping Network

Feifei Zhao

School of Intelligent Manufacturing, Zibo Vocational Institute
Zibo, China

e-mail: zhaoffzhaoff@163.com

With the continuous advancement of intelligent manufacturing and robot perception capabilities, traditional robotic arm grasping methods still face problems such as inaccurate pose prediction and poor task adaptability when operating in dynamic scenes, handling complex objects, and executing functional action. Therefore, this research develops an optimized contact grasping network model that integrates scene and task constraints. It combines a UR5 six-degree-of-freedom robotic arm, visual input, and the Contact-GraspNet architecture based on point cloud, and introduces a PointNet++ local feature enhancement mechanism and a lightweight encoder design to effectively improve the spatial perception and action planning capabilities of grasping points. Experimental results show that, on the GraspNet and YCB datasets, the model achieves $F1$-scores of 92.54 % and 91.82 %, respectively, with the average execution time reduced to 0.61 s. In functional operation scenarios involving door handles, kettle handles, and drawer pulls, the grasping accuracy remained above 0.89, with the task completion rates significantly outperforming those of mainstream baseline models. Under visual interference conditions with an occlusion rate of up to 75 %, the average inference latency was controlled within 0.82 s. Under varying light intensities, the pose angle error remained within the range of 1.21° to 1.87°. Therefore, this model exhibits comprehensive advantages in grasping accuracy, latency control, and deployment efficiency, and has the potential to be largely applied in industrial, service, and specialized task environments.

Keywords: robotic arm grasping, scene constraints, task constraints, Contact-GraspNet, point cloud.

Copyright © 2026 The Author(s).
Published by IPPT PAN. This work is licensed under the Creative Commons Attribution License CC BY 4.0 (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In the fields of automated manufacturing, service robots, and intelligent warehousing, robotic arm grasping systems, as a key component in human-computer interaction and object manipulation, are gradually developing toward high precision, high robustness, and adaptive capabilities. In practical applica-

tion scenarios, robotic arm grasping is not only limited by the physical environment but also affected by dual constraints of task objectives [1, 2]. Traditional contact grasping networks fail to consider these constraints simultaneously, often resulting in issues such as mis-grasping, collisions, or grasping failures in complex environments. Therefore, how to introduce task-level and scene-level dual constraints while retaining the efficient feature learning ability of deep grasping networks is the core problem that urgently needs to be solved.

1.1. Research progress and limitations of traditional grasping and visual recognition methods

Early robotic arm grasping primarily relied on methods based on visual detection, control strategies, and trajectory planning. For instance, Chiu *et al.* [3] employed channel pruning to enhance You Only Look Once version 5 (YOLOv5), integrating it with positioning algorithms and robotic arm motion planning to achieve grasping. Their approach reduced the number of parameters while improving grasping accuracy. Geng [4] enhanced robotic arm grasping stability and robustness through visual servoing and fractional-order control approaches. Concurrently, Yuan [5] improved adaptability to diverse objects by integrating multimodal information fusion and refining world models. However, such methods generally rely on rigid geometric assumptions or explicit scene configurations, exhibiting limited adaptability to complex conditions such as occlusion, deformation, and multi-object interference. Furthermore, they prioritize the executability of grasping actions while paying less attention to whether these actions fulfill functional task objectives, making it difficult to support functional operations such as opening doors, lifting objects, or rotating knobs.

1.2. Deep learning-based point cloud grasping and contact semantic inference research

With the advancement of deep learning, point cloud-based grasping recognition methods have emerged as a mainstream research direction. Xu *et al.* [6] proposed a grasping pose estimation method integrating multi-scale residual networks with RGB sensors, maintaining high recognition accuracy even in low-light environments. Grasping strategies represented by Contact-GraspNet (CG) enhance generalization capabilities for complex scenarios by inferring contact points, grasp scores, and grasp poses from point clouds [7]. Related extension studies include: Yan *et al.* [8] enhanced RGB-D feature extraction by incorporating attention and residual modules, Gilles *et al.* [9] achieved efficient grasping of unknown objects by integrating MetaGraspNetV2, and Hoang [10] improved point cloud denoising and the grasp position inference stability through attention mechanisms and the VoteGrasp strategy. While these approaches strengthen

the end-to-end mapping from visual information to grasp semantics, two core issues persist:

1. Networks remain vulnerable to point cloud density variations and noise, leading to unstable geometric feature extraction.
2. Models primarily focus on the geometric feasibility of grasp actions while lacking task-related functional constraints, making them unable to determine whether grasp poses match specific operational requirements.

1.3. Developments in functional grasping, task constraints, and pose estimation research

In task-relevant grasping research, scholars have gradually expanded their focus from 'whether an object can be grasped' to 'whether the task action can be correctly executed.' For example, Yu *et al.* [11] introduced SCNet, which includes a self-supervised mechanism for category-level pose estimation to enhance the model's transferability from simulation to reality. Other studies have focused on action planning for deformable objects or functional scenarios (e.g., knobs, handles). However, most approaches still lack systematic modeling of task constraints and rarely address critical factors such as motion direction consistency, rotational axis constraints, or grasping torque requirements. Furthermore, existing grasping strategies typically fail to integrate pose alignment mechanisms for coarse and fine registration, leading to amplified pose errors during functional component manipulation. This significantly impacts grasping success rates and operational stability.

In summary, although existing robotic arm grasping methods have made some progress in pose estimation and point cloud-based grasping pose inference, they still suffer from performance degradation and insufficient robustness in handling geometric interference, adapting to functional operation constraints, and ensuring contact grasping stability in complex scenarios. Therefore, this research develops a robotic arm contact grasping strategy model that fuses an optimized Contact-GraspNet architecture with a dual-constraint mechanism. This model improves point cloud representation efficiency by introducing local feature extraction and channel lightweight pruning strategies based on PointNet++, while introducing action mechanics models under task constraints to achieve task-oriented grasping pose discrimination. On this basis, this research integrates coarse and fine registration algorithms to effectively optimize pose estimation accuracy and operational consistency. The innovation lies in constructing an end-to-end grasping control system that integrates perception, structural optimization, and functional feedback, providing a high-precision, high-efficiency, and engineering-scalable technical solution for intelligent robotic arm operations under multiple constraint conditions.

2. Methods and materials

To avoid confusion, the study first introduces the standard Contact-GraspNet architecture to illustrate its fundamental workflow and input-output format. Subsequently, several proposed improvements are presented, including:

- a point cloud uniform downsampling strategy,
- a local geometric feature enhancement mechanism based on PointNet++,
- a lightweight design for encoder channel pruning,
- a task-constraint-coupled grasp candidate filtering strategy,
- a two-stage pose optimization based on the point pair feature (PPF) algorithm and the iterative closest point (ICP) algorithm.

2.1. Construction of a robotic arm grasping network model considering scene constraints

A UR5 six-degree-of-freedom robotic arm is selected as the operating platform in this study. The arm has a compact structure and high repeatability, with a positioning accuracy of up to ±0.1 mm. It is widely used in manipulation and grasping of complex objects [12–14]. The study first models grasping operations within a 2D plane, assuming the target object rests stationary on a horizontal workbench. The schematic diagram of the 2D grasp and the grasp pose diagram are shown in Fig. 1 [15, 16]. It should be noted that the introduction of the 2D grasping schematic (Fig. 1) during the modeling phase is not intended to construct a 2D grasping model. Rather, it serves to visually illustrate the geometric relationship between the robotic end-effector and the target object, including the grasp base point, gripper width, formation of the contact triangle, and definition of the pose vector. This simplified schematic serves solely to explain the 3D symbols, parameters, and grasp semantics, and does not involve subsequent algorithmic derivation or experimental processes.

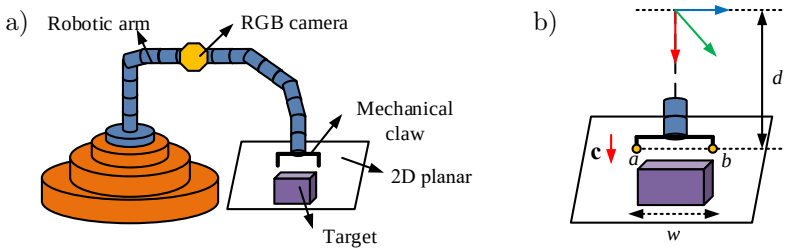

Fig. 1. Schematic diagram of 2D planar grasping (a) and grasp pose (b) representation of the robotic arm.

Figure 1 presents a 2D schematic of 3D grasping parameters, serving solely to illustrate the grasp pose definition method. Both model training and infer-

ence are implemented based on 3D point clouds. In Fig. 1a, the target object is stationary on a horizontal workbench, and the RGB-D camera is fixedly installed to obtain image data containing depth information. The robotic arm infers and executes the grasp point position and pose based on visual input. As shown in Fig. 1b, the gripper establishes contact with the surface of the object through the points a and b on both sides, forming a stable supporting contact triangle. The parameter w represents the width of the gripper, d signifies the Euclidean distance from the gripper center to the reference coordinate origin. The grasping direction of the gripper is represented by the vector $\mathbf{c}$. To efficiently predict grasp point positions and poses, the study introduces Contact-GraspNet as the basic network model. This network is an end-to-end grasp detection framework based on 3D point clouds, which can directly predict feasible grasp poses of each point from point cloud data generated by RGB-D images. It demonstrates excellent scene adaptation ability and contact semantic reasoning capability [17, 18]. The training process is shown in Fig. 2.

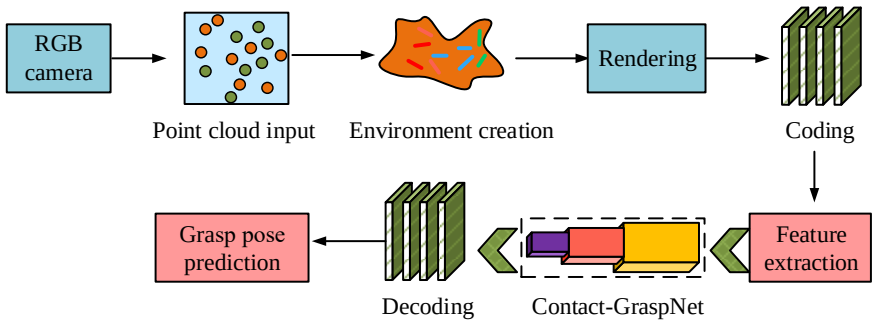

FIG. 2. Training flow of Contact-GraspNet.

In Fig. 2, the training of Contact-GraspNet mainly consists of four stages: encoding, feature extraction, grasp pose prediction, and loss feedback from the point cloud input $P \in R^{N\times 3}$ generated by the simulator. In terms of the workflow, we first explore using an RGB-D camera to capture depth images within the grasping scene. By integrating this depth information, we generate a 3D point cloud as the input for Contact-GraspNet. The input point cloud is voxelized into a 3D voxel grid $V \in R^{D\times H\times W\times C}$, and its calculation is as follows:

$$V(x,y,z) = \sum_{i=1}^{N} \left[\left[\frac{p_i - o}{r} \right] - (x,y,z) \right] \cdot f_i, \tag{2.1}$$

where p_i represents the i-th coordinate point, o signifies the voxel grid origin, r represents the voxel resolution, and f_i signifies the local feature of the i-th point. Secondly, the network predicts the grasp transformation $\widehat{T}_j \in \mathrm{SE}$ for each

voxel center point v_j, usually represented by the translation vector $\mathbf{t}_j$ and the axial rotation vector $\boldsymbol{\omega}_j$, as follows:

$$\widehat{T}_j = \begin{bmatrix} \exp([\mathbf{w}_j]_\times) & \mathbf{t}_j \\ 0 & 1 \end{bmatrix}, \tag{2.2}$$

where $\exp([\mathbf{w}_j]_\times) \in SO(3)$ represents the mapping from the Lie algebra to the Lie group, and $[\mathbf{w}_j]_\times$ represents the antisymmetric matrix form of vector $\mathbf{w}_j$. Then, Contact-GraspNet assigns a grasp score $\widehat{s}_j \in [0,1]$ to each candidate grasp pose g_j during training. The regression uses binary cross-entropy calculated as

$$L_{\text{score}} = -\frac{1}{M}\sum_{j=1}^{M}(s_j \log \widehat{s}_j + (1-s_j)\log(1-\widehat{s}_j)), \tag{2.3}$$

where L_{score} represents the grasp score loss, $s_j \in [0,1]$ represents the ground-truth label determining whether it is feasible to grasp, $\widehat{s}_j$ represents the retrievable probability predicted by the network, and M represents the number of candidate samples to be grasped. The total loss function is the sum of the grasp score loss and pose regression loss:

$$L_{\text{total}} = L_{\text{score}} + \lambda_{\text{pose}} \cdot L_{\text{pose}}, \tag{2.4}$$

where L_{pose} represents the pose regression loss, including the point-to-point translation error and the rotation angle error, λ_{pose} represents the weight coefficient of the pose loss, and L_{total} represents the total loss function. Although Contact-GraspNet has achieved good results in point cloud-based grasp prediction, its original model lacks a unified standard for inputting point cloud data, resulting in insufficient learning stability of target grasp features when object density changes significantly. Therefore, a point cloud downsampling optimization based on distance uniformity is introduced:

$$P' = \text{FPS}(P, N), N = 1024, \tag{2.5}$$

where P represents the original input point cloud, P' represents the sampled point cloud input, and FPS(_) represents the farthest point sampling function. PointNet++ local feature encoding is used to replace single-layer global perception:

$$f_i^* = \text{MLP}\left(\bigoplus_{j \in N(p_i)} \phi(p_j - p_i)\right), \tag{2.6}$$

where $N(p_i)$ represents the spherical neighborhood centered at p_i, $\phi(_)$ represents the geometric encoding function, and $\oplus$ represents the feature concatenation operation. MLP represents multilayer perceptron. Finally, the overall

channels and layers are pruned to achieve a lightweight architecture, and the calculation expression is

$$R^{1024\times3} \xrightarrow{\text{Enc}} R^{512\times128} \xrightarrow{\text{Dec}} R^{1024\times64}, \tag{2.7}$$

where Enc represents the encoder, Dec represents the decoder, 1024 represents the number of points, 128/64 represents the number of feature channels. The improved Contact-GraspNet structure is shown in Fig. 3.

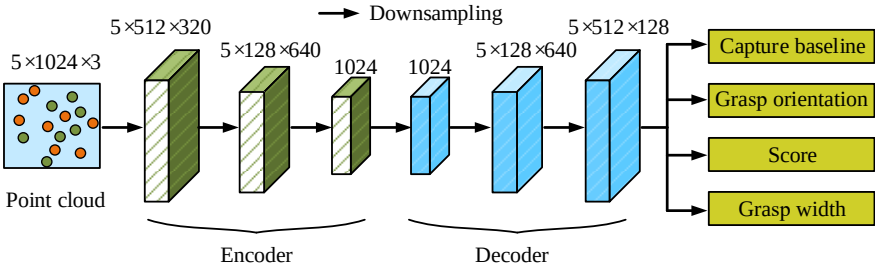

Fig. 3. Improved Contact-GraspNet structure.

In Fig. 3, the input point cloud is first unified into a batch data of size $5 \times 1024 \times 3$ and uniformly sampled using farthest-point sampling. The encoder consists of three convolutional layers, producing feature maps of sizes $5 \times 512 \times 320$, $5 \times 128 \times 640$, and $5 \times 512 \times 128$, respectively. Each layer incorporates channel compression to reduce redundant information while introducing a local feature enhancement mechanism based on PointNet++ to extract key local geometric structures in the third layer. The decoder progressively restores these features to a spatial dimension matching the input point count, producing an output of size $5 \times 1024 \times 128$. Based on this, the network predicts four parallel outputs: grasp base point coordinates, grasp pose quaternion, grasp score, and grasp width.

2.2. Construction of a robotic arm contact grasping network strategy model integrating scene constraints and task constraints

After constructing a robotic arm grasping network model that considers scene constraints, to further improve task adaptability and operational feasibility of grasping actions, a task constraint mechanism is introduced to recognize the geometric feasibility of grasp positions, and comprehensively evaluates whether grasping actions meet the requirements of specific operational tasks. The so-called task constraints refer to additional restrictions on target function, operation sequence, motion direction, and torque control, added to the grasping behavior. Taking a typical door handle grasping task as an example, the motion mechanics model and motion filtering operation diagram are shown in Fig. 4.

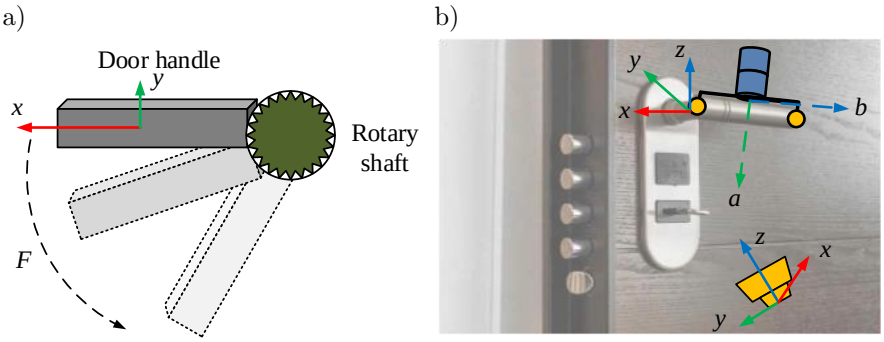

Fig. 4. Physical model of door handle motion (a) and operation schematic diagram of motion filtering (b) (picture source: https://colorhub.me/photos/JP4m1).

As shown in Fig. 4, the rotational torque T of the door handle is determined by the normal force F applied by the gripper at the contact point and the distance R from that point to the rotation axis, satisfying formula $T = F \cdot R$. Taking the local coordinate system of the door handle as a reference, the angular relationship between each candidate grasp pose and the rotation axis, and the consistency of the applied force direction, are calculated. If the contact direction of a certain grasp point deviates too much from the expected force direction, it will be filtered out by the task constraint module, and only grasp candidates that meet both contact stability and effective task execution will be retained. To achieve higher accuracy in grasp execution while satisfying task constraints, this study further adopts a pose estimation optimization method on the basis of the PPF algorithm, which accurately perceives and models the actual pose in the pre-grasping stage. Subsequently, based on coarse registration, the ICP algorithm is introduced to finely align the model point cloud with the target point cloud, further optimizing the rigid body transformation parameters to minimize the matching error. The overall process is shown in Fig. 5 [19, 20].

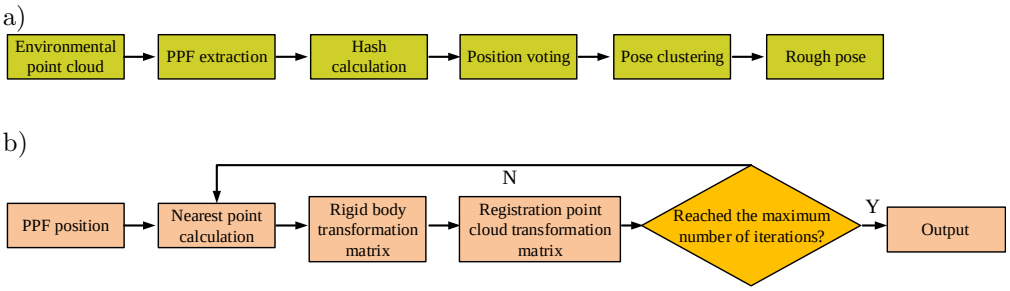

Fig. 5. PPF and ICP-based pose estimation processes: a) optimization process for pose estimation using the PPF algorithm, b) fine pose alignment process using the ICP algorithm.

As shown in Fig. 5, the PPF method first calculates feature relationships based on the distance, normal angle, and direction information of point pairs. It then generates locally invariant point pair features based on these relationships. The formal expression of the features is

$$\Phi_{ab} = (\|x_a - x_b\|, \angle(\mathbf{n}_a, d_{ab}), \angle(\mathbf{n}_b, d_{ab}), \angle(\mathbf{n}_a, \mathbf{n}_b)), \tag{2.8}$$

where x_a and x_b both represent the position coordinates of the two points in a point pair, $\mathbf{n}_a$ and $\mathbf{n}_b$ denote the unit normal vectors corresponding to two point pairs, and d_{ab} represents the point pair direction. After establishing an index for the feature quadruples in the hash table, PPF calculates candidate transformations one by one and performs pose voting based on the feature set of point pairs extracted from the query point cloud. The transformation score function is

$$S(H_k) = \sum_{\Phi \in F} \delta(\|\Phi - H_k(\Phi_m)\| < \varepsilon), \tag{2.9}$$

where H_k denotes the k-th candidate rigid body transformation, Φ_m is the model features, $\Phi \in F$ represents the feature set extracted from the scene point cloud, δ is the matching count function within the threshold, and ε represents the matching tolerance coefficient. The ICP algorithm aims to minimize the Euclidean distance between the target point cloud and the transformed model point cloud. A registration error function is constructed as

$$\tau(T) = \sum_{i=1}^{N} \|y_i - T \cdot z_i\|^2, \tag{2.10}$$

where $T \in SE(3)$ represents the rigid body transformation matrix to be optimized, z_i denotes the point in the model point cloud, y_i is the nearest neighbor point in the corresponding observation point cloud. When the registration error is below the set threshold or the transformation converges, ICP stops iterating and outputs the final fine alignment transformation as

$$T^* = \arg\min_{R,t} \sum_{i=1}^{N} \|y_i - (R \cdot z_i + t)\|^2, \tag{2.11}$$

where T^* represents the final rigid-body transformation matrix. At this point, the optimal solution is obtained by solving the covariance matrix through singular value decomposition, ensuring that the rigid-body transformation satisfies the minimum mean squared error criterion, thereby achieving high-precision fitting of the target point cloud pose. The pseudocode for the PPF and ICP algorithms is as follows:

Input:

$P_{\text{scene}} \leftarrow$ scene point cloud
$P_{\text{model}} \leftarrow$ model point cloud
$T_{\text{init}} \leftarrow$ initial pose estimate (optional)

Output:

$T_{\text{final}} \leftarrow$ final refined pose

Stage 1: Coarse Pose Estimation via Point Pair Features (PPF)

Sample keypoints from P_{model} to obtain $M = \{m_k\}$
Sample keypoints from P_{scene} to obtain $S = \{s_l\}$

Build PPF hash table:
    for each pair (m_k, m_j) in M do
        $f_{\text{model}} \leftarrow \text{PPF}(m_k, m_j)$
        insert f_{model} into hash table H

Estimate coarse pose by feature matching:
    for each pair $(s_l,\ s_i)$ in S do
        $f_{\text{scene}} \leftarrow \text{PPF}(s_l, s_i)$
        retrieve matching entries from H
        accumulate votes for each candidate transform T_c
    $T_{\text{PPF}} \leftarrow$ candidate transform with the highest vote count

if T_{init} exists then
    $T_{\text{coarse}} \leftarrow T_{\text{init}} \circ T_{\text{PPF}}$
else
    $T_{\text{coarse}} \leftarrow T_{\text{PPF}}$

Stage 2: Fine Pose Refinement via ICP

Set max iterations K and convergence threshold ε
$E_{\text{prev}} \leftarrow +\infty$

for iter = 1 to K do
    $P'_{\text{model}} \leftarrow T_{\text{coarse}}(P_{\text{model}})$
    $C \leftarrow$ nearest-neighbor correspondences between P'_{model} and P_{scene}
    $E \leftarrow \Sigma\|p' - q\|^2$ for (p', q) in C
    if $|E - E_{\text{prev}}| < \varepsilon$ then
        break
    $\Delta T \leftarrow$ optimal rigid transform estimated by SVD over C
        $T_{\text{coarse}} \leftarrow \Delta T \circ T_{\text{coarse}}$
        $E_{\text{prev}} \leftarrow E$

$T_{\text{final}} \leftarrow T_{\text{coarse}}$

Additionally, scene constraints refer to restrictions in 3D space that determine whether a candidate grasp pose $\widehat{g}_j$ satisfies environmental geometry, object surface accessibility, and collision feasibility. Scene constraints are expressed as

$$C_{\text{scene}}(\widehat{g}_j) = \begin{cases} 1 & \text{if } d(\widehat{g}_j, O) > \varepsilon, n_j \cdot s_j > 0, \\ 0 & \text{other}, \end{cases} \tag{2.12}$$

where O denotes the set of obstacles in the scene, $d(\widehat{g}_j, O)$ represents the minimum distance between the grasp pose and the nearest obstacle, ε denotes the safety margin threshold, n_j is the surface normal of the grasping surface, s_j denotes the gripper closing direction, and $n_j \cdot s_j$ represents the dot product of the two vectors. A value greater than 0 indicates that the angle between the closing direction of the gripper and the normal direction of the surface is less than 90°, thus meeting the geometric conditions for stable contact. Task constraints determine whether the grasp pose satisfies functional operation requirements (e.g., rotating door handles, pulling drawers, pouring kettles). These constraints ensure that the grasp pose aligns with the intended force direction, as defined by the constraint function:

$$C_{\text{task}}(\widehat{g}_j) = \begin{cases} 1 & \text{if } \angle(f_j, f_{\text{req}}) < \theta_{\max}, \tau_j \geq \tau_{\min}, \\ 0 & \text{other}, \end{cases} \tag{2.13}$$

where $\theta_{\max}$ denotes the force direction vector corresponding to the grasp pose, f_{req} represents the desired force direction for the task (e.g., the rotation direction of a door handle), $\theta_{\max}$ indicates the permissible directional deviation, τ_j signifies the torque generated by the grasping action, and $\tau_{\min}$ denotes the minimum torque required to complete the task. A dual-constraint grasping operation model for a robotic arm based on scene constraints and task constraints is proposed, which integrates an optimized Contact-GraspNet. The process is shown in Fig. 6.

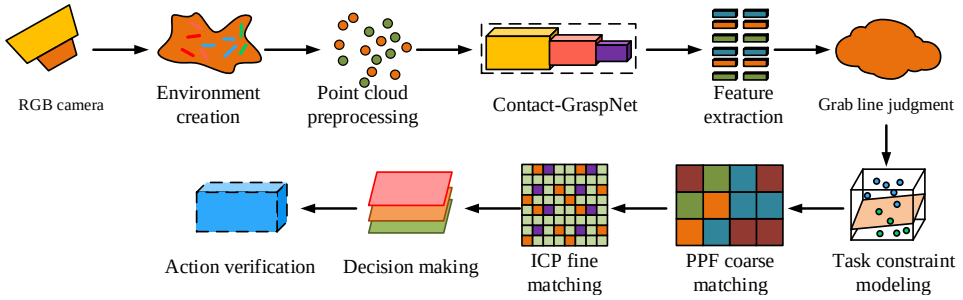

FIG. 6. Dual-constraint grasping operation model process of the robotic arm based on the optimized contact grasping network.

In Fig. 6, the point cloud is first obtained through an RGB-D camera and uniformly sampled to construct a 2D scene model containing target object and ob-

stacle information. Subsequently, based on the improved Contact-GraspNet, point cloud features are extracted, and the grasping base points, pose quaternions, scores, and grasp widths are output. On this basis, a task constraint module is introduced and combined with the functional attributes and operating direction of the target object to construct a physical motion model and filter out grasp candidates that do not satisfy the functional execution requirements. Next, PPF is used for coarse registration, and ICP is combined to complete point cloud fine registration, producing a more accurate object pose estimation. Ultimately, the system integrates grasp score, stability, and task feasibility to select the optimal grasping solution for path planning and action execution, and achieves closed-loop grasping control through sensor feedback to improve the overall grasping success rate and adaptability.

3. Results

3.1. Performance testing of the new robotic arm contact grasping network strategy model

The research is performed on the Ubuntu 20.04 platform to deploy the grasping model. The experimental environment is configured with an Intel Core i9-12900K CPU, an NVIDIA RTX 3090 GPU, and a memory capacity of 64 GB. The deep learning framework uses PyTorch 2.0, and the point cloud processing part integrates the Open3D and PCL libraries. The GraspNet-1Billion dataset (GraspNet) and the Yale-CMU-Berkeley Object and Model Set (YCB Dataset) are used as the test data sources. GraspNet is currently the largest real-world 3D grasping dataset. This dataset covers 190 different objects and generates over 1 billion grasp annotations, supporting RGB-D point cloud input and 6D grasp pose output. The YCB Dataset includes 77 common object categories encountered in daily life, such as cups, nuts, scissors, pot lids, etc. It contains RGB images, depth maps, segmentation masks, and object pose information. The study conducted training, validation, and testing on two datasets: GraspNet-1Billion and YCB Dataset. To ensure experimental reproducibility, GraspNet employs the officially provided scene-wise split: all objects and poses from scenes 0000–0845 form the training set, scenes 0846–0888 constitute the validation set, and scenes 0889–1000 comprise the test set. The training set contains 88.2 % of the data volume, while the test set comprises 11.8 %. For the YCB Dataset, the study employs an object-category-based splitting method: 70 % of objects are allocated to the training set and 30 % to the test set based on object ID, ensuring that the test objects do not appear during training to evaluate the model's cross-category generalization capability. No additional data augmentation is applied to either dataset. All point clouds underdo uniform cropping and

normalization processing before entering the network. Both datasets are widely used in robotic grasping, pose estimation, and action planning tasks. First, the two types of hyperparameters that have the greatest impact on the model performance are validated, and the selected values are shown in Fig. 7.

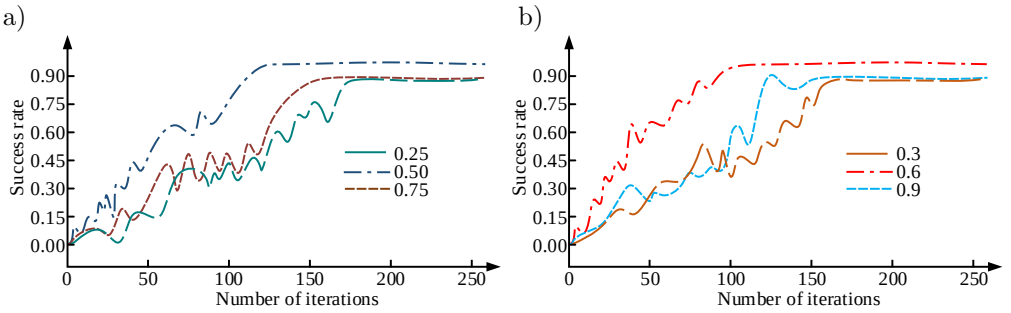

Fig. 7. Test results of hyperparameter selection: a) pose loss weight coefficient, b) matching tolerance coefficient.

According to Fig. 7a, as the number of iterations increased, the success rates of all three weight settings gradually increased, but the difference gradually became apparent after about 150 iterations. When the pose loss weight was set to 0.5, the model achieved a good balance between convergence speed and final success rate, and the final success rate remained stable above 0.93, which was better than for the 0.25 and 0.75 settings. The latter produced convergence fluctuation or overfitting trend. According to Fig. 7b, the curve converged the fastest when the matching tolerance was 0.6, reaching a success rate of 0.94 after about 100 iterations and maintaining high stability thereafter. When the tolerance was 0.3, the convergence speed was significantly slower, and the final success rate was slightly lower. Although the initial convergence was rapid with a tolerance of 0.9, it was prone to excessive mismatches in the middle and later stages, which could affect matching accuracy. After comprehensive consideration, the study ultimately chooses a pose loss weight of 0.5 and a matching tolerance coefficient of 0.6 as the default hyperparameter settings.

To simulate the closing process of a mechanical gripper in offline evaluation, this study adopts the parallel gripper closing model consistent with the official GraspNet evaluation. For each grasp candidate pose $\widehat{g}_j = (t_j, q_j, w_j)$ output by the network, the parallel gripper is first instantiated at the pose (t_j, q_j) with an initial opening width $w_{\max}$. The predicted grasp width w_j is truncated to the interval $[w_{\max}, w_{\min}]$. Subsequently, the distance between the fingers is gradually reduced along the gripper closing direction at a fixed step size (1 mm). Collisions between the gripper finger surfaces and the object/scene mesh are detected at each iteration step. When both finger surfaces make contact with the object

surface without colliding with the environment, the grasp is recorded as having reached a stable closing state, and the clamping width at this point is locked. If, during closure, either finger surface intersects with the table surface or other obstacles, or if effective contact cannot be achieved on both sides before reaching $w_{\min}$, the grasp candidate is deemed an infeasible grasp. Based on this, grasp success is calculated according to the rules defined in the GraspNet benchmark. If the closed pose simultaneously falls within the annotated grasp tolerance for both position and orientation, and no illegal collisions occur during the closing process, it is considered a successful grasp and counted toward valid grasp counts and metrics such as precision, recall, $F1$; otherwise, it is deemed a failed grasp. The entire closing and evaluation process is conducted entirely within the simulation environment, independent of real hardware testing. This ensures consistent evaluation conditions across different models. The research conducts ablation tests on the proposed model to further determine its performance, as presented in Fig. 8.

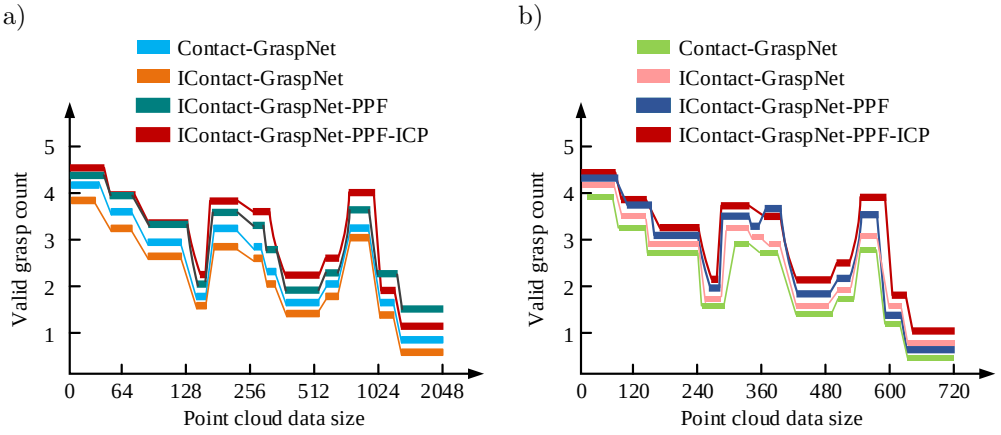

Fig. 8. Ablation test results: a) GraspNet, b) YCB Dataset.

In Fig. 8, the four model names represent different refinement stages: Contact-GraspNet denotes the original Contact-GraspNet baseline model, IContact-GraspNet refers to the proposed Improved Contact-GraspNet (abbreviated as 'IContact'), which incorporates point cloud uniform downsampling and PointNet++ local feature enhancement into the original model. IContact-GraspNet-PPF denotes the model with a coarse registration module based on PPFs added to IContact-GraspNet, IContact-GraspNet-PPF-ICP denotes the final version of the research model, which incorporates an ICP fine-alignment module after PPF coarse alignment to achieve the highest pose estimation accuracy and grasping stability. Here, the 'I' indeed stands for 'improved,' indicating that this model is an enhanced version of the original Contact-GraspNet.

From Fig. 8a, the original Contact-GraspNet model had an effective grasp count of 2.8 at 1024 input points. while the IContact-GraspNet model increased this value to 3.2. After introducing PPF, it was increased to 3.5. Finally, the integrated PPF+ICP model achieved the highest grasp count of 3.8 at the same number input points, indicating an increase of 35.7 %. Under 512 input points, the model also achieved an effective grasp of 3.4, significantly better than other model versions. In Fig. 8b, on the YCB dataset, with 720 input points, the IContact-GraspNet-PPF-ICP model had an effective grasp count of 3.6, which was higher than that of Contact-GraspNet (2.9), representing a performance improvement of over 24 %. Advanced grasping models were introduced for comparison, such as the grasp pose detection (GPD) model, the generative grasping convolutional neural network (GC-CNN), and the grasping residual convolutional network v2 (GR-ConvNet v2). Table 1 presents the test results, taking precision, recall, $F1$-score, and average execution time as indicators.

Table 1. Performance results of different models.

Dataset	Model	Precision [%]	Recall [%]	$F1$-score [%]	Average execution time [s]
GraspNet	GPD	85.73	82.45	84.06	0.94
	GC-CNN	88.22	85.36	86.77	0.68
	GR-ConvNet v2	89.91	87.52	88.74	0.72
	Our model	93.67	91.44	92.54	0.61
YCB Dataset	GPD	83.48	80.13	81.78	0.98
	GC-CNN	86.75	84.02	85.36	0.71
	GR-ConvNet v2	88.66	86.19	87.41	0.74
	Our model	92.88	90.75	91.82	0.63

In Table 1, on the GraspNet dataset, the precision of the proposed model reached 93.67 %, a recall was 91.44 %, and an $F1$-score reached 92.54 %, which was significantly better than GPD ($F1$-score of 84.06 %) and GR-ConvNet v2 ($F1$-score of 88.74 %). Meanwhile, its average execution time was only 0.61 s, better than GC-CNN (0.68 s), demonstrating strong computational efficiency. On the YCB dataset, the model still maintained its lead with an $F1$-score of 91.82 %, which was about 10 % higher than GPD. In addition, its precision and recall both exceeded 90 %, indicating that the model also has high robustness and good generalization capability in general object grasping scenarios. Through comprehensive comparison, the optimized contact grasping network is superior to the current mainstream models in terms of accuracy and real-time performance, and has significant advantages.

3.2. Simulation testing of the new robotic arm contact grasping network strategy model

All functional grasping simulations were implemented based on the standard DH parameters and dynamic model of the UR5 six-degree-of-freedom robotic arm. The corresponding motion execution performance metrics, such as grasping success rate, pose error, and energy consumption, are presented in Fig. 9, Fig. 10, and Table 2, respectively. To verify the practical performance effect of the proposed model, three typical grasping objects are randomly selected from the GraspNet dataset to evaluate whether the model can meet the functional execution requirements in operations with task objectives. The results are shown in Fig. 9.

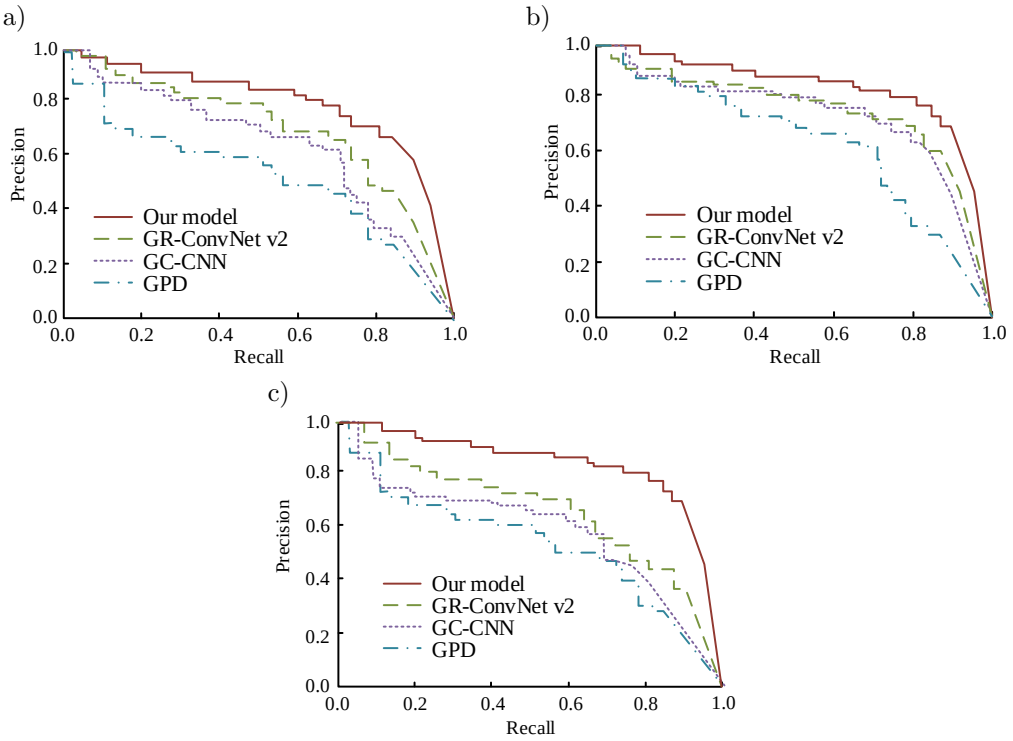

Fig. 9. Test results of functional operational adaptability: a) door handles, b) kettle handles, c) drawer pulls.

As shown in Fig. 9, the proposed model maintained the highest precision throughout the entire recall interval, especially in the 0.4–0.8 recall segment, where the precision remained stable above 0.85, while the precision of GPD dropped below 0.6 in this segment, showing the weakest performance. In the kettle handle grasp task, the proposed model still achieved a precision of 0.88 when the recall was 0.6, significantly higher than those of GC-CNN (0.74) and

GR-ConvNet v2 (0.79), demonstrating stronger stability and anti-interference ability. The drawer pull test results displayed that the proposed model maintained a precision of around 0.90 in most of the recall intervals. Even at high recall rates (above 0.9), there was no significant decrease in precision. However, the other three models showed a significant decrease in precision in this area, indicating that this model had stronger adaptability and robustness for edge grasping and functional component operation. The study continues to verify the average time latency of grasp prediction using different methods under different occlusions, and the results are shown in Fig. 10.

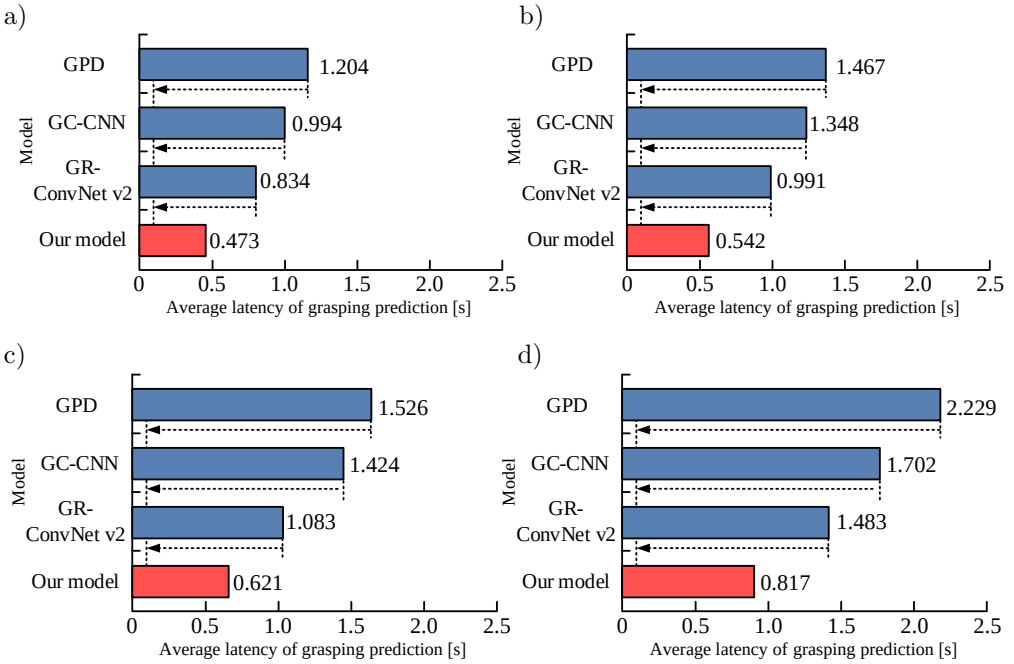

Fig. 10. Average latency results of grasp prediction of different methods under visual occlusion conditions: a) 0 %, b) 25 %, c) 50 %, d) 75 %.

As shown in Fig. 10, the proposed model consistently had the lowest grasp prediction latency and faster computational response under various occlusion conditions. Even in complex scenes with an occlusion rate of up to 75 %, the average latency of the system was still controlled within 0.82 s, far lower than other mainstream models whose latencies generally exceeded 1.4 s. In contrast, GPD and GC-CNN showed a significant increase in latency as the occlusion intensified, with larger fluctuations and insufficient stability. Although GR-ConvNet v2 showed a certain anti-interference ability, its latency still remained above 1 s in high occlusion situations. Relatively speaking, the inference process of the proposed model was more robust in occluded environments, reflecting the synergis-

tic advantages of structural optimization and lightweight coding. It is therefore more suitable for deployment in application scenarios that require high real-time performance. Finally, the grasp pose angle error, model parameter count, and energy consumption per grasp of each model under different lighting intensities were verified, as displayed in Table 2.

Table 2. Test results of model indicators under different lighting conditions.

Illumination level	Model	Pose angle error [°]	Model parameters [M]	Energy per grasp [J]
Low light	GPD	3.87	34.78	11.32
	GC-CNN	3.15	28.32	9.85
	GR-ConvNet v2	2.93	22.14	9.36
	Our model	1.64	18.53	8.45
Normal light	GPD	3.02	34.78	10.25
	GC-CNN	2.41	28.32	8.98
	GR-ConvNet v2	2.05	22.14	8.42
	Our model	1.21	18.53	7.84
High light	GPD	3.55	34.78	11.87
	GC-CNN	2.84	28.32	10.34
	GR-ConvNet v2	2.42	22.14	9.63
	Our model	1.87	18.53	9.27

To quantify the energy consumption differences among the various models during grasp action, this study employs a method based on integrating the joint torques and angular velocities to estimate the energy consumed in each grasping cycle. Specifically, during the entire process of the UR5 robotic arm moving from its initial standby pose to the target grasping pose and completing the closing motion, the torque $\tau_i(t)$ and angular velocity $\omega_i(t)$ of each joint were recorded at a frequency of 1 kHz. The mechanical energy consumption per grasping action is defined as

$$E = \sum_{i=1}^{6} \int_{t_0}^{t_f} \mid \tau_i(t)\omega_i(t) \mid \mathrm{d}t, \tag{3.1}$$

where t_f and t_0 denote the start and end times of the current grasp trajectory, respectively. The absolute values are used to prevent positive and negative work from canceling each other out. During actual computation, the integral is discretized into a sum over sampling points:

$$E^* = \sum_{i=1}^{6} \sum_{k} |\tau_i(k)\omega_i(k)| \, \Delta t. \tag{3.2}$$

For each lighting condition and each model, 50 successful grasps were performed repeatedly under identical initial poses and trajectory planning strategies. The average energy consumption value obtained is listed as 'energy per grasp [J]' in Table 2.

According to Table 2, in terms of pose angle error, the error of the proposed model under low-light, normal-light, and high-light conditions was 1.64°, 1.21°, and 1.87°, respectively, all significantly lower than those of other comparative models, reflecting the stable prediction ability of the model for grasp pose under complex visual conditions. The model parameter count was controlled at 18.53 M, which was more compact and computationally efficient compared with GPD (34.78 M) and GC-CNN (28.32 M). In terms of energy consumption, the new model had the lowest average energy consumption per grasp, only at 7.84 J (normal light), 8.45 J (low light), and 9.27 J (high light), further verifying the efficiency and practicality of the model in resource-limited scenarios. Overall, the proposed model outperformed the mainstream methods in terms of grasp pose estimation accuracy, structural complexity, and energy efficiency, demonstrating strong practical potential.

4. Conclusion

This study proposed an optimized contact grasping network model that integrates scene constraints and task constraints to satisfy the practical needs of robotic arm grasping operations in complex unstructured environments. By introducing point cloud downsampling, a lightweight encoder design, and a PointNet++ local feature enhancement mechanism, the improved network had a precision of 93.67 % and an $F1$-score of 92.54 % on the GraspNet, with an average execution time reduced to 0.61 s. Its performance was significantly better than that of mainstream models such as GPD, GC-CNN, and GR-ConvNet v2. In the functional grasping task test, the proposed model maintained accuracy advantages in three types of objects: door handles, kettle handles, and drawer pulls. At a recall rate of 0.6, the precision exceeded 0.88, verifying its adaptability and robustness in executing task-constrained grasping actions. Under visual occlusion conditions with a rate of up to 75 %, the model still maintained an average inference time of less than 0.82 s, displaying strong robustness to occlusion. Under different lighting conditions, the lowest pose angle error was only 1.21°, the model parameters were controlled at 18.53 M, and the lowest energy consumption per grasp was reduced to 7.84 J. The above results indicate that the proposed dual-constraint grasping strategy not only has comprehensive advantages in grasping accuracy, response speed, and resource efficiency, but also significantly improves grasp pose estimation and task execution performance in complex task-oriented grasping scenarios, and shows good potential for

engineering deployment. However, this study did not focus on grasping small objects. Future research will further combine multi-sensor fusion and reinforcement learning mechanisms to enhance the decision-making intelligence and execution stability of the model in multitask concurrent operations and dynamic scene adaptation.

Fundings

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Conflicts of interest

The author declares that there are no known competing financial interests or personal relationships that could have influenced the work described in this paper.

Author's contribution

The author conceptualized the study, performed the analysis, wrote the original draft, and approved the final manuscript.

References

1. Fang H.S., Gou M., Wang C., Lu C., Robust grasping across diverse sensor qualities: the GraspNet-1Billion dataset, *The International Journal of Robotics Research*, **42**(12): 1094–1103, 2023, https://doi.org/10.1177/02783649231193710.

2. Tian H., Wu W., Liu H., Zou J., Zhao Y., Robotic grasping of pillow spring based on MG-YOLOv5s object detection algorithm and image-based visual serving, *Journal of Intelligent & Robotic Systems*, **109**(3): 67–69, 2023, https://doi.org/10.1007/s10846-023-01989-x.

3. Chiu Y.J., Yuan Y.Y., Jian S.R., Design of and research on the robot arm recovery grasping system based on machine vision, *Journal of King Saud University-Computer and Information Sciences*, **36**(4): 102014–102017, 2024, https://doi.org/10.1016/j.jksuci.2024.102014.

4. Geng H., Hu Q., Wang Z., Optimization of robotic arm grasping through fractional-order deep deterministic policy gradient algorithm, *Journal of Physics: Conference Series*, **2637**(1): 12006–12011, 2023, https://doi.org/10.1088/1742-6596/2637/1/012006.

5. Yuan Y., Wang S., Mei Y., Zhang W., Sun J., Wang G., Improving world models for robot arm grasping with backward dynamics prediction, *International Journal of Machine Learning and Cybernetics*, **15**(9): 3879–3891, 2024, https://doi.org/10.1007/s13042-024-02125-3.

6. Xu F., Zhu Z., Feng C., Leng J., Zhang P., Yu X., Wang C., Chen X., An object planar grasping pose detection algorithm in low-light scenes, *Multimedia Tools and Applications*, **84**(9): 5583–5604, 2025, https://doi.org/10.1007/s11042-024-19128-5.

7. Yu X., Huang R., Zhao C., Zhou L., Ou L., Def-Grasp: a robot grasping detection method for deformable objects without force sensor, *Neural Processing Letters*, **55**(8): 11739–11756, 2023, https://doi.org/10.1007/s11063-023-11398-8.

8. Yan S., Zhang L., CR-Net: robot grasping detection method integrating convolutional block attention module and residual module, *IET Computer Vision*, **18**(3): 420–433, 2024, https://doi.org/10.1049/cvi2.12252.

9. Gilles M., Chen Y., Zeng E.Z., Wu Y., Furmans K., Wong A., Metagraspnetv2: all-in-one dataset enabling fast and reliable robotic bin picking via object relationship reasoning and dexterous grasping, *IEEE Transactions on Automation Science and Engineering*, **21**(3): 2302–2320, 2023, https://doi.org/10.1109/TASE.2023.3328964.

10. Hoang D.C., Nguyen A.N., Vu V.D., Nguyen V.T., Tran C.T., Ho N.T., Grasp configuration synthesis from 3D point clouds with attention mechanism, *Journal of Intelligent & Robotic Systems*, **109**(3): 71–76, 2023, https://doi.org/10.1007/s10846-023-02007-w.

11. Yu S., Zhai D.H., Xia Y., Robotic grasp detection based on category-level object pose estimation with self-supervised learning, *IEEE/ASME Transactions on Mechatronics*, **29**(1): 625–635, 2023, https://doi.org/10.1109/TMECH.2023.3287635.

12. Zhao X., Multifeature video modularized arm movement algorithm evaluation and simulation, *Neural Computing and Applications*, **35**(12): 8637–8646, 2023, https://doi.org/10.1007/s00521-022-08060-0.

13. Uçar K., Koçer H.E., Determination of angular status and dimensional properties of objects for grasping with robot arm, *IEEE Latin America Transactions*, **21**(2): 335–343, 2023, https://doi.org/10.1109/TLA.2023.10015227.

14. Zheng X., Yuan S., Chen P., Robotic autonomous grasping strategy and system for cluttered multi-class objects, *International Journal of Control, Automation and Systems*, **22**(8): 2602–2612, 2024, https://doi.org/10.1007/s12555-023-0358-y.

15. Muñoz J., López B., Quevedo F., Barber R., Garrido S., Moreno L., Geometrically constrained path planning for robotic grasping with differential evolution and fast marching square, *Robotica*, **41**(2): 414–432, 2023, https://doi.org/10.1017/S0263574722000224.

16. Denoun B., Hansard M., León B., Jamone L., Statistical stratification and benchmarking of robotic grasping performance, *IEEE Transactions on Robotics*, **39**(6): 4539–4551, 2023, https://doi.org/10.1109/TRO.2023.3306613.

17. Sekkat H., Moutik O., Ourabah L., Elkari B., Chaibi Y., Tchakoucht T.A., Review of reinforcement learning for robotic grasping: analysis and recommendations, *Statistics, Optimization & Information Computing*, **12**(2): 571–601, 2024, https://doi.org/10.19139/soic-2310-5070-1797.

18. Hu Z., Zheng Y., Pan J., Grasping living objects with adversarial behaviors using inverse reinforcement learning, *IEEE Transactions on Robotics*, **39**(2): 1151–1163, 2023, https://doi.org/10.1109/TRO.2022.3226108.

19. Lin T., Yue C., Wang P., Yu H., Cao X., Fixture-free assembly sequence and manipulation planning for single-arm robots, *Science China Technological Sciences*, **68**(9): 19204–19211, 2025, https://doi.org/10.1007/s11431-025-2992-8.

20. Shuai Y., Optimizing forward kinematics of a 6-DOF robotic arm for precision and efficiency, *Journal of Physics: Conference Series*, **3019**(1): 12020–12027, 2025, https://doi.org/10.1088/1742-6596/3019/1/012020.

Received August 26, 2025; revised March 2, 2026; accepted March 21, 2026; available online April 7, 2026; version of record September 16, 2026; published issue September 23, 2026.