GCMA6D:Graph Convolution and Cross-Modality Attention Fusion for 6D Pose Estimation

Shihan Zhang[1], Ling Cao[2], Wei Pan[2,*], Lei Lu[1]
[1]Institute for Complexity Science, Henan University of Technology, Zhengzhou 450001, Henan, China
[2]Department of R&D, OPT Machine Vision Tech Co., Ltd., Dongguan 523860, Guangdong, China
zhangshihan@stu.haut.edu.cn, caoling@optmv.com, vpan@foxmail.com, lulei@haut.edu.cn
*Corresponding author: Wei Pan

Abstract

6D pose estimation plays a crucial role in object recognition and localization tasks in complex scenes. However, existing methods still show significant limitations when handling occluded or low-texture objects. To address these challenges, this paper proposes the GCMA6D method. In the point cloud branch, a 3DGCN-based geometric feature extraction module is designed, while the image branch incorporates Large Kernel Attention to achieve dual representation of the target. Furthermore, a Cross-Modality Attention mechanism is constructed to promote deep interaction between RGB features and geometric features. Combined with the Squeeze-and-Excitation module, this approach effectively enhances foreground regions while suppressing background interference, thereby improving the model's robustness in complex scenarios. Ablation experiments demonstrate that the three core modules significantly enhance performance on the LineMOD dataset, with an overall network accuracy of 97.7%. Comparative experiments further validate the effectiveness of the proposed method: compared with DenseFusion, the ADD-S metric on the LineMOD dataset increases by 3.4 percentage points, and by 0.8 percentage points on the YCB-Video dataset.

1 Introduction

Object 6D pose estimation aims to infer the three-dimensional rotation and translation of a target relative to the camera, and is crucial in applications such as robotic manipulation [1] and augmented reality [2]. However, achieving high accuracy and robustness remains challenging under low-texture conditions or partial occlusions.

Traditional methods are mainly feature-based or template-based. Feature-based methods rely on 2D–3D correspondences but struggle when keypoints are sparse, while template-based methods match rendered templates, with performance dropping under occlusion or appearance changes.

Deep learning has significantly advanced the field. RGB-based methods, such as PoseCNN [3] and Bb8 [4], achieve progress but lack depth information, limiting robustness in complex environments. RGB-D methods, such as DenseFusion [5], fuse RGB and point cloud features at the pixel level, improving accuracy, but rely on PointNet for geometric features, which may miss fine local details. Later methods, such as PVN3D [6], add local feature modeling but increase computational cost, limiting real-time use. Large-scale foundation models [7] offer strong generalization and robustness, yet their size and inference latency hinder real-time deployment.

To address these issues, we propose an efficient 6D pose estimation model. The method localizes the region of interest using the RGB-D mask, enhances texture features with large kernel attention in the image branch, and models local geometry using a 3D graph convolutional network. Cross-modal attention enables interaction between RGB and point cloud features, and a squeeze-and-excitation module emphasizes the foreground while suppressing background interference. Finally, RGB and geometric information are fused at the pixel level to produce the 6D pose. Experiments show that the method achieves competitive accuracy and robustness.

2 Related Works

2.1 RGB-Based Pose Estimation Methods

RGB-based 6D pose estimation methods can be primarily categorized into keypoint-based and direct regression approaches. Unlike traditional schemes that rely on handcrafted feature descriptors to extract keypoints, deep learning methods can learn more discriminative semantic keypoints. For instance, YOLO-6D predicts the 3D bounding box of a target based on the YOLO framework, projects it onto the 2D image plane, and then

estimates the 6D pose using the PnP algorithm.

In contrast, direct regression methods employ end-to-end networks to predict the 6D pose directly, without explicit keypoint correspondences. PoseCNN [3] performs semantic segmentation followed by regression of rotation and translation, achieving a compact architecture. Building upon this, DeepIM introduces iterative refinement, progressively optimizing the pose by minimizing the discrepancy between observed and rendered images. Furthermore, CosyPose [8] improves rotation parameterization to enhance training stability and achieves significant performance gains in multi-object scenarios.

Beyond keypoint-based and regression approaches, other RGB-based 6D pose estimation methods have also made progress. CRT-6D achieves real-time estimation via cascaded regression trees, balancing speed and accuracy, yet its performance degrades under low-resolution or complex backgrounds. FS6D [9] targets few-shot scenarios, predicting poses of novel categories with only a small number of support views, thereby greatly improving generalization. RNNPose [10] leverages temporal continuity in video sequences for more stable estimation, but incurs high computational cost, limiting its application in real-time or resource-constrained environments. Overall, RGB-based methods benefit from simple input requirements and fast inference, making them suitable for real-time applications; however, due to the lack of geometric information, their robustness and accuracy remain limited under occlusion, complex backgrounds, or low-texture conditions.

2.2 RGB-D-Based Pose Estimation Methods

With the advent of RGB-D sensors, models can access richer information. Early approaches often processed RGB and depth separately. For example, PoseCNN and YOLO-6D leveraged depth point clouds to refine predictions using ICP [11] or GICP [12]. Another line of methods directly concatenated RGB and depth features as network inputs [26], but this strategy often failed to fully exploit their complementary information. Subsequent research has focused more on cross-modal correlations. PointFusion [13], for instance, extracts features via CNN and PointNet and designs a fusion network to achieve deep integration. Recent methods further enhance fusion effectiveness: SurfEmb [14] jointly learns geometric and visual representations through surface embedding; MaskedFusion introduces an occlusion-aware mechanism for pixel-level fusion; CMFF6D [15] employs a multi-branch progressive fusion framework to integrate high-resolution RGB and depth geometric features, improving representation but incurring substantial computational overhead.

To address the limitations of RGB-D methods in exploiting local geometry under occlusion, low-texture, and complex backgrounds, this work proposes a point cloud feature extraction module based on graph convolution to strengthen neighborhood modeling, combined with cross-modal attention to facilitate deep interaction between RGB and point cloud features. This design enables the construction of a 6D pose estimation network that is both accurate and robust.

2.3 Attention Mechanisms

In large-scale or redundant-input models, attention mechanisms can guide the network to focus on critical information while suppressing irrelevant features. SENet [16] adaptively adjusts feature weights by modeling channel dependencies, whereas Cond-Conv employs dynamic convolution kernels to enhance model flexibility. In comparison, self-attention enables global interactions, capturing long-range dependencies and dynamically modulating feature importance, demonstrating superior performance in complex tasks. In the context of 6D pose estimation, SaMfENet [17] leverages self-attention to enhance multi-scale geometric representation of point clouds, significantly improving performance under occlusion and low-texture conditions. Inspired by this, CMA-Net applies self-attention for cross-modal fusion between images and point clouds, enhancing robustness while maintaining accuracy, enabling stable 6D pose estimation even in challenging scenarios.

The main contributions of this work are as follows:

- We propose an end-to-end RGB-D 6D pose estimation network that predicts the pose from a single RGB-D image through instance-level semantic segmentation.
- We introduce Large Kernel Attention and 3D Graph Convolutional Network (3DGCN) modules within the network architecture to enhance local and global feature modeling, thereby improving spatial structure understanding.
- We conduct a systematic analysis of pixel-level feature fusion methods and compare them with conventional approaches, demonstrating the critical role of cross-modal fusion in improving 6D pose estimation performance.

3 Method

3.1 Network Overview

Figure 1 illustrates the overall architecture of the proposed network, which consists of four main components:

Semantic segmentation module employs the PoseCNN semantic segmentation network to process input RGB images. It identifies bounding boxes of target objects and crops the relevant regions. The resulting masks are

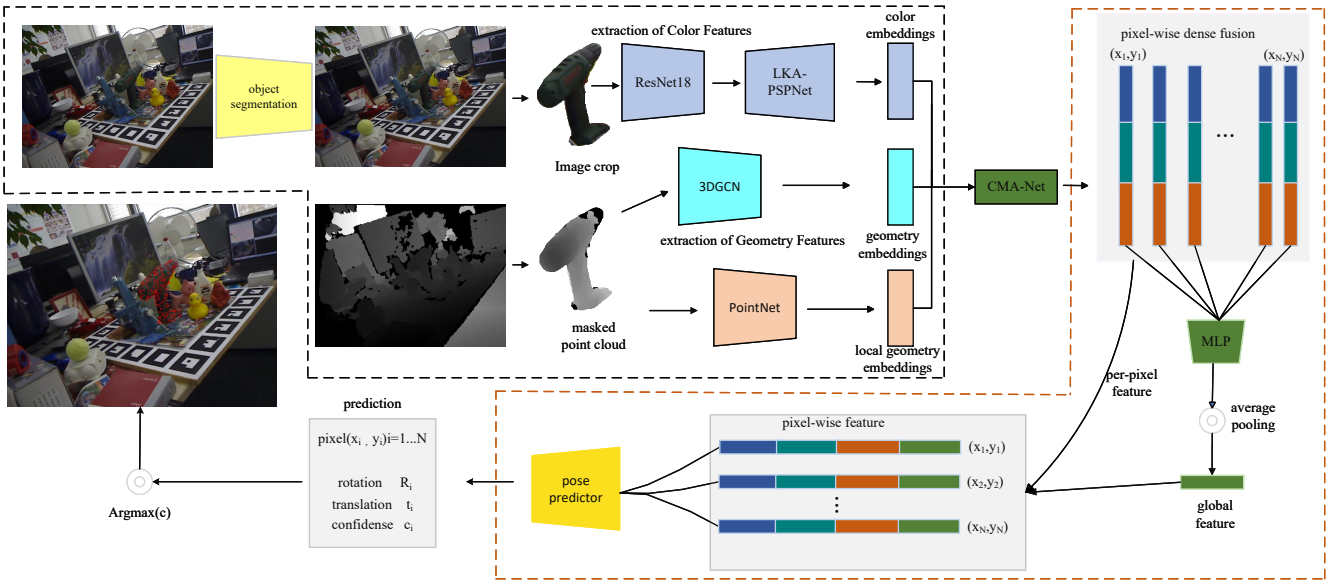

Figure 1: The overall architecture of GCMA6D consists of five components: performing semantic segmentation on the RGB-D image to obtain the target and its point cloud; feeding the image patch into the image feature module to extract color features; processing the point cloud with 3DGCN to enhance local geometric features; fusing pixel-level and global features and inputting them into the pose regression module to predict the object's rotation and translation, thereby achieving 6D pose estimation.

used to precisely locate objects in the depth map. Combining this information with camera intrinsics, depth map pixels are converted into corresponding 3D point clouds, providing an accurate and reliable foundation for subsequent image and point cloud feature extraction. This process effectively suppresses background interference, ensuring the pose estimation network focuses on target objects.

Point cloud feature extraction module consists of two branches, 3D-GCN and PointNet, which extract point-wise and global features from the segmented point cloud. PointNet processes unordered point clouds through multiple fully connected layers and obtains global features via global max pooling, capturing the overall geometric structure of the target object. To better represent the local structure of point clouds, a novel 3D graph convolutional network (3D-GCN) is introduced, combining learnable 3D graph convolution kernels with graph max pooling, effectively modeling local geometric relationships and enhancing the understanding of 3D spatial structures.

Image feature extraction module inputs cropped target regions into a ResNet18 backbone to extract basic color features and generate preliminary feature maps. These feature maps are further processed by a pyramid scene parsing network (LKA-PSPNet), which extracts rich color embedding features across multi-scale receptive fields and generates multi-resolution feature maps. Multi-scale representations enhance the semantic understanding of target regions, providing more precise image information for subsequent multi-modal feature fusion.

Feature fusion and pose estimation module fuses image and point cloud features at the pixel level and encodes them into a global feature representation, forming a multi-level, multi-modal composite feature. Fused features are input into the pose regression module, which directly predicts translation and rotation parameters of the target object, enabling accurate 6D pose estimation.

3.2 Semantic Segmentation

Semantic segmentation is a critical component of the network, providing essential data support for the backbone. At this stage, we adopt the semantic segmentation network from PoseCNN to process the input RGB images. The module first separates the target region from the background and crops the image to extract the object of interest. Through this process, we obtain RGB images and corresponding depth maps containing only the target object, thereby effectively suppressing background interference and allowing the subsequent pose estimation network to focus on key information. In addition to enabling precise fusion of image and point cloud data, the generated masks effectively filter out background noise, concentrate on the target, and significantly improve estimation accuracy under occlusion and low-texture conditions, as illustrated in Figure 2.

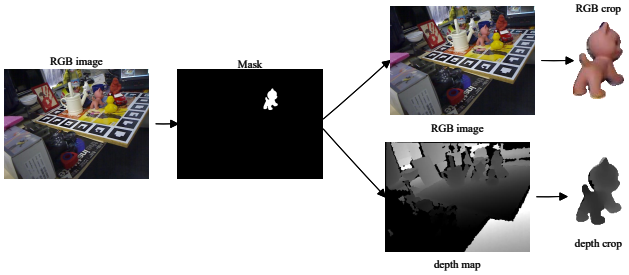

Figure 2: Semantic segmentation network. For each RGB image, N+1 binary masks are generated for objects and background; the masks are used to crop the image and extract the object region from the depth map, providing accurate input for feature extraction and pose estimation.

3.3 Point Cloud Feature Extraction Module

In traditional pose estimation methods, the complementary nature of RGB and depth information is often underutilized. To address this limitation, this study first generates the object's surface point cloud from the depth map and feeds it into the network for processing. Subsequently, geometric features of the point cloud are extracted using the PointNet and 3D-GCN modules. The process of geometric feature extraction is illustrated in Figure 3.

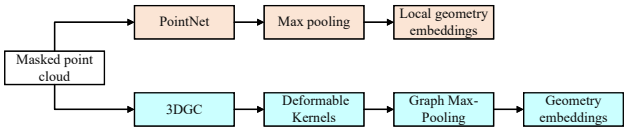

Figure 3: PointNet encodes 3D point clouds via fully connected layers and global max pooling, capturing only single-scale geometry. To address this, we introduce a 3D-GCN module that extracts features using learnable 3D graph convolutions on local neighborhoods, capturing directional information and local structures. Combined with graph max pooling, it generates multi-scale geometric features.

3.4 Image Feature Extraction Module

An image feature extraction network based on ResNet18 and LKA-PSPNet was constructed to extract deep features containing rich semantic information. Compared with PSPNet, LKA-PSPNet enhances the model's ability to capture local structures, textures, and edge information in the image while effectively modeling the local and global relationships of features after pyramid pooling, enabling the capture of key information at multiple scales.

Large Kernel Attention (LKA) is a novel linear attention mechanism that combines the strengths of convolutional operations and self-attention. It captures both local contextual information and long-range dependencies while avoiding the limitations of traditional self-attention in channel-wise adaptability, thereby improving the network's representation of multi-scale critical features. The structure of the LKA-PSPNet module is illustrated in Figure 4.

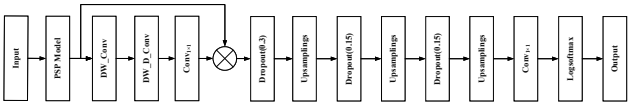

Figure 4: Structure of the LKA-PSPNet module. The input to the module is the preliminary feature map extracted from the image patch cropped by the bounding box, and the output is the refined feature map of the image patch.

The preliminary features are processed through the PSPModule (pyramid pooling module) to extract multi-scale features. The resulting features are then fed into the LKA module. Within this module, the features first pass through a depth-wise convolution layer (DW_Conv) that independently applies convolution to all 1024 input channels while maintaining the spatial dimensions of the feature map. Next, the feature map goes through a depth-wise dilated convolution layer (DW_D_Conv), which aligns the convolution kernel center with the input feature map center to effectively capture contextual information over a larger receptive field. A subsequent 1×1 pointwise convolution integrates the feature information. Finally, the original input feature map is multiplied element-wise with the generated attention map to produce the weighted feature representation, which serves as the module output.

During the output stage, the feature map gradually restores spatial resolution through three upsampling modules (using bilinear interpolation) and three Dropout layers. This design, combining pyramid pooling and attention mechanisms, effectively enhances the network's ability to capture multi-scale object features while preserving rich image information, thereby improving the robustness and accuracy of pose estimation.

3.5 Feature Fusion and Pose Estimation

Feature Fusion. In the fusion stage, pixel-level image features from the RGB branch are integrated with point-level geometric features from the depth branch. Leveraging the alignment between RGB and depth images, these features are concatenated at the pixel level to form a comprehensive representation that encodes geometry,

color, and local context. The fused features are then passed into the pose regression network for 6D pose prediction.

Point cloud features are extracted from depth maps that naturally correspond to RGB images. Based on this spatial alignment, pixel-wise concatenation is performed. The resulting representation captures geometric structure, visual appearance, and neighborhood relationships. To further refine this information, the fused features are processed by two shared-weight multilayer perceptrons (MLPs), followed by average pooling to obtain a compact global descriptor. Finally, the pixel-level fused features are combined with this global descriptor to generate the final feature embedding, which is fed into the pose estimation network to infer the 6D pose. The overall feature fusion procedure is illustrated in Figure 5.

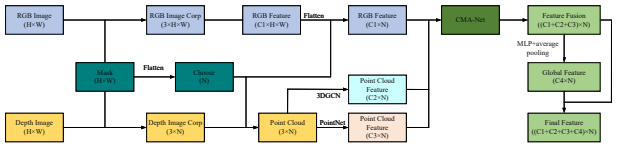

Figure 5: The target region is masked, and RGB and depth features are pixel-aligned. RGB features are extracted via CNN C1 ×N, point cloud features via PointNet and 3DGCN C2 ×N and C3 ×N. Features are concatenated, encoded with MLP, globally pooled, and combined with global features to form the final representation.

The CMA-Net module takes RGB features (F1), 3DGCN point cloud features (F2), and PointNet point cloud features (F3) as inputs and implements a multi-feature cross-attention weighting mechanism to efficiently fuse RGB and point cloud features.

Within this module, multimodal feature fusion is achieved through a combination of attention mechanisms and feature concatenation. Specifically, the three input features are mapped to the Query, Key, and Value components of the self-attention mechanism. A dot-product attention operation is then applied to generate the weighted feature representation. The self-attention mechanism captures global dependencies, enabling deep spatial and semantic interactions between RGB and point cloud features, thereby significantly enhancing the representational power of cross-modality feature fusion. The dot-product attention operation is computed as follows.

$$\text{Attention}(Q, K, V) = \text{Softmax}\Big(\frac{QK^\top}{\sqrt{d}}\Big)V \qquad (1)$$

In CMA-Net, two attention computations are used. The first enhances RGB features, and the second enhances point cloud features. Their outputs, together with the original point cloud feature, are fed into the SE module, which adaptively reweights channels. The reweighted outputs are then concatenated along the channel dimension to form the final multimodal representation, effectively fusing RGB and point cloud features while emphasizing critical channels.

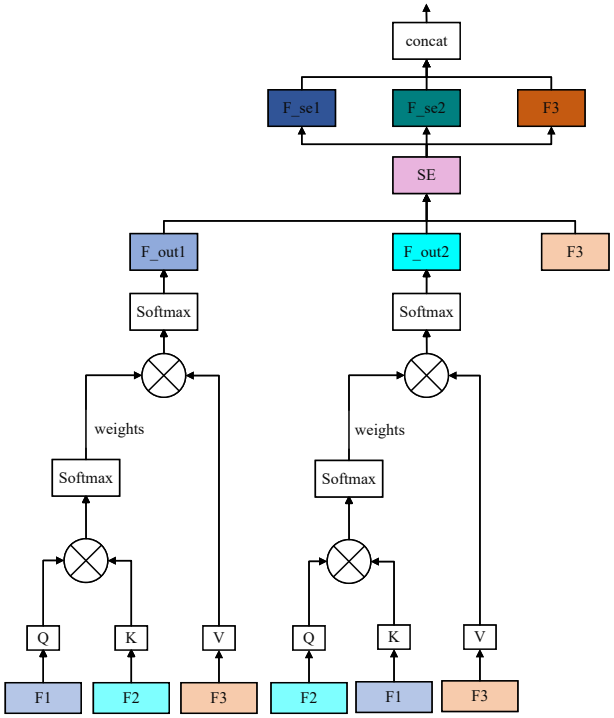

Figure 6: In CMA-Net, two attention computations are applied: one on (F1, F2, F3) to enhance RGB features, the other on (F2, F1, F3) to enhance point cloud features. Both outputs, along with F3, pass through the SE module, then all three are concatenated along the channel dimension to form the final multimodal representation.

Pose Estimation. After feature extraction and fusion, multimodal fused features are obtained for the sampled points. In the pose estimation stage, these fused features are fed into a neural network to regress the object's 6D pose parameters, including the rotation matrix R, translation vector t, and confidence score c.

The network loss function is computed as follows:

$$\text{Loss} = \frac{1}{N}\sum_{i=1}^{N}\left(L_i^p\, c_i - \omega \log(c_i)\right) \qquad (2)$$

In the equation, N represents the number of fused features; c_i denotes the confidence of the i-th fused feature

in regressing the pose, with higher confidence indicating more accurate pose parameters; ω is a balancing coefficient; L_i^p represents the pose loss of the i-th fused feature. For asymmetric objects, the pose loss is computed as follows:

$$L_i^p = \frac{1}{M} \sum_{j=1}^{M} \left\| (Rx_j + t) - (\hat{R}_i x_j + \hat{t}_i) \right\| \tag{3}$$

In this equation, M denotes the number of points randomly sampled from the target model; x_j is the j-th point in the sampled point cloud; $Rx_j + t$ represents the coordinates transformed by the ground-truth pose; $\hat{R}_i x_j + \hat{t}_i$ represents the coordinates transformed by the pose predicted from the i-th fused feature. This formula calculates the average distance between corresponding points under the ground-truth and estimated poses.

For objects with rotational symmetry, the object shape or texture does not define a unique canonical frame. Therefore, for symmetric objects, the pose loss is defined as the average distance between each point in the estimated transformed point cloud and its nearest point in the ground-truth transformed point cloud:

$$L_i^p = \frac{1}{M} \sum_{j=1}^{M} \min_{0<k<M} \left\| (Rx_j + t) - (\hat{R}_i x_k + \hat{t}_i) \right\| \tag{4}$$

After obtaining the initial pose estimation, a pose refinement process is typically performed to improve accuracy. Traditional ICP-based optimization methods are time-consuming and unsuitable for real-time applications. Following DenseFusion, we employ a CNN-based refinement approach consisting of three fully connected layers. The fused features are aggregated via a max-pooling layer to form a global feature, which is then used for iterative pose refinement. At each iteration, the network predicts a residual pose. After k iterations, the predicted residual poses are sequentially combined with the initial pose to obtain the final pose estimation:

$$[\hat{R} \,|\, \hat{t}] = [R_k \,|\, t_k] \cdot [R_{k-1} \,|\, t_{k-1}] \cdots [R_0 \,|\, t_0] \tag{5}$$

In this equation, $[R_k \,|\, t_k]$ represents the residual pose after the k-th iteration, and $[R_0 \,|\, t_0]$ denotes the initial pose output from the pose estimation network.

3.6 Evaluation Metrics

The accuracy of object pose estimation methods is evaluated using the average distance metrics ADD and ADD-S. For asymmetric objects, the ADD metric is used, which computes the mean distance between corresponding points of the object model transformed by the ground-truth pose $[R|t]$ and the estimated pose $[\hat{R}|\hat{t}]$. ADD is defined as follows:

$$\text{ADD} = \frac{1}{m} \sum_{x \in M} \left\| (Rx + t) - (\hat{R}x + \hat{t}) \right\| \tag{6}$$

In this equation, M denotes the set of points in the object model, x represents a point in M, and m is the number of sampled points.

For symmetric objects, the ADD-S metric is used, which computes the average distance between each point in the point cloud transformed by the estimated pose and its nearest neighbor in the point cloud transformed by the ground-truth pose:

$$\text{ADD-S} = \frac{1}{m} \sum_{x \in M} \min_{0<k<n} \left\| (Rx + t) - (\hat{R}x_k + \hat{t}) \right\| \tag{7}$$

A threshold of 10% of the object's diameter is set. If the average distance is below this threshold, the pose estimation is considered correct; otherwise, it is deemed incorrect. Accuracy is calculated as follows:

$$\text{Accuracy} = \frac{\text{Num}_{\text{pre}}}{\text{Num}_{\text{GT}}} \times 100\% \tag{8}$$

4 Experiments

4.1 Experimental Environment

The training and testing experiments were conducted on the following hardware and software environment: Intel i5-12600KF CPU, NVIDIA RTX 4060 Ti GPU, Ubuntu 20.04.4 operating system, Python 3.9, CUDA 11.8, cuDNN 9.1.0, and PyTorch 2.4.

The training parameters are set as follows: the initial learning rate is 0.0001 to accelerate model convergence and is dynamically adjusted using a learning rate decay strategy. The total number of training epochs is 500, with a batch size of 8. To improve data preprocessing efficiency, data loading is performed with 20 parallel threads. The learning rate decay is triggered when the loss reaches a threshold of 0.016, reducing the learning rate by 70% each time to gradually decrease the step size for parameter updates. The weight decay is initially set to 0.015 with a decay rate of 0.3 to mitigate overfitting and enhance generalization.

To evaluate the performance of the proposed method, experiments were conducted on two public datasets: LineMOD and YCB-Video. The LineMOD dataset is a common benchmark for 6D pose estimation, characterized by low-texture objects with complex backgrounds and varying illumination, making it effective for assessing improvements under challenging conditions. During training, 15% of the images were used for training, and the remaining 85% for testing. The YCB-Video dataset contains 21 objects with diverse shapes and textures, totaling 133,827 frames across 92 videos. Among them, 80

videos are used for training, and 2,949 key frames from the remaining 12 videos are used for testing. Additionally, the training set includes 80,000 synthetic images generated from public data. This dataset encompasses varying levels of occlusion, complex shapes, and diverse textures, providing challenging scenarios that help comprehensively evaluate the model's generalization ability and robustness.

4.2 Ablation experiments

All ablation experiments were conducted on the LineMOD dataset. The results of the ablation study are shown in Table 1, Accuracy is defined according to the LineMOD evaluation metric. DenseFusion serves as the baseline. The 3DGCN variant incorporates a graph convolution module into the original PointNet-based geometric feature extraction. CMA-Net introduces a cross-modality attention mechanism, and LKA refers to the PSPNet image feature extraction module enhanced with Large Kernel Attention.

Table 1: Comparison of Ablation Study Results on LineMOD. Accuracy (%) is reported.

3DGCN	CMA-Net	LKA	Accuracy (%)
			94.3
✓			96.9
	✓		95.4
		✓	96.2
✓	✓		97.2
✓		✓	97.4
	✓	✓	96.8
✓	✓	✓	97.7

4.3 Algorithm Comparison on the LineMOD Dataset

For the LineMOD benchmark, consistent with prior studies, we define a prediction as correct when the ADD-(S) distance is within 10% of the corresponding object's diameter. The accuracy, measured as the ratio of correctly estimated keyframes to the total number of keyframes, serves as the evaluation criterion.

Table 2 presents a comparison of our method with several representative RGB-based and RGB-D-based approaches on the LineMOD dataset, where bold numbers indicate the best performance and objects marked with an asterisk (*) denote symmetric objects. Among RGB-based techniques, HRPose, PoseCNN+ICP, and DPOD obtained average accuracies of 87.6%, 88.6%, and 95.1%, respectively. These results suggest that, even without depth input, competitive performance can be achieved by leveraging effective temporal modeling strategies.

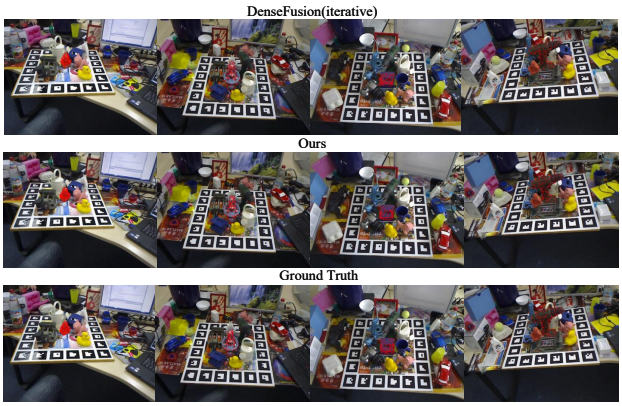

Figure 7: Comparison of pose estimation results for selected objects in the LineMOD dataset. Red points indicate the point cloud transformed according to the predicted object pose.

Figure 7 illustrates comparative results for selected categories. From a category-wise perspective, the proposed method performs exceptionally well on objects with occlusion or low texture, such as camera, duck, lamp, cat, and phone. This is primarily attributed to the graph convolution-based point cloud fusion module, which enhances local geometric detail extraction and spatial context modeling. For symmetric objects like eggbox and glue, the method achieves 100% accuracy, demonstrating robustness in handling symmetry-induced ambiguities.

In contrast, most existing algorithms struggle with low-texture objects such as ape, driller, and duck. RGB-based methods lack stable features, while improvements from depth information and iterative optimization are limited. Our method fully leverages the complementary information from RGB and depth, achieving superior results on low-texture objects.

4.4 Algorithm Comparison on the YCB-Video Dataset

For evaluating pose estimation on the YCB-Video dataset, we used two metrics: the area under the ADD-(S) curve (AUC) for thresholds between 0 and 10 cm, and the proportion of poses with an ADD-(S) error below 2 cm, which indicates high-precision predictions. In Table 3, the highest score for each object under both metrics is highlighted in bold, and objects with an asterisk (*) are symmetric and assessed accordingly.

Based on the AUC metric, our method achieved an average accuracy of 92.9% on the YCB-Video dataset, outperforming SaMfNet (92.6%), G2L-Net (92.4%), DenseFusion (91.2%), and MSCNet (91.5%). Compared with DenseFusion and MSCNet, this represents improvements of 1.7% and 1.4%, respectively, demonstrating stronger

Table 2: Performance comparison of various methods on the LineMOD dataset.

Object	HRPose	PoseCNN+ICP	DPOD	DenseFusion	QaQ	TMFNet	Ours
ape	61.2	77.0	87.5	92.3	90.3	93.4	**96.4**
benchvise	95.5	97.5	97.5	93.2	94.3	94.3	**97.8**
camera	84.9	93.5	96.3	94.4	96.8	97.1	**98.1**
can	93.6	96.5	**99.6**	93.1	95.6	97.2	97.5
cat	86.0	82.1	94.3	96.5	95.8	97.7	**98.0**
driller	96.2	95.0	95.5	87.0	90.0	96.3	**96.5**
duck	68.0	77.7	88.0	92.3	92.1	93.6	**94.8**
eggbox*	99.0	97.1	99.9	99.8	**100.0**	99.9	**100.0**
glue*	97.0	99.4	97.8	**100.0**	**100.0**	99.7	**100.0**
holepuncher	78.1	52.8	88.9	92.1	93.0	95.4	**96.8**
iron	95.5	98.3	**99.8**	97.0	97.9	97.2	98.5
lamp	96.6	97.5	96.7	95.3	96.9	98.0	**98.4**
phone	86.7	87.7	95.0	92.8	96.5	97.6	**97.8**
MEAN	87.6	88.6	95.1	94.3	95.3	96.8	**97.7**

Note: Bold values represent the top-performing results. Objects with an asterisk (*) denote symmetric cases.

pose regression capability. Under the <2cm accuracy measure, our method reached 95.7%, surpassing SaMfNet (95.6%), MSCNet (93.9%), and DenseFusion (95.3%).

Overall, our method performs well across both metrics and object categories, showing strong generalization and robustness. However, as shown in Table 3, all methods, including ours, exhibit lower accuracy on large_clamp and extra_large_clamp, primarily due to limitations in PoseCNN semantic segmentation: the two types of clamps are visually very similar, making it difficult to generate precise masks, which in turn affects pose estimation. Nevertheless, our method achieves significant improvements in overall 6D pose estimation accuracy, particularly under the <2cm measure, and demonstrates enhanced robustness and precision compared with other RGB-D methods.

To visually compare the predictions of different algorithms on YCB-Video, we rendered the estimated results of each method. All methods use PoseCNN output for semantic segmentation; the predicted poses are combined with sampled point information to generate point clouds, which are then projected back onto the original images to directly observe the consistency between estimated poses and ground truth. The comparative results are shown in Figure 8.

5 Conclusion

This paper presents an end-to-end 6D pose estimation network based on RGB-D images. The network remains robust under severe occlusion, complex backgrounds, and low-texture conditions. Ablation studies show that graph convolutional feature extraction and multimodal feature fusion significantly improve pose estimation accuracy and local feature perception. While limitations exist for highly symmetric or heavily occluded objects, the method demonstrates potential for optimization in feature representation, occlusion handling, and cross-modal fusion. Future work will focus on improving background-masked region processing and exploring few-shot or zero-shot applicability to enhance its use in robotic tasks.

References

[1] Y. Yu, H. Xie, K. Zhang, Y. Wang, Y. Li, J. Zhou, and L. Xu. Design, development, integration, and field evaluation of a ridge-planting strawberry harvesting robot. *Agriculture*, 14:2126, 2024.

[2] A. Firintepe, A. Pagani, and D. Stricker. A comparison of single and multi-view ir image-based ar glasses pose estimation approaches. In *Proceedings of the 2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pages 571–572, Lisbon, Portugal, March 27–April 1 2021.

[3] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv*, 1711.00199, 2017.

[4] M. Rad and V. Lepetit. Bb8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth. In *Proceedings of the IEEE Inter-*

Table 3: Performance comparison of various algorithms on the YCB-Video dataset.

Object	SaMfNet [17]		DenseFusion [5]		MSCNet [23]		G2L-Net [19]	
	AUC	< 2cm	AUC	< 2cm	AUC	< 2cm	AUC	< 2cm
002_master_chef_can	**95.9**	**100.0**	95.2	**100.0**	94.7	96.0	94.0	-
003_cracker_box	94.1	94.8	92.5	**99.3**	93.4	89.4	88.7	-
004_sugar_box	97.2	**100.0**	95.1	**100.0**	95.7	**100.0**	96.0	-
005_tomato_soup_can	94.2	96.8	93.7	96.9	94.1	96.7	86.4	-
006_mustard_bottle	95.4	**100.0**	95.9	**100.0**	95.3	**100.0**	95.9	-
007_tuna_fish_can	94.7	**100.0**	94.9	**100.0**	95.5	98.3	96.0	-
008_pudding_box	95.9	**100.0**	94.7	**100.0**	94.4	99.0	93.5	-
009_gelatin_box	97.3	**100.0**	95.8	**100.0**	97.6	**100.0**	96.8	-
010_potted_meat_can	91.2	94.1	90.1	93.1	90.0	89.6	86.2	-
011_banana	97.3	**100.0**	91.5	93.9	90.3	94.5	96.3	-
019_pitcher_base	96.7	**100.0**	94.6	**100.0**	93.9	**100.0**	91.8	-
021_bleach_cleanser	95.5	99.6	94.3	99.8	94.7	99.3	92.0	-
024_bowl*	89.5	93.0	86.6	69.5	**93.1**	93.1	86.7	-
025_mug	97.1	**100.0**	95.5	**100.0**	94.4	**100.0**	95.4	-
035_power_drill	96.0	99.6	92.4	97.1	91.7	99.3	95.2	-
036_wood_block*	91.1	**99.2**	85.5	93.4	90.3	98.8	86.2	-
037_scissors	84.1	67.4	**96.4**	**100.0**	87.6	59.7	85.0	-
040_large_marker	97.0	**99.9**	94.7	99.2	96.1	99.7	96.8	-
051_large_clamp*	73.8	77.5	71.6	78.5	71.6	76.1	**94.4**	-
052_extra_large_clamp*	67.6	66.1	69.0	69.5	68.2	61.7	**92.3**	-
061_foam_brick*	95.5	**100.0**	92.4	**100.0**	95.1	**100.0**	94.9	-
MEAN	92.6	95.6	91.2	95.3	91.5	93.9	92.4	-

Note: Bold values represent the top-performing results. Objects with an asterisk (*) denote symmetric cases.

Figure 8: Visual comparison of estimated poses. Various colors and shapes represent the same object visualized as point clouds using different methods.

national Conference on Computer Vision (ICCV), pages 3828–3836, Venice, Italy, October 22–29 2017.

[5] C. Wang, D. Xu, Y. Zhu, R. Martín-Martín, C. Lu, L. Fei-Fei, and S. Savarese. Densefusion: 6d object pose estimation by iterative dense fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3343–3352, Long Beach, CA, USA, June 15–20 2019.

[6] Y. He, W. Sun, H. Huang, J. Liu, H. Fan, and J. Sun. Pvn3d: A deep point-wise 3d keypoints voting network for 6dof pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11632–11641, Seattle, WA, USA, June 13–19 2020.

[7] T. Hodan, M. Sundermeyer, Y. Labbe, V.N. Nguyen, G. Wang, E. Brachmann, B. Drost, V. Lepetit, C. Rother, and J. Matas. Bop challenge 2023 on detection, segmentation and pose estimation of seen and unseen rigid objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5610–5619, Seattle, WA, USA, June 16–22 2024.

[8] Y. Labbé and et al. Cosypose: Consistent multi-view multi-object 6d pose estimation. In *European Conference on Computer Vision (ECCV)*, Cham, Switzerland, 2020. Springer International Publishing.

[9] Y. He, Y. Wang, H. Fan, J. Sun, and Q. Chen. Fs6d: Few-shot 6d pose estimation of novel objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6814–6824, New Orleans, LA, USA, June 18–24 2022.

[10] Y. Xu, K.Y. Lin, G. Zhang, X. Wang, and H. Li. Rnnpose: Recurrent 6-dof object pose refinement with robust correspondence field estimation and pose optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14880–14890, New Orleans, LA, USA, June 18–24 2022.

[11] P.J. Besl and N.D. McKay. Method for registration of 3-d shapes. In *Proceedings of Sensor Fusion IV: Control Paradigms and Data Structures*, volume 1611, pages 586–606, Boston, MA, USA, November 12–15 1991.

[12] A. Segal, D. Haehnel, and S. Thrun. Generalized-icp. In *Proceedings of the Robotics: Science and Systems (RSS)*, volume 2, page 435, Seattle, WA, USA, June 28–July 1 2009.

[13] D. Xu, D. Anguelov, and A. Jain. Pointfusion: Deep sensor fusion for 3d bounding box estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 244–253, Salt Lake City, UT, USA, June 18–23 2018.

[14] R.L. Haugaard and A.G. Buch. Surfemb: Dense and continuous correspondence distributions for object pose estimation with learnt surface embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6749–6758, New Orleans, LA, USA, June 18–24 2022.

[15] Z. Han, L. Chen, and S. Wu. Cmff6d: Cross-modality multiscale feature fusion network for 6d pose estimation. *Neurocomputing*, 623:129416, 2025.

[16] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7132–7141, Salt Lake City, UT, USA, June 18–23 2018.

[17] Z. Li, X. Li, S. Chen, J. Du, and Y. Li. Samfenet: Self-attention based multi-scale feature fusion coding and edge information constraint network for 6d pose estimation. *Mathematics*, 10:3671, 2022.

[18] Q. Guan, Z. Sheng, and S. Xue. Hrpose: Real-time high-resolution 6d pose estimation network using knowledge distillation. *Chinese Journal of Electronics*, 32:189–198, 2023.

[19] W. Chen, X. Jia, H.J. Chang, J. Duan, and A. Leonardis. G2l-net: Global to local network for real-time 6d pose estimation with embedding vector features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4233–4242, Seattle, WA, USA, June 13–19 2020.

[20] S. Zakharov, I. Shugurov, and S. Ilic. Dpod: 6d pose object detector and refiner. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[21] T. Petitjean and et al. Qaq: Robust 6d pose estimation via quality-assessed rgb-d fusion. In *2023 18th International Conference on Machine Vision and Applications (MVA)*. IEEE, 2023.

[22] W. Zhou and et al. Tmfnet: Three-input multilevel fusion network for detecting salient objects in rgb-d images. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 6(3):593–601, 2021.

[23] F. Gao, Q. Sun, S. Li, W. Li, Y. Li, J. Yu, and F. Shuang. Efficient 6d object pose estimation based on attentive multi-scale contextual information. *IET Computer Vision*, 16:596–606, 2022.