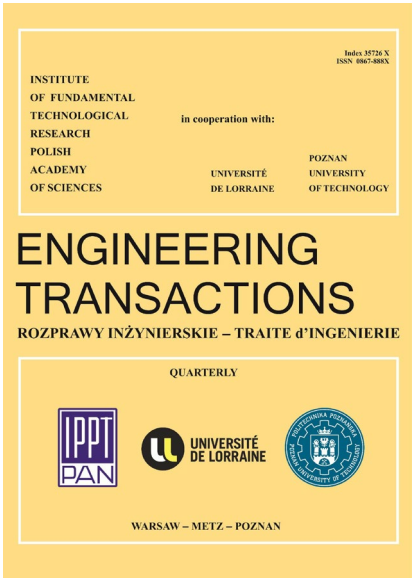
JOURNAL PRE-PROOF

This is an early version of the article, published prior to copyediting, typesetting, and editorial correction. The manuscript has been accepted for publication and is now available online to ensure early dissemination, author visibility, and citation tracking prior to the formal issue publication.

It has not undergone final language verification, formatting, or technical editing by the journal's editorial team. Content is subject to change in the final Version of Record.

To differentiate this version, it is marked as "PRE-PROOF PUBLICATION" and should be cited with the provided DOI. A visible watermark on each page indicates its preliminary status.

The final version will appear in a regular issue of *Engineering Transactions*, with final metadata, layout, and pagination.



Title: Optimization of Dual-Constrained Grasping Operation of Robotic Arm Based on Optimized Contact Grasping Network

Author(s): Feifei Zhao

DOI: https://doi.org/10.24423/engtrans.2026.3644

Journal: *Engineering Transactions*

ISSN: 0867-888X, e-ISSN: 2450-8071

Publication status: In press

Received: 2025-08-26

Revised: 2026-03-02

Accepted: 2026-03-21

Published pre-proof: 2026-04-07

Please cite this article as:

Zhao F., Optimization of Dual-Constrained Grasping Operation of Robotic Arm Based on Optimized Contact Grasping Network, *Engineering Transactions*, 2026, https://doi.org/10.24423/engtrans.2026.3644

Optimization of Dual-Constrained Grasping Operation of Robotic Arm Based on Optimized Contact Grasping Network

Feifei Zhao

School of Intelligent Manufacturing, Zibo Vocational Institute, Zibo, 255300, China

zhaoffzhaoff@163.com

Abstract: With the continuous improvement of intelligent manufacturing and robot perception capabilities, traditional robotic arm grasping methods still face problems such as inaccurate posture prediction and poor task adaptability in dealing with dynamic scenes, complex objects, and functional action execution. Therefore, the research develops an optimized contact grasping network model that integrates scene constraints and task constraints. It combines UR5 six degree of freedom robotic arm, visual input, and Contact-GraspNet architecture based on point cloud, and introduces PointNet++ local feature enhancement mechanism and 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 Dataset, the model achieves F1 scores of 92.54% and 91.82% respectively, with average execution time reduced to 0.61 seconds. In functional operation scenarios involving door handles, kettle handles, and drawer pulls, grasping accuracy remained above 0.89, with task completion rates significantly outperforming mainstream baseline models. Under visual interference conditions with up to 75% occlusion rate, the average inference latency was controlled within 0.82 seconds. 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 precision, latency control, and deployment efficiency, and has the potential to be largely applied in industrial, service, and special task environments.

Keywords: Robotic arm grasping; Scene constraints; Task constraints; Contact-GraspNet; Point cloud

1 Introduction

In the fields of automated manufacturing, service robots, and intelligent warehousing, the

robotic arm grasping system, as a key link in human-computer interaction and object manipulation, is gradually developing towards high precision, high robustness, and adaptive capabilities. In practical application scenarios, the grasping of robotic arms is not only limited by the physical environment of the scene, but also affected by the dual-constraint 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 dual-constraint of task level and scene level while retaining the efficient feature learning ability of deep grasping networks is the core problem that urgently needs to be solved.

(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 Y J et al. 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 [3]. Researchers like Geng H. enhanced robotic arm grasping stability and robustness through visual servo and fractional-order control approaches [4]. Concurrently, Yuan Y. improved adaptability to diverse objects by integrating multimodal information fusion and refining world models [5]. However, such methods generally rely on rigid geometric assumptions or explicit scene configurations, exhibiting limited adaptability to complex conditions like 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, or rotating.

(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 direction. Xu F et al. proposed a grasping pose estimation method integrating multi-scale residual networks with RGB sensors, maintaining high recognition accuracy

even in low-light environments [6]. 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 S et al. enhanced RGB-D feature extraction by incorporating attention and residual modules; Gilles M et al. achieved efficient grasping of unknown objects by integrating MetaGraspNetV2; Hoang D C improved point cloud denoising and grasp position inference stability through attention mechanisms and the VoteGrasp strategy [8-10]. While these approaches strengthen 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-functional constraints, failing to determine whether grasp poses match specific operational requirements.

(3) Development 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 S et al.'s SCNet introduced a self-supervised mechanism in 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) [11]. However, most approaches still lack systematic modeling of task constraints and rarely address critical factors such as mechanical 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 action stability.

In summary, although existing robotic arm grasping methods have made some progress in pose estimation and grasping pose inference based on point cloud, they still face performance degradation and insufficient robustness in handling geometric interference, adapting functional

operation constraints, and ensuring contact grasping stability in complex scenarios. Therefore, the research builds a robotic arm contact grasping strategy model that fuses an optimized Contact-GraspNet structure and a dual-constraint mechanism. This model improves the point cloud representation efficiency by introducing local feature extraction and channel lightweight clipping strategies based on PointNet++, and combines action mechanics models under task constraints to achieve functional target oriented grasping posture discrimination. On this basis, the research further integrates coarse registration 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 architecture of Contact-GraspNet to illustrate its fundamental workflow and input-output format. Subsequently, several proposed improvements are presented, including: ① a point cloud uniformity downsampling strategy, ② a local geometric feature enhancement mechanism based on PointNet++, ③ lightweight design for encoder channel pruning, ④ task-constraint-coupled grasp candidate filtering strategy, and ⑤ two-stage pose optimization based on Point Pair Feature Matching (PPF) algorithm + Iterative Closest Point (ICP).

2.1 Construction of robotic arm grasping network model considering scene constraints

The UR5 six degree of freedom robotic arm is selected as the operating subject for the study. The arm has a compact structure and high repeatability positioning accuracy (up to ± 0.1 mm). It is widely used in the operation and grasping research of complex objects [12-14]. The study first models grasping operations within a two-dimensional plane, assuming the target object rests stationary on a horizontal workbench. The schematic diagram of the two-dimensional grasp and the grasp pose diagram are shown in Figure 1 [15-16]. It should be noted that the introduction of the two-dimensional

grasping schematic (Figure 1) during the modeling phase is not intended to construct a two-dimensional grasping model. Rather, it serves to visually illustrate the geometric relationship between the robotic end-effector and the target object, including the grasping base point, gripper width, formation of the contact triangle, and definition of the pose vector. This flattened schematic serves solely to explain three-dimensional symbols, parameters, and grasp semantics, and does not involve subsequent algorithmic reasoning or experimental processes.

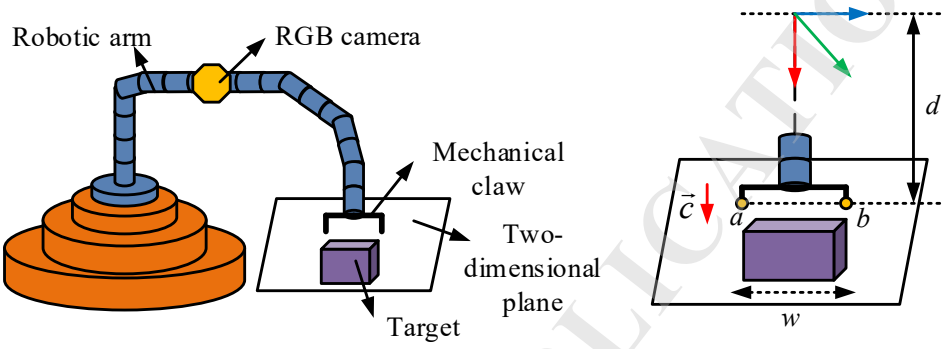


Figure 1 Schematic diagram of 2D planar gripping and gripping position of the robotic arm (Source from: Author's self drawn)

Figure 1 (a) displays the 2D plane grasping of the robotic arm, and Figure 1 (b) displays the 2D plane grasping pose. Figure 1 presents a two-dimensional schematic of three-dimensional grasping parameters, serving solely to illustrate the pose definition method. Both model training and inference are implemented based on three-dimensional point clouds. In Figure 1 (a), 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 grasping point position and posture based on visual input. As shown in Figure 1 (b), the gripper establishes contact with the surface of the object through points a and b on both sides, forming a stable supported contact triangle. Parameter w represents the width of the gripper, d signifies the Euclidean distance from the gripper center to the reference coordinate origin. The pose direction of the gripper is represented by vector $\vec{c}$. To efficiently predict grasping point position and posture, the study introduces Contact-GraspNet as the basic network model. This network is an end-to-end grasping

detection framework based on 3D point clouds, which can directly predict the feasible grasping pose of each point from the point cloud data generated from RGB-D images, and has excellent scene adaptation ability and contact semantic reasoning ability [17-18]. The training process is shown in Figure 2.

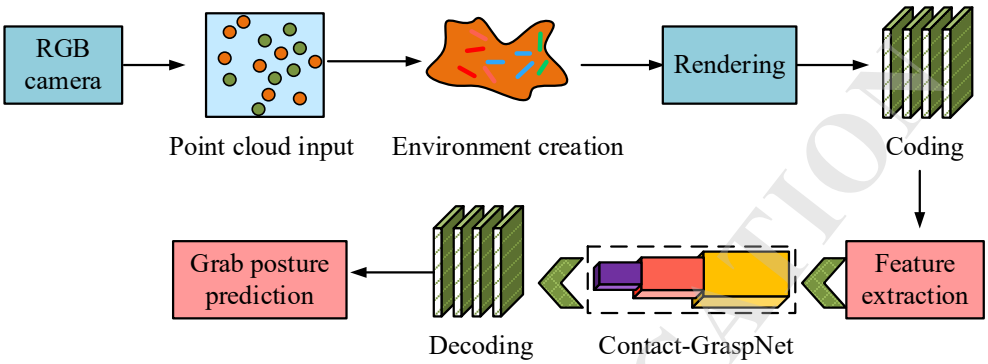


Figure 2 Training flow of Contact-GraspNet (Source from: Author's self drawn)

In Figure 2, the training of Contact-GraspNet mainly revolves around four stages: encoding, feature extraction, grasping 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 scenario. By integrating this depth information, we generate a 3D point cloud as 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 form is presented in equation (1).

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

In equation (1), p_i represents the i-th coordinate point. o signifies the voxel grid origin. r represents voxel resolution. f_i signifies the local feature of the i-th point. Secondly, the network predicts the grasping transformation $\hat{T}_j \in SE$ for each voxel center point v_j, usually represented by the translation vector t_j and the axial rotation vector ω_j, as shown in equation (2).

$$\hat{T}_j=\begin{bmatrix}\exp(\left[w_j\right]_\times) & t_j\\ 0 & 1\end{bmatrix} \qquad (2)$$

In equation (2), $\exp(\left[w_j\right]_\times)\in SO(3)$ represents the mapping from Lie algebra to Lie group.

$\left[w_j\right]_\times$ represents the antisymmetric matrix form of vector w_j. Then, Contact-GraspNet assigns a graspable score $\hat{s}_j \in [0,1]$ to each candidate grasping pose g_j during training. The regression uses binary cross-entropy, as calculated by equation (3).

$$L_{score} = -\frac{1}{M}\sum_{j=1}^{M}(s_j \log \hat{s}_j + (1-s_j)\log(1-\hat{s}_j)) \qquad (3)$$

In equation (3), L_{score} represents the grasp rating loss. $s_j \in [0,1]$ represents the true label, which determines whether it is feasible to grasp. $\hat{s}_j$ represents the retrievable probability predicted by the network. M represents the number of candidate samples to be grasped. The total loss function is the sum of the grasping score loss and pose regression loss, as shown in equation (4).

$$L_{total} = L_{score} + \lambda_{pose} \cdot L_{pose} \qquad (4)$$

In equation (4), L_{pose} represents the point-to-point translation error and the rotation angle error. λ_{pose} represents the weight coefficient of attitude loss. L_{total} represents the total loss function. Although Contact-GraspNet has achieved good results in grasping prediction based on point cloud, its original model lacks a unified standard for inputting point cloud data, resulting in insufficient learning stability of target grasping features when object density changes significantly. Therefore, a point cloud down-sampling optimization based on distance uniformity is analyzed, as shown in equation (5).

$$P' = FPS(P, N), N = 1024 \qquad (5)$$

In equation (5), P represents the original input point cloud. P' represents the sampled point cloud input. $FPS(\_)$ represents the Farthest Point Sampling function. PointNet++ local feature encoding is used to replace single-layer global perception, as shown in equation (6).

$$f_i^* = MLP\left(\underset{j \in N(p_i)}{\oplus} \phi(p_j - p_i)\right) \qquad (6)$$

In equation (6), $N(p_i)$ represents a spherical neighborhood centered around p_i. $\phi(\_)$ represents the geometric encoding function. $\oplus$ represents the feature concatenation operation.

MLP represents Multi-Layer Perceptron (MLP) processing. Finally, the overall channel and layers are trimmed and lightweight, and the calculation expression is shown in equation (7).

$$R^{1024\times3} \xrightarrow{Enc} R^{512\times128} \xrightarrow{Dec} R^{1024\times64} \qquad (7)$$

In equation (7), *Enc* represents the encoder. *Dec* represents the decoder. 1024 represents the number of points. 128 / 64 represents the number of feature channels. At this point, the improved Contact-GraspNet structure is shown in Figure 3.

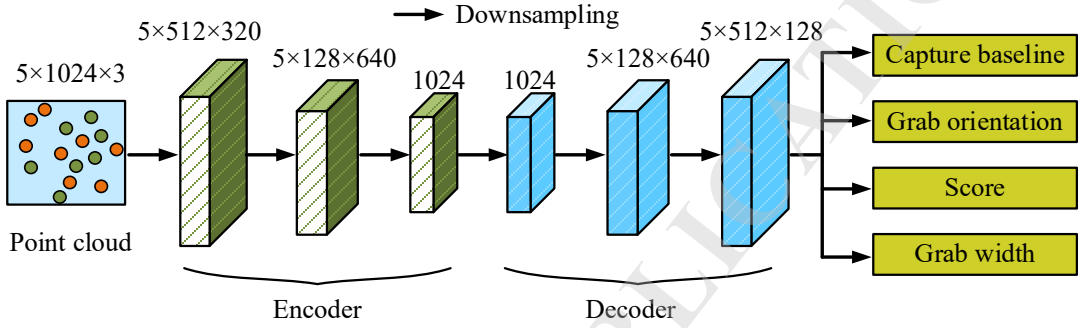


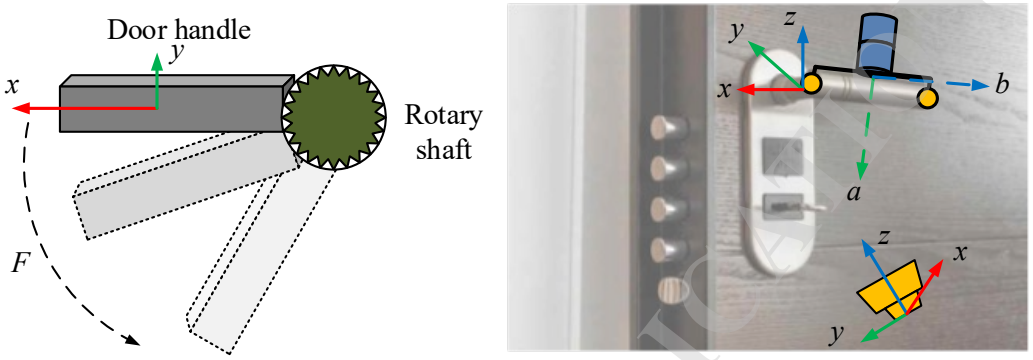
Figure 3 Improved Contact-GraspNet structure (Source from: Author's self drawn)

In Figure 3, the input point cloud is first unified into batch data of size 5×1024×3 and uniformly sampled using Farthest-Point Sampling. The encoder consists of three convolutional layers, producing feature maps of sizes 5×512×320, 5×128×640, and 5×512×128 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 5×1024×128. 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 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 the task adaptability and operational rationality of grasping actions, a task constraint mechanism is introduced to recognize the geometric feasibility of grasping positions, and

comprehensively judge whether the grasping actions meet the dynamic requirements of specific operational goals. 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 physics model and motion filtering operation diagram are shown in Figure 4.



(a) Door handle motion physical model

(b) Schematic diagram of motion filtering operation

Figure 4 The physical model of door handle motion and the operation schematic diagram of motion filtering (Figure 4 (a) Source from: Author's self drawn) (Figure 4 (b) Source from: https://colorhub.me/photos/JP4m1)

Figure 4 (a) shows the physical model of door handle motion, and Figure 4 (b) illustrates the motion filtering operation. As shown in Figure 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 axis, satisfying formula $T = F \cdot R$. Taking the local coordinate system of the door handle as a reference, the angle relationship between each candidate grasping posture and the rotation axis and the consistency of the applied force direction are calculated. If the contact direction of a certain grasping point deviates too much from the expected force direction, it will be filtered out by the task constraint module, and only grasping candidates that meet both contact stability and effective action ability will be retained. To achieve higher accuracy in grasping execution while satisfying task constraints, this study further adopts a pose estimation optimization method on the basis of Point Pair Feature Matching (PPF) algorithm, which accurately perceives and models the actual pose in the pre-grasping stage. Subsequently, based on coarse registration, the

Iterative Closest Point (ICP) 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 process of both is shown in Figure 5 [19-20].

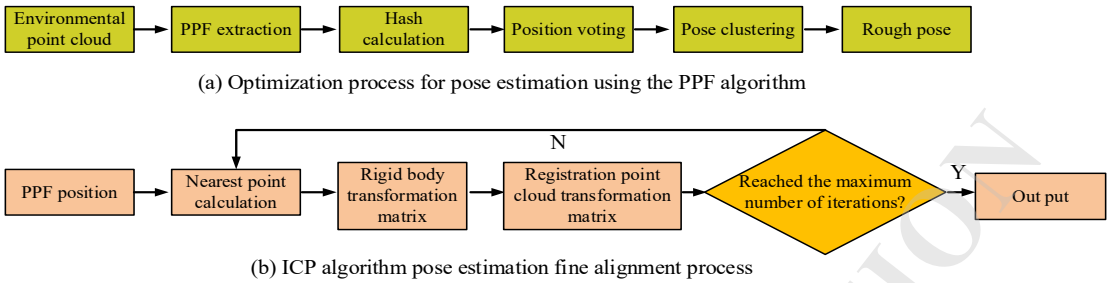


Figure 5 PPF and ICP processes (Figure 5 (a) Source from: Author's self drawn) (Figure 5 (b) Source from: Author's self drawn)

Figure 5 (a) shows the pose estimation optimization process of the PPF algorithm, and Figure 5 (b) shows the pose estimation fine alignment process of the ICP algorithm. As shown in Figure 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 shown in Equation (8).

$$\Phi_{ab}=(\left\|x_a-x_b\right\|,\angle(n_a,d_{ab}),\angle(n_b,d_{ab}),\angle(n_a,n_b)) \qquad (8)$$

In equation (8), x_a and x_b both represent the position coordinates of the point pair. n_a and n_b represent the unit normal vectors corresponding to two point pairs. d_{ab} represents the point pair direction. After establishing an index on the feature quadruple 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 shown in equation (9).

$$S(H_k)=\sum_{\Phi\in F}\delta(\left\|\Phi-H_k(\Phi_m)\right\|<\varepsilon) \qquad (9)$$

In equation (9), H_k represents the k-th candidate rigid body transformation. Φ_m represents the model features. $\Phi\in F$ represents the feature set extracted from the scene point cloud. δ represents the matching count function within the threshold. ε represents the matching tolerance

coefficient. ICP aims to minimize the Euclidean distance between the target point cloud and the transformed model point cloud. A matching error function is constructed, as shown in equation (10).

$$\tau(T) = \sum_{i=1}^{N} \left\| y_i - T \cdot z_i \right\|^2 \qquad (10)$$

In equation (10), $T \in SE(3)$ represents the rigid body transformation matrix to be optimized. z_i represents the points in the model point cloud. y_i represents 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 shown in equation (11).

$$T^* = \arg\min_{R,t} \sum_{i=1}^{N} \left\| y_i - (R \cdot z_i + t) \right\|^2 \qquad (11)$$

In equation (11), 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 square 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_scene   ← scene point cloud
    P_model   ← object model point cloud
    T_init    ← initial pose estimate (optional)

Output:
    T_final   ← 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_model ← 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_scene ← PPF(s_l, s_i)
        retrieve matching entries from H
        accumulate votes for each candidate transform T_c
    T_ppf ← candidate transform with the highest vote count

if T_init exists then
    T_coarse ← T_init ◦ T_ppf
else
    T_coarse ← T_ppf

------------------------------------------------------------
# Stage 2: Fine Pose Refinement via ICP
------------------------------------------------------------

Set max iterations K and convergence threshold ε
E_prev ← +∞

for iter = 1 to K do
    P_model' ← T_coarse(P_model)
    C ← nearest-neighbor correspondences between P_model' and P_scene
    E ← Σ || p' - q ||²    for (p', q) in C
    if |E - E_prev| < ε then
        break
    ΔT ← optimal rigid transform estimated by SVD over C
    T_coarse ← ΔT ◦ T_coarse
    E_prev ← E

T_final ← T_coarse
```

Additionally, scene constraints refer to restrictions in three-dimensional space that determine whether a grasping candidate pose $\hat{g}_j$ satisfies environmental geometry, object surface accessibility, and collision feasibility. Scene constraints are expressed as shown in Equation (12).

$$C_{scene}(\hat{g}_j) = \begin{cases} 1 & if d(\hat{g}_j, O) > \varepsilon, n_j \cdot s_j > 0 \\ 0 & other \end{cases} \quad (12)$$

In Equation (12), O denotes the set of obstacles in the scene; $d(\hat{g}_j, O)$ represents the minimum distance between the grasping pose and the nearest obstacle; ε denotes the safety margin threshold; n_j denotes the surface normal of the grasping surface; s_j denotes the claw closing direction; $n_j \cdot s_j$ represents the dot product of 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 grasping pose satisfies functional operation requirements (e.g., rotating door handles, pulling drawers, pouring kettles). These constraints ensure the grasping pose aligns with the intended force direction, as defined by the constraint function in Equation (13).

$$C_{task}(\hat{g}_j)=\begin{cases}1 & if \angle(f_j, f_{req})<\theta_{\max}, \tau_j \geq \tau_{min} \\ 0 & other\end{cases} \quad (13)$$

In Equation (13), f_j denotes the force direction vector corresponding to the grasping posture; 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; τ_{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 optimized contact grasping network. The process is shown in Figure 6.

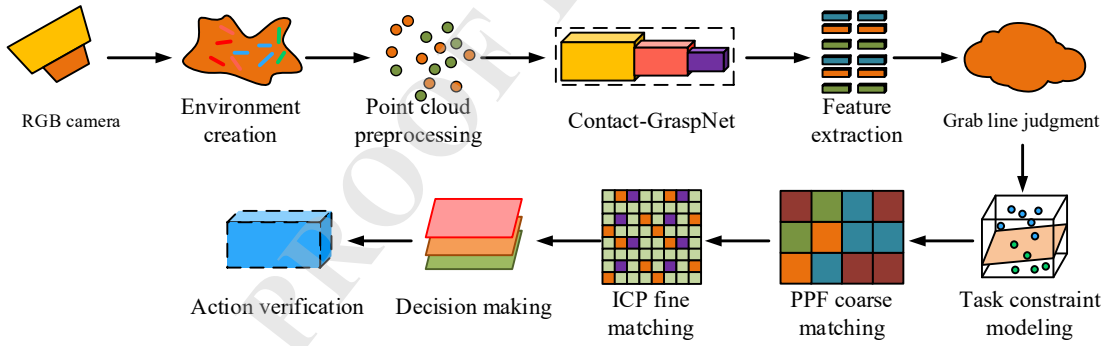


Figure 6 The dual-constraint grasping operation model process of the robotic arm based on the optimized contact grasping network (Source from: Author's self drawn)

In Figure 6, firstly, the point cloud is obtained through RGB-D camera and uniformly sampled to construct a 2D scene model containing target object and obstacle information. Subsequently, based on the improved Contact-GraspNet, point cloud features are extracted, and the grasped base points, pose quaternions, scores, and grasp widths are output. On this basis, a task constraint module is introduced, combined with the functional attributes and operation direction of the target object, to construct a physical motion model and filter out grasping candidates that do not satisfy

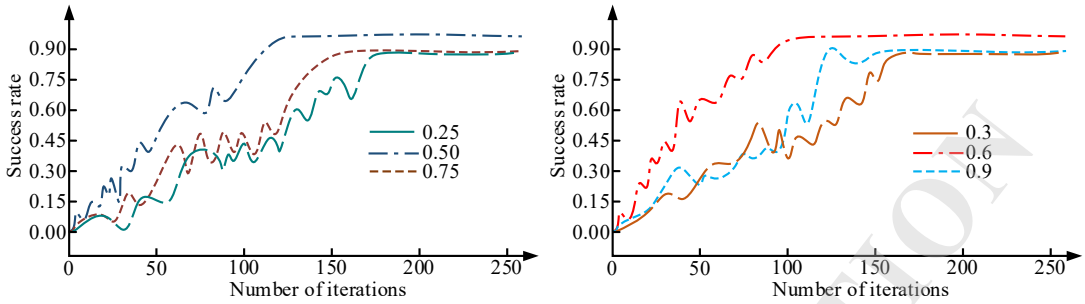
the functional execution requirements. Next, the PPF is used for coarse registration, and the ICP is combined to complete point cloud fine registration, obtaining more accurate object pose estimation. Ultimately, the system integrates scoring, stability, and task feasibility to select the optimal grasping solution for path planning and action execution, and achieves closed-loop grasping control through sensor feedback to improve 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 based on the Ubuntu 20.04 platform to deploy a grasping model. The experimental environment is configured with Intel Core i9-12900K CPU and NVIDIA RTX 3090 GPU, with a memory capacity of 64GB. The deep learning framework uses PyTorch 2.0, and the point cloud processing part integrates Open3D and PCL libraries. The GraspNet-1Billion dataset (GraspNet) and the Yale-CMU-Berkeley Object and Model Set (YCB Dataset) are used as test data sources. GraspNet is currently the largest real 3D grasp dataset. This dataset covers 190 different objects and generates over 1 billion grasping annotations, supporting RGB-D point cloud input and 6D grasping pose output. The YCB Dataset dataset includes 77 common object categories in daily life, such as cups, nuts, scissors, pot lids, etc. It is equipped with 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 test objects do not appear during training to evaluate the model's cross-category generalization capability. No additional data augmentation was applied to either dataset. All point clouds underwent uniform cropping and normalization processing before

entering the network. It is widely used in robotic grasping, pose estimation, and action planning tasks. Firstly, the two types of hyperparameters that have the greatest impact on the model performance are validated, and the selected values are shown in Figure 7.



(a) Pose loss weight coefficient (b) Matching tolerance coefficient

Figure 7 Test results of hyperparameter selection (Source from: Author's self drawn)

Figure 7 (a) displays the pose loss weight coefficient λ_{pose}, and Figure 7 (b) displays the matching tolerance coefficient ε. According to Figure 7 (a), as the number of iterations increased, all three sets of weights gradually increased the success rate, 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 the 0.25 and 0.75 groups. The latter had convergence fluctuation or over-fitting trend. According to Figure 7 (b), the curve converged the fastest when the matching tolerance was 0.6, reaching a success rate of 0.94 after about the 100th iteration and maintaining high stability. When the tolerance was 0.3, the convergence speed was significantly slower and the final success rate was slightly lower. Although the initial convergence is rapid with a tolerance of 0.9, it is prone to excessive mismatches in the middle and later stages, which can affect the judgment 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 $\hat{g}_j = (t_j, q_j, w_j)$ output by the network, the parallel gripper is first instantiated into

position (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 fingers is gradually reduced along the gripper closure 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, it is recorded as a stable closure 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 closure, it is considered a successful grasp and counted toward valid grasp counts and metrics like Precision/Recall/F1; otherwise, it is deemed a failed grasp. The entire closure 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, as presented in Figure 8.

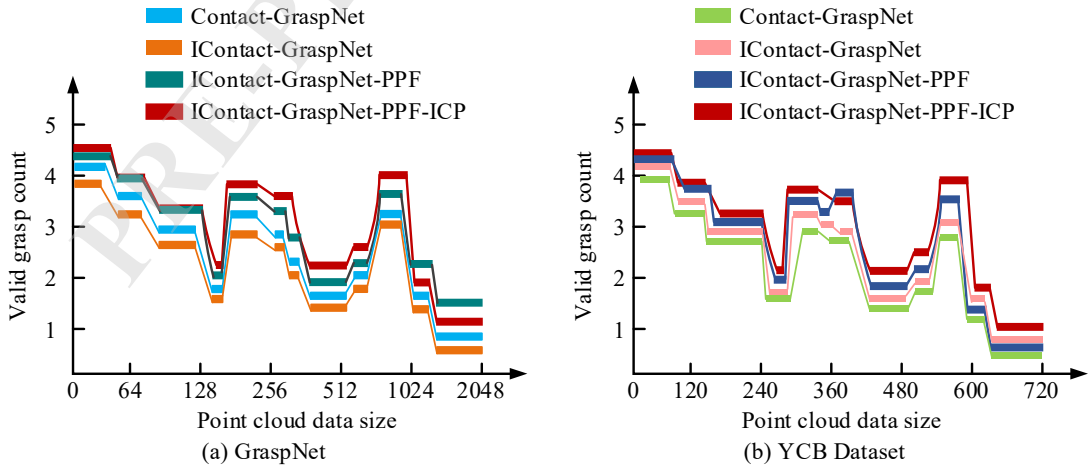


Figure 8 Ablation test results (Source from: Author's self drawn)

Figure 8 (a) displays the results on the GraspNet dataset, and Figure 8 (b) displays the results

on the YCB dataset. In Figure 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 uniformity downsampling and PointNet++ local feature enhancement over the original model; IContact-GraspNet-PPF denotes the model with a coarse registration module based on Point Pair Features (PPF) added to IContact-GraspNet; IContact-GraspNet-PPF-ICP denotes the final version of the research model, which incorporates an Iterative Closest Point (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 Figure 8 (a), the original Contact-GraspNet model had an effective grasp count of 2.8 at 1,024 input points. while the IContact-GraspNet model was increased 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 times at the same input points, with an increase of 35.7%. Under 512 input points, the model also achieved 3.4 iterations, significantly better than other structures. In Figure 8 (b), 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), with a performance improvement of over 24%. Advanced grasping models have been introduced for comparison, such as the Grasp Pose Detection (GPD) model, Generative Grasping Convolutional Neural Network (GC-CNN), and 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 The index test 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

YCB Dataset	GR-ConvNet v2	89.91	87.52	88.74	0.72
	Our model	93.67	91.44	92.54	0.61
	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, the precision of the proposed model reached 93.67%, the recall was 91.44%, and the F1 score reached 92.54%, which was significantly better than GPD (F1 score of 84.06%) and GR-ConvNet v2 (F1 score of 88.70%). Meanwhile, its average execution time was only 0.61 seconds, better than GC-CNN (0.68 seconds), demonstrating strong operational efficiency. On the YCB dataset, the model still maintained its lead with an F1 score of 91.80%, which was about 10% higher than GPD. In addition, its precision and recall both exceed 90%, indicating that the model also has high robustness and generalization ability in general object scenarios. Through comprehensive comparison, the optimized contact grasping network is superior to current mainstream models on precision 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 Figure 9, Figure 10, and Table 2, respectively. To verify the practical application effect, 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 Figure 9.

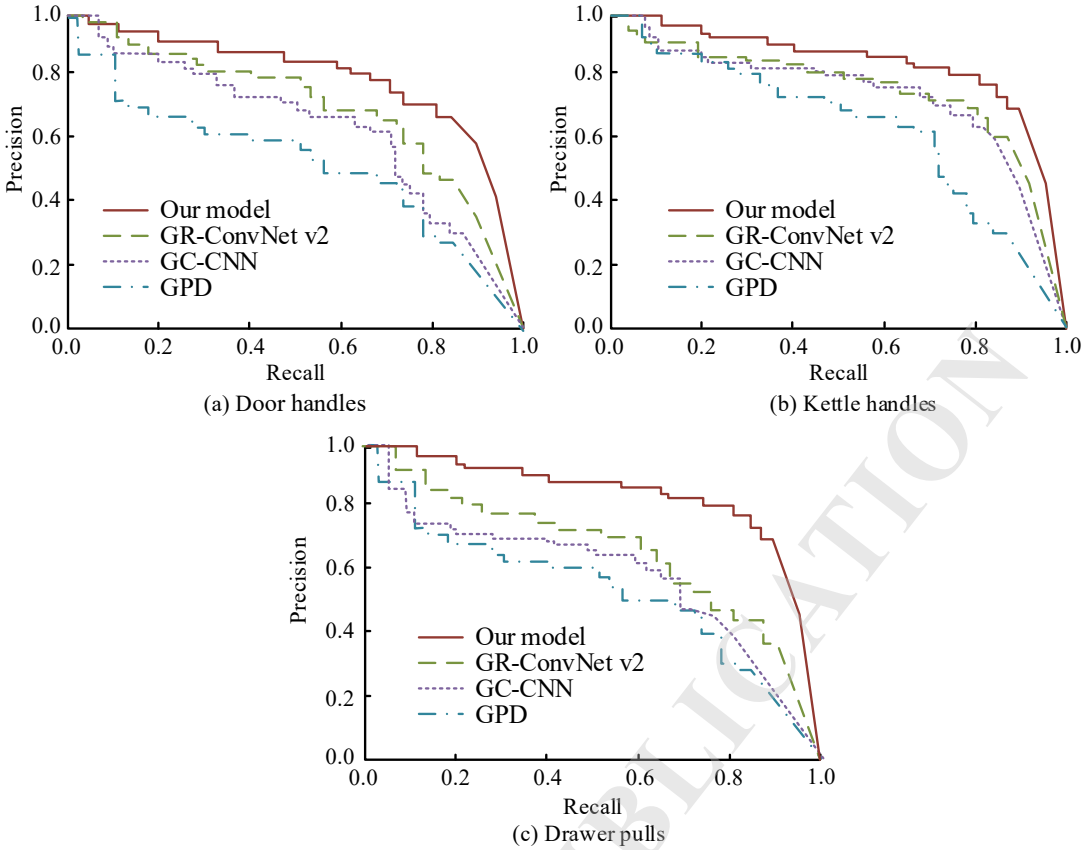


Figure 9 Test results of functional operational adaptability (Source from: Author's self drawn)

Figure 9 (a) displays the grasping test results on door handles, Figure 9 (b) displays the grasping test results on kettle handles, and Figure 9 (c) displays the grasping test results on drawer pulls. From Figure 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 GPD dropped below 0.6 in this segment, showing the weakest performance. In the grab task of the kettle handle, the proposed model still achieved a precision of 0.88 when the recall was 0.6, significantly higher than that 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 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 in edge grabbing and functional component operation. The study continues to verify the average time latency of grasping prediction using different methods under

different occlusions, and the results are shown in Figure 10.

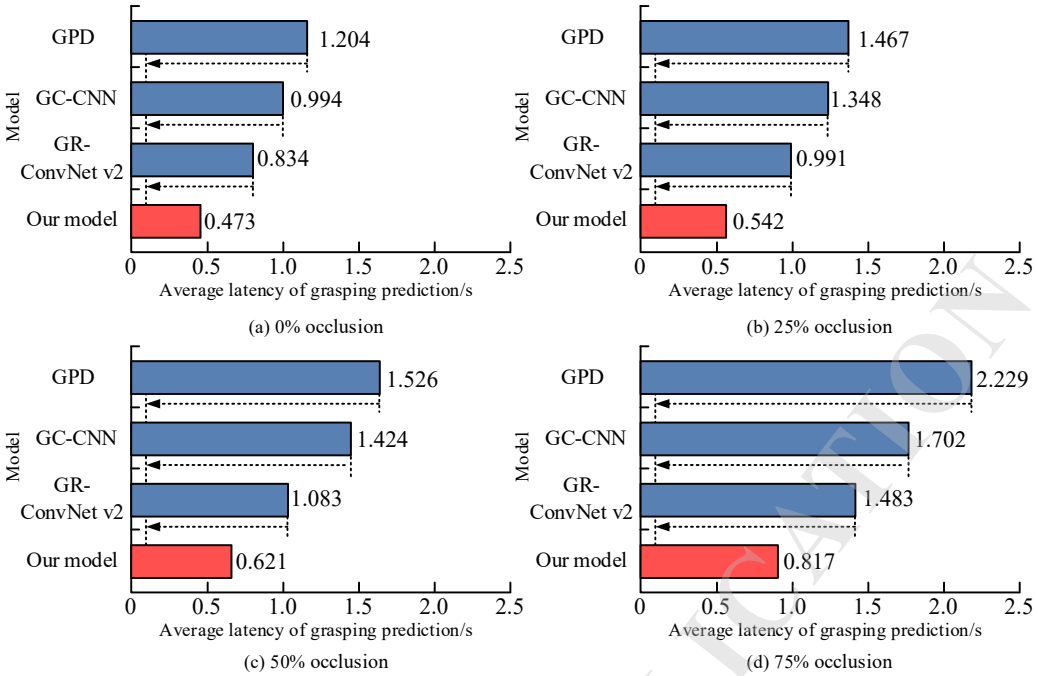


Figure 10 The average latency results of grasping by different methods under visual occlusion conditions (Source from: Author's self drawn)

Figure 10 (a) compares the grasping latency under 0% occlusion, Figure 10 (b) compares the grasping latency under 25% occlusion, Figure 10 (c) compares the grasping latency under 50% occlusion, and Figure 10 (d) compares the grasping latency under 75% occlusion. As shown in Figure 10, the proposed model consistently had the lowest grasping 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 seconds, far lower than other mainstream models that generally exceed 1.4 seconds. In contrast, GPD and GC-CNN show a significant increase in latency when occlusion intensifies, with larger fluctuations and insufficient stability. Although GR-ConvNet v2 has a certain anti-interference ability, the latency still remains above 1 second in high occlusion situations. Relatively speaking, the inference process of the proposed model is more robust in occluded environments, reflecting the synergistic advantages of structural optimization and lightweight coding. It is also more suitable for deployment in application scenarios that require high real-time performance. Finally, the grasping pose angle error, model parameter count, and single grasping energy consumption of each model

under different lighting intensities are verified, as displayed in Table 2.

To quantify the energy consumption differences among various models during the grasping action, this study employs a method based on integrating joint torques and angular velocities to estimate the energy per grasping cycle. Specifically, during the entire process of the UR5 robotic arm moving from its initial standby posture to the target grasping posture and completing the closure, 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 shown in Equation (14).

$$E = \sum_{i=1}^{6} \int_{t_0}^{t_f} |\ \tau_i(t)\omega_i(t)|\ dt \qquad (14)$$

In Equation (14), 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, as shown in Equation (15).

$$E^* = \sum_{i=1}^{6} \sum_{k} \left|\tau_i(k)\omega_i(k)\right| \Delta t \qquad (15)$$

For each lighting condition and each model, 50 successful grasps were performed repeatedly under identical initial poses and trajectory planning strategies. The average value obtained is listed as "Energy per Grasp (J)" in Table 2.

Table 2 Test results of model indicators under different light intensities

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

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 other comparative models, reflecting the stable prediction ability of the model for grasping posture under complex visual conditions. Its model parameter count was controlled at 18.53M, which was more compact and computationally efficient compared with GPD (34.78M) and GC-CNN (28.32M). In terms of energy consumption performance, the new model had the lowest average energy consumption per grasp, only at 7.84J (normal light), 8.45J (low light), and 9.27J (high light), further verifying the efficiency and practicality of the model in resource limited scenarios. Overall, the proposed model outperforms mainstream methods on accuracy, structural complexity, and energy efficiency, demonstrating strong practical potential.

4 Conclusion

This study proposed an optimized contact grasping network model that integrated scene constraints and task constraints to satisfy the practical needs of robotic arm grasping operations in complex unstructured environments. By introducing point cloud downsampling, lightweight encoder design, and 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 seconds. Its performance was significantly better than 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 constraint actions. Under visual interference conditions with an occlusion rate of up to 75%, the model still maintained an average inference time of less than 0.82 seconds, displaying strong anti-interference ability. Under different lighting intensities, the lowest attitude angle error was only 1.21°, the model parameters were controlled at 18.53M, and the lowest single energy consumption was reduced to 7.84J. The above results indicate that the dual-constraint grasping strategy not only has comprehensive advantages in accuracy, response speed, and resource efficiency, but also significantly improves the perception and execution ability of the grasping system for dynamic task targets, and has good engineering deployment potential. However, this study has not yet focused on fine 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 multitasking concurrent operations and dynamic scene adaptation.

Conflicts of interest

All authors declare that they have no conflicts of interest.

Reference

[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, 2023, 42(12): 1094-1103. 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, 2023, 109(3): 67-69. 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, 2024, 36(4): 102014-102017. 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. IOP Publishing, 2023, 2637(1): 12006-12011. 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, 2024, 15(9): 3879-3891. 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, 2025, 84(9): 5583-5604. 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, 2023, 55(8): 11739-11756. 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, 2024, 18(3): 420-433. 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, 2023, 21(3): 2302-2320. 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, 2023, 109(3): 71-76. 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, 2023, 29(1): 625-635. https://doi.org/10.1109/TMECH.2023.3287635

[12] Zhao X. Multifeature video modularized arm movement algorithm evaluation and simulation. Neural Computing and Applications, 2023, 35(12): 8637-8646.

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, 2023, 21(2): 335-343. 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, 2024, 22(8): 2602-2612. 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, 2023, 41(2): 414-432. 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, 2023, 39(6): 4539-4551. 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, 2024, 12(2): 571-601. 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, 2023, 39(2): 1151-1163. 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, 2025, 68(9): 19204-19211. 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. IOP Publishing, 2025, 3019(1): 12020-12027. https://doi.org/10.1088/1742-6596/3019/1/012020