ITC 4/54
Information Technology and Control
Vol. 54 / No. 4/ 2025
pp. 1428-1458
DOI 10.5755/j01.itc.54.4.40202

MSPF-LMFF: Category-Level 6D Object Pose Estimation via Multi-Scale Prior Point Cloud Fusion and Lightweight Multi-Feature Fusion

Received 2025/01/16 | Accepted after revision 2025/07/13

HOW TO CITE: Cao, P., Weng, T., Han, Q., Ye, P., Gao, L., Han, C., Tian, Y. (2025). MSPF-LMFF: Category-Level 6D Object Pose Estimation via Multi-Scale Prior Point Cloud Fusion and Lightweight Multi-Feature Fusion. *Information Technology and Control*, 54(4), 1428-1458. https://doi.org/10.5755/j01.itc.54.4.40202

MSPF-LMFF: Category-Level 6D Object Pose Estimation via Multi-Scale Prior Point Cloud Fusion and Lightweight Multi-Feature Fusion

Peng Cao, Tengfei Weng*, Qi Han, Peng Ye, Long Gao, Cong Han

School of Computer Science and Engineering (School of Artificial Intelligence), Chongqing University of Science and Technology, Chongqing 401331, China

Yuan Tian

Department of Computer Science, Michigan Technological University, Michigan 49931, USA

Corresponding author: wengtf_cq@163.com

Object pose estimation is a critical task in the field of machine vision. Existing pose estimation methods often suffer from challenges such as large parameter sizes, complex architectures, and high computational costs, which limit their applica-bility in real-world scenarios. To address these issues, we propose a novel category-level object pose estimation model, named MSPF-LMFF. This model eliminates the reliance on attention mechanisms or precise 3D models, significantly re-duces computational complexity, and enhances pose estimation accuracy, demonstrating superior performance on both real and synthetic datasets. Specifically, the MSPF module enriches the features of point clouds by integrating multi-scale image texture features with prior point cloud features, making them closer to the target object point cloud. Subsequently, the LMFF module combines geometric features of fused point cloud, depth image features, and geometric features of the target object point cloud to enhance the robustness of the model. At the same time, this module fuses adaptive point cloud features with the target object's geometric features to improve the reliability of shape informa-

tion, thereby enhancing the model's generalization capability across different instances of the same category. Following this, a multi-layer perceptron (MLP) generates deformation and mapping matrices to reconstruct the target object's normalized object coordinate space (NOCS) model. Finally, based on the NOCS model, the point cloud registration module computes the target object's 6D pose and 3D dimensions. Experimental results demonstrate that MSPF-LMFF outperforms existing methods on the NOCS-REAL and NOCS-CAMERA datasets while significantly reducing parameter sizes and training time. Moreover, the proposed model exhibits exceptional generalization capabilities on the Wild 6D dataset, further validating its effective-ness. The code is open-sourced at https://github.com/caopenging/MSPF-LMFF.git.

KEYWORDS: Feature Fusion, Fusion Point Cloud, Lightweight Model, Multi-Scale Feature Fusion, Lightweight Multi-Feature Fusion.

1. Introduction

6D object pose estimation is a core research problem in the field of computer vision, with wide applications in radar and hyperspectral data processing [43], medical image analysis (such as brain tumor detection) [35], and intelligent transportation systems (such as traffic sign recognition and autonomous driving) [49]. The primary goal is to predict the 6D rigid transformation between the object coordinate system and the camera coordinate system based on given observation data (such as RGB images, RGB-D images, or point cloud data). To achieve high-precision object pose estimation in complex environments or occlusion scenarios, researchers have proposed various instance-level object pose estimation methods, which have demonstrated significant success in handling complex scenes and oc-clusions [30], [2], [13], [42], [5], [14], [32], [18], [37], [12], [38], [25], [26]. However, instance-level methods typically rely on precise 3D models and surface texture information of objects. When the target object lacks a corresponding 3D model or distinctive texture features, the accuracy of pose estimation declines significantly, limiting the broader applicability of these methods in real-world scenarios.

To address this issue, Wang et al. [39] proposed a category-level object pose estimation method. Unlike instance-level approaches, category-level methods exhibit stronger generalization capabilities, enabling the prediction of the poses of previously unseen instances within the same category, thereby expanding their applicability. This method employs the normalized object coordinate space (NOCS) to provide a unified representation of different object instances within a category, serving as a reference for object pose prediction. However, category-level methods typically lack detailed 3D model information for specific instances and primarily rely on extracting key geometric features from the object's 3D point cloud. Nonetheless, NOCS struggles to accurately capture the intra-category shape variations, leading to reduced estimation accuracy.

To address this challenge, researchers have introduced several shape prior-based methods [3], [52], [28], [45], [27], [47]. While these approaches have made notable progress in enhancing accuracy, they often perform suboptimally when handling point clouds of objects with complex geometric structures, such as cameras, particularly at object edges. This limitation adversely affects the precision of pose estimation.

To further address these challenges, researchers have also proposed several prior-free methods [20], [4], [6], [48], [24], [21], [50], which directly regress object poses to achieve better real-time performance during inference. However, these methods still face certain limitations. Due to the absence of prior point clouds, the extracted features often exhibit significant discrepancies from the target object's point cloud features, thereby affecting the accuracy of pose estimation.

For instance, CD-POSE [51] estimates object poses by extracting features from both prior and target object point clouds using a single-scale extraction strategy. However, this approach presents notable limitations in practical object pose estimation tasks. Even when prior point clouds undergo comprehensive feature extrac-tion, substantial inconsistencies

may still exist between them and the target object's point cloud, ultimately compromising pose estimation performance in real-world scenarios. This issue becomes even more pronounced in complex or dynamically changing environments, where texture errors further degrade estimation accuracy.

Therefore, effectively reducing the feature discrepancy between prior and target point clouds and enhancing model performance in real-world applications remain critical challenges that need to be addressed in this field.

To this end, we propose a Multi-Scale Prior Point Cloud Fusion Module (MSPF). This module significantly enhances the feature representation capability of prior point clouds by integrating multi-scale image texture features with prior point cloud features. By performing pixel-wise aggregation of texture and geometric features at multiple scales, MSPF generates more ac-curate prior point cloud feature representations, which are more consistent with the pose of the target object.

Compared to CD-POSE, which relies solely on a single-scale feature extraction strategy, the proposed multi-scale information fusion approach demonstrates higher robustness and accuracy in handling complex environments and unseen objects. This method provides a more effective solution for real-world applications, ensuring improved pose estimation performance under challenging conditions.

Recently, several methods based on attention mechanisms or Transformer architectures have been proposed for object pose estimation. For instance, CR-Net [40] achieves accurate category-level 6D pose estimation through a cascaded relational and recursive reconstruction network. SGPA [3] enhances 6D object pose estimation performance by dynamically adjusting the structural similarity between shape priors and observed instances, ensuring better alignment with the target object.

AG-Pose [22] leverages four attention modules to adaptively detect sparse keypoints, effectively representing the geometric structures of different instances. This enhances instance feature extraction capability, leading to improved pose estimation accuracy. CD-Pose captures global geometric features in point clouds using a self-attention mechanism, making it particularly suitable for handling objects with complex structures.

Furthermore, GPT-COPE utilizes self-attention mechanisms to learn multi-scale geometric features from observed point clouds. It employs a graph-guided point transformer to extract these features and integrates them using an iterative non-parametric decoder, thereby achieving more precise pose estimation.

However, CD-Pose, GPT-COPE, and AG-Pose all rely on Transformer architectures, resulting in large model parameters and slow inference speeds, which limit their performance in real-time applications.

To address the limitations of existing methods in category-level object pose estimation, this paper proposes a novel MSPF-LMFF network, which integrates four key components: a Multi-Scale Prior Point Cloud Fusion Module, a Lightweight Multi-Feature Fusion Module, a Normalized Object Coordinate Space (NOCS) Model Reconstruction Module, and a Point Cloud Registration Module. Existing approaches often struggle with accurately capturing multi-scale texture and geometric information, leading to suboptimal pose estimation accuracy and robustness. The proposed MSPF-LMFF network addresses these challenges by effectively fusing image and point cloud features at multiple scales, thereby enriching the feature representation of point clouds and enhancing the model's robustness.

First, the Multi-Scale Prior Point Cloud Fusion Module (MSPF) integrates prior point cloud features and image texture features at multiple scales, leveraging residual networks and PointFeatureNet to extract and aggregate multi-scale features. This fusion process ensures that the prior point cloud closely resembles the target object's shape, significantly improving pose estimation accuracy. Second, the Lightweight Multi-Feature Fu-sion Module (LMFF) enhances the model's robustness by cross-fusing and concatenating shape prior features with target object point cloud features. This fusion em-beds the geometric information of the fused point cloud into the target object point cloud, improving ro-bustness, while simultaneously refining the accuracy and reliability of prior point cloud features.

Subsequently, the NOCS Model Reconstruction Module generates deformation and mapping matrices to restore object shapes and establish dense correspondences with NOCS coordinates. Finally, the Umeyama algorithm [36] is employed to compute

the optimal similarity transformation between the observed point cloud and the reconstructed NOCS coordinates, thereby obtaining the 6D object pose and 3D object size.

By integrating and fusing image and point cloud features at multiple scales, the MSPF and LMFF modules effectively capture multi-scale texture and geometric information of objects, thereby enriching the feature representation of the point cloud. The LMFF module not only enhances the perception of object shape and texture features but also eliminates the need for attention mechanisms while generating high-quality NOCS coordinate representations, significantly improving object pose estimation accuracy.

Experimental results demonstrate that the proposed MSPF-LMFF network achieves significantly higher accuracy than existing methods on the NOCS-REAL and NOCS-CAMERA datasets, with inference speeds reaching 20 fps and 23 fps, respectively. Moreover, the network exhibits exceptional performance and strong generalization capability in category-level object pose estimation tasks. The evaluation on the Wild 6D dataset further validates its reliability and practicality.

Our Key Contributions Are Summarized as Follows:

1. We propose a category-level pose estimation network (MSPF-LMFF) based on multi-scale prior point cloud fusion and lightweight multi-feature fusion. The network leverages residual networks and PointFeatureNet to enrich prior point cloud features and performs cross-fusion of fused point cloud features and target object point cloud features, thereby enhancing the geometric robustness of the prior point cloud.
2. We introduce the Multi-Scale Prior Point Cloud Fusion Module (MSPF), which first extracts multi-scale features from images using a residual network and then employs PointFeatureNet to extract multi-scale geometric features from the prior point cloud. These features are subsequently aggregated at corresponding scales through pixel-wise addition. By fusing texture and geometric features, the prior point cloud acquires rich multi-scale information, making it more closely resemble the target object's shape, thereby improving pose estimation accuracy.
3. We design a Lightweight Multi-Feature Fusion Module (LMFF), which cross-fuses the current point cloud features with prior point cloud features, enhancing the fused point cloud's perception of geometric and texture information. This improves the robustness and generalization ca-pability of the model.
4. Our method achieves higher accuracy than existing approaches on the NOCS-REAL and NOCS-CAMERA datasets, while maintaining inference speeds of 20 fps and 23 fps, respectively. Additionally, our approach significantly reduces parameter complexity, demonstrating higher efficiency and lightweight advantages compared to state-of-the-art methods.

The structure of this paper is as follows: Section 2 provides an overview of related research on 6D object pose estimation, categorizing existing approaches into instance-level methods (Section 2.1) and category-level methods (Section 2.2).

Section 3 presents a detailed description of the proposed MSPF-LMFF model, including its technical details (Sections 3.1-3.5) and a description of the loss functions used for training (Section 3.6). Section 4 reports the experimental results, covering the experimental setup (Section 4.4) and performance comparisons with state-of-the-art methods on the NOCS-CAMERA (Section 4.5), NOCS-REAL (Section 4.6), and WILD6D (Section 4.7) datasets. The results demonstrate that our method outperforms existing approaches across multiple evaluation metrics. Additionally, this section includes a comparative analysis across object categories (Section 4.8), an ablation study (Section 4.9), and an evaluation of runtime performance and parameter efficiency (Section 4.10). Section 5 concludes the paper by summarizing the findings.

2. An Overview of Related Research on 6D Object Pose Estimation

2.1. Instance-Level Object Pose Estimation

Instance-level object pose estimation methods are trained on known objects [22] and can be primarily categorized into three types: correspondence-based methods, template-based methods, and direct re-

gression-based methods. Correspondence-based methods can be further divided into 2D-3D correspondence and 3D-3D correspondence. The 2D-3D correspondence methods [30], [2] define keypoints between RGB images and CAD models of objects, train models to predict 2D keypoints, and solve the object pose using perspective algorithms. The 3D-3D correspondence methods [13], [42] directly define keypoints on CAD models and use observed point clouds to predict predefined 3D key-points, followed by applying least-squares algorithms to solve the object pose. However, most correspondence-based methods heavily rely on rich texture information, and their performance may degrade when applied to textureless objects. Template-based methods primarily rely on point cloud registration [5], [14]. Here, the template is a CAD model of the object in a canonical pose, and the goal of these methods is to find the optimal relative pose that aligns the observed point cloud with the template. Additionally, there are RGB-based template methods [32], [18], which require collecting and annotating images of the object from different viewpoints during the training phase to create templates. Subsequently, these methods train a template matching model to find the template that best matches the observed image and use the template's pose as the actual pose of the object. Overall, template-based methods can be effectively applied to textureless objects, but the template matching process is often computationally intensive and time-consuming. With the rapid development of deep learning techniques, direct regression-based methods [37], [12], [38], [25], [26] have gained increasing attention in recent years. These methods use the ground truth object pose as supervi-sion signals to train models for end-to-end pose regression. DenseFusion [37] fuses RGB and depth fea-tures and proposes a pixel-wise dense fusion network for pose regression. FFB6D [12] further designs a bidirectional feature fusion network to fully integrate RGB and depth features. GDR-Net [38] introduces a geometry-guided network for end-to-end monocular object pose regression. HFF6D [25] designs a hierarchical feature fusion framework suitable for object pose tracking in dynamic scenes. Although instance-level methods exhibit excellent accuracy, they are limited to fixed instances and are only applicable to specific objects seen during training.

2.2. Category-Level Object Pose Estimation

In recent years, category-level methods have garnered significant attention in the field of object pose estimation, primarily due to their ability to generalize to unseen objects within the same category, thereby enhancing their practical applicability. NOCS [39] introduced the Normalized Object Coordinate Space, providing a standardized representation for objects within the same category and utilizing the Umeyama algorithm to recover object poses. SPD [33] proposed a shape prior deformation mechanism to address the challenge of sig-nificant intra-class shape variations. Given the notable performance advantages of SPD, subsequent research has proposed more shape prior-based methods. For instance, CR-Net [40] designed a recurrent framework to enhance pose estimation capabilities through iterative residual optimization, achieving a coarse-to-fine pose estimation process. SGPA [3] dynamically adjusts shape priors by computing structural similarity between shape priors and observed instances, thereby adapting to different object instances. 6D-ViT [52] incorporated Transformer architectures, introducing Pixelformer and Pointformer networks to extract more refined object features. STG6D [28] further fused feature differences between shape priors and observed objects, enabling more precise shape deformation. RBP-Pose [45] proposed a geometry-guided residual object bounding box projection network to address the issue of insufficient pose-sensitive feature extraction. CATRE [27] refined poses by aligning observed point clouds with shape priors, which can be used to further optimize pose estimation results from the aforementioned methods. GeoReF [47] built upon CATRE [27] by introducing hybrid scope layers and learnable affine transformations to handle geometric variations. Despite the significant progress made by these shape prior-based methods in pose estimation, they still exhibit certain limitations. First, the process of constructing CAD model libraries is cumbersome and time-consuming, particularly when dealing with geometrically complex prior point clouds, leading to suboptimal results. Additionally, these methods may fail to produce accurate results when object surface textures change. Meanwhile, some prior-free methods have also gained attention. DualPoseNet [20] introduced a dual pose encoder, employing two par-

allel decoding paths for pose regression to enhance pose consistency learning. FS-Net [4] proposed a shape-based 3D graph convolution network to separately regress translation, rotation, and scale information. GPV-Pose [6] adopted a geometry-guided point-wise voting approach to strengthen the learning of category-level pose-sensitive features. HS-Pose [48] proposed a hybrid scope feature extraction network to address the limitations of 3D graph convolution networks in terms of size and displacement invariance. IST-Net [24] explored the necessity of shape priors in category-level pose estimation and proposed a prior-free method based on latent space transformation. VI-Net [21] improved estimation accuracy by decoupling rotation into viewpoint and inplane rotation, addressing the instability in rotation estimation. Diff9D, on the other hand, eliminates the need for any 3D shape priors during training or inference by employing a denoising diffusion implicit model (DDIM) to enhance pose estimation accuracy, although its precision remains lower compared to methods utilizing prior point clouds.

2.3. Transformer-Based Object Pose Estimation

In recent years, attention-based methods have been widely applied to object pose estimation. GPT-COPE [50] leverages a graph-guided point attention mechanism to extract geometric features of point clouds from local to global levels. CD-Pose [51] employs a geometry consistency and geometry difference learning framework, combining self-attention mechanisms with depth images to achieve category-level 6D pose estimation. Although depth images provide rich geometric information, relying solely on them may overlook important visual information in RGB images.

Lin et al. proposed AG-Pose [22], which includes two key designs: first, an instance-adaptive keypoint detection mechanism utilizes four attention modules to adaptively detect a set of sparse keypoints representing the geometric structures of different instances; second, a geometry-aware feature aggregation module effectively integrates local and global geometric information into keypoint features. These two modules work collaboratively to establish robust keypoint-level correspondences for unseen instances, thereby enhancing the model's generalization capability. Despite the performance improvements achieved by these attention-based and Transformer methods, they suffer from high computational complexity, large parameter counts, and the need for further exploration of multi-modal feature fusion. Currently, only a few studies [44] have focused on the application of multimodal feature fusion in category-level 6D pose estimation.

Our work is based on multimodal feature fusion for category-level object pose estimation. Specifically, we adopt the shape prior deformation step introduced in [50], [51] to reconstruct the Normalized Object Coordinate Space (NOCS) representation. However, unlike [50], [51], we do not rely on attention mechanisms or Transformer architectures. Instead, we enhance the model's robustness and the reliability of shape priors through lightweight multi-feature fusion, thereby improving the model's generalization ability across different instances of the same category. Experimental results demonstrate that the proposed method significantly outperforms existing attention-based or Transformer-based methods, particularly on the NOCS-REAL dataset, where a notable improvement in accuracy is achieved.

2.4. Lightweight-Based Object Pose Estimation

Zhang et al. [46] proposed a lightweight network for pose estimation. This method employs a two-stage refinement training strategy: first, an efficient skeleton detection network is used to obtain an initial pose estimation, and then a refinement module is applied in the second stage to further optimize keypoint detection results. This approach significantly improves keypoint detection accuracy while reducing computational complexity. Yang et al. [44], on the other hand, enhanced the accuracy of category-level 6D object pose estimation by eliminating RGB features, optimizing geometric information extraction, and designing a lightweight feature fusion encoder. Additionally, this method reduces the number of model parameters while increasing inference speed to 32 fps, making it more suitable for resource-constrained devices and real-time applications. However, this network fails to effectively capture the feature differences between the prior point cloud and the current point cloud, resulting in lower pose estimation accuracy. To address this limitation, we

introduce a Prior Adaptation module into the lightweight network and incorporate a point cloud feature subtraction operation to explicitly highlight the feature differences between the two. This enhances the model's sensitivity to point cloud variations, thereby improving the stability and accuracy of pose estimation.

2.5. Loss Function-Based Object Pose Estimation

In 6D pose estimation, geometric loss functions play a crucial role. Traditional loss functions, such as Cham-fer Distance (CD) and Earth Mover's Distance (EMD), offer certain advantages in handling geometric errors but also exhibit limitations. To improve the accuracy and stability of models, many studies have introduced more complex loss terms to optimize point cloud alignment and deformation. Chamfer Distance (CD) is the most commonly used point cloud alignment loss function, which calculates the distance between each point in one point cloud to its nearest point in another. It is widely used in point cloud matching and reconstruction tasks. However, since CD only considers the distance between nearest points, it may fail to fully capture the global geometric structure and details of point clouds, particularly under incomplete or missing point cloud conditions. Earth Mover's Distance (EMD) measures the overall structural differences between point clouds, providing a better representation of geometric shapes and distributions. However, its computational cost is high, making it less efficient for real-time applications, especially when dealing with large-scale point clouds. ICP Loss, a classic point cloud alignment algorithm, computes rigid transformations by minimizing the Euclidean distance between point clouds. However, ICP is primarily suitable for rigid transformations and performs poorly in handling occlusions or missing points, with limited capability in optimizing deformations. Quaternion Loss is mainly used for rotation optimization, effectively avoiding the singularity issues associated with Euler angles. How-ever, it is not the optimal choice for pose estimation tasks, particularly in terms of point cloud spatial alignment and deformation.

Although the design of loss functions is not the innovation of this study, we have combined existing geometric loss functions to develop an efficient optimization strategy for addressing geometric errors, deformation smoothness, and global consistency in 6D pose estimation tasks. In our MSPF-LMFF (Multi-Scale Prior Point Cloud Fusion and Lightweight Multi-Feature Fu-sion) framework, the following loss functions are employed: Chamfer Loss: Used to optimize the deformation matrix by calculating the Chamfer distance between the reconstructed 3D point cloud model and the ground truth 3D point cloud model, ensuring geometric alignment between point clouds and further improving the accuracy of the deformation matrix.Laplace Loss: Ensures the smoothness of point cloud deformation. Drawing on the method from [38], we introduce Laplace Loss to constrain the differences between the original prior point cloud and the fused point cloud, preventing excessive deformation and ensuring physical consistency. Smooth Loss (L1 Loss): Used to optimize the correspondence matrix by calculating the error between predicted coordinates and ground truth coordinates, ensuring accurate estimation of pose and spatial transformations while reducing error accumulation. Regularization Terms: Penalize excessive deformation caused by the deformation field D and enforce sparsity in the correspondence matrix. We introduce two regularization terms to ensure the rationality of the deformation process and avoid overfitting. By combining these loss terms, we effectively address geometric errors, deformation smoothness, and global consistency, enhancing the accuracy and stability of the model in 6D pose estimation. Although the design of loss functions is not our innovation, the careful selection and combination of these loss functions enable us to develop an efficient and stable optimization strategy tailored to the task requirements.

3. Method

3.1. Overview

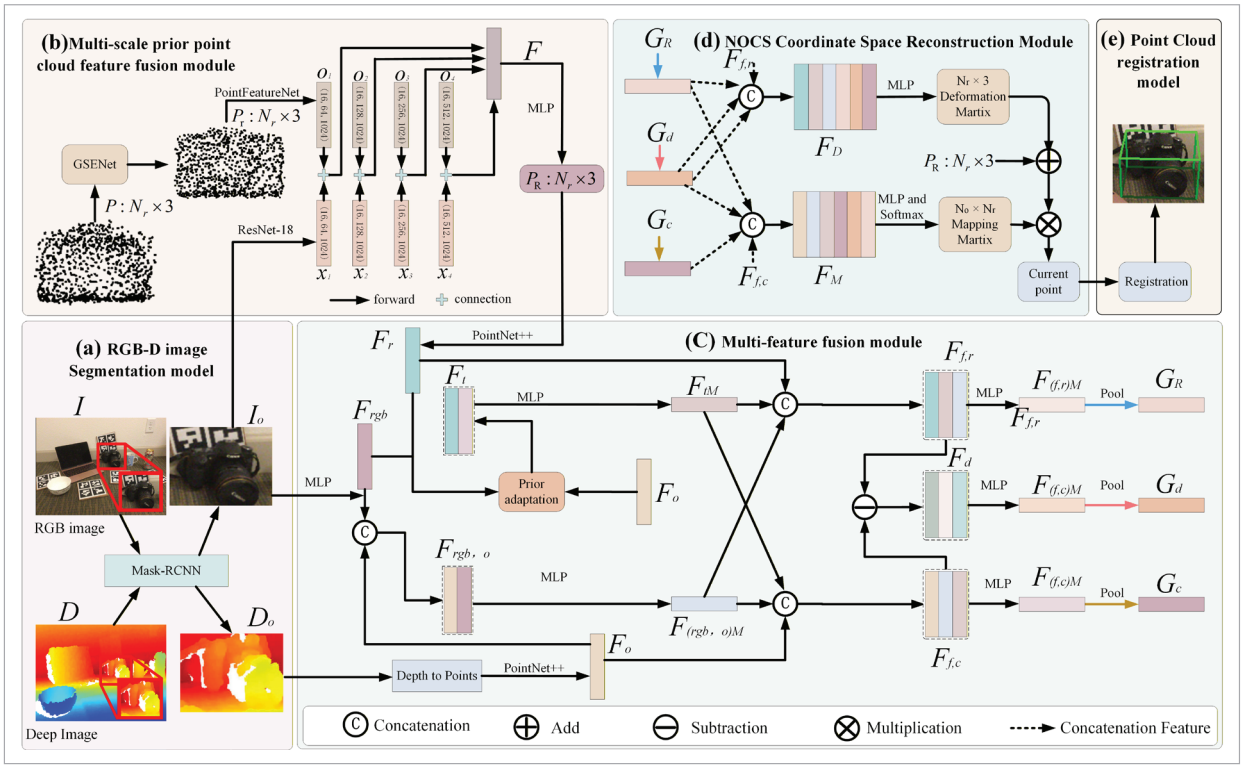
To address the limitations of existing category-level object pose estimation methods in real-world scenarios, this paper proposes the MSPF-LMFF network. The core feature of this network is its ability to significantly enhance the feature representation of point clouds through a multi-scale feature fu-

sion module, enabling more precise alignment with the geometric information of target object point clouds. Additionally, a lightweight multi-feature fusion module is introduced to optimize the interaction process between fused and target point clouds, thereby improving the network's ability to perceive and predict object poses with higher accuracy. The overall framework of MSPF-LMFF is illustrated in Figure 1, comprising five main compo-nents: (a) target segmentation module, (b) multi-scale prior point cloud feature fusion module, (c) light-weight multi-feature fusion module, (d) NOCS coordinate space reconstruction module, and (e) point cloud registration module.

The input to the MSPF-LMFF network includes an RGB image $I \in R^{H\times W\times 3}$,a depth image $D \in R^{H\times W\times 1}$, and a known 3D model (point cloud) $P \in R^{N_r\times 3}$ of the object category. First, the RGB image and its corresponding depth image are fed into the target segmentation module to crop the images to include only the target object $I_o \in R^{H\times W\times 3}$ and $D_o \in R^{H\times W\times 1}$, as shown in Figure 1(a). The cropped RGB image $I_o \in R^{H\times W\times 3}$ and the 3D model $P_r \in R^{N_r\times 3}$ of the known category are then passed into the multi-scale prior point cloud feature fusion module, which generates a fused point cloud $P_R \in R^{N_r\times 3}$ by integrating multi-scale texture features with prior point cloud features, as depicted in Figure 1(b).

Figure 1

The Architecture of MSPF-LMFF. The overall workflow of our framework is as follows. The network consists of five main modules: (a) Object Detection and Segmentation: First, we use Mask R-CNN to detect the object and crop the RGB image, while simultaneously cropping the corresponding depth map to generate the target object's point cloud.(b) Multi-Scale Prior Point Cloud Fusion Module: Next, the point cloud is enriched with features through the multi-scale prior point cloud fusion module to enhance its adaptability to the target object.(c) Feature Extraction and Fusion: Subsequently, fea-tures from the fused point cloud, the target object point cloud, and the cropped RGB image are extracted. These features are input into the lightweight multi-feature fusion module to generate the fused features GR, Gd, and Gc.(d) NOCS Coordi-nate Space Reconstruction: These three features are then fed into the NOCS coordinate space reconstruction module to gen-erate the target object's NOCS representation.(e) Point Cloud Registration: Finally, the Umeyama algorithm [36] is used for point cloud registration, calculating the target object's 6D pose and 3D dimensions.



Next, the fused point cloud $P_R \in R^{N_r \times 3}$, along with the RGB $I_o \in R^{H \times W \times 3}$ and depth images $D_o \in R^{H \times W \times 1}$, is in-put into the lightweight multi-feature fusion module to produce enhanced features $G_R \in R^{B \times C \times 1}$ $G_d \in R^{B \times C \times 1}$ $G_c \in R^{B \times C \times 1}$, as illustrated in Figure 1(c). These features are then provided to the NOCS coordinate space reconstruction module, which reconstructs the NOCS coordinate space of the target object to generate a normalized model of the object, as shown in Figure 1(d). Finally, the reconstructed NOCS coordinate space is input into the point cloud registration module, where the Umeyama algorithm is used to compute the 6D pose of the target object, as detailed in Figure 1(e).

3.2. RGB-D image Segmentation Module

The objective of this section is to segment regions con-taining only the target object from the RGB image and its corresponding depth image, as illustrated in Figure 1(a). Specifically, the RGB image $I \in R^{H \times W \times 3}$ and depth image $D \in R^{H \times W \times 1}$ are input into a Mask R-CNN [9] network, which crops the image and depth map to re-tain only the regions containing the target object $I_0 \in R^{H_0 \times W_0 \times 3}$ and $D_0 \in R^{H_0 \times W_0 \times 1}$. Let H and W de-note the width and height of the input image, and H_0, W_0 denote the width and height of the cropped image I_0. The process can be represented by the following equations:

$$I_o = M\text{ask} - RCNN\left(I\right) \tag{1}$$

$$D_o = Mask - RCNN\left(D\right) \tag{2}$$

3.3. Multi-scale Point Cloud Fusion Module

The objective of this section is to enhance the feature representation of point clouds by integrating multi-scale prior point cloud features with image texture features, thereby improving the completeness of geometric information. Through multi-scale feature fusion, prior point clouds are able to comprehensively capture both local details and global structural characteristics of the target object, avoiding potential information loss caused by single-scale fusion. Specifically, we construct a global shape representation from a point cloud consisting of 1024 points, aiming for the global shape to capture shared geometric features of objects within the same category. This global shape serves as prior information for category-level 6D object pose es-timation. Previous studies utilized PointNet-based en-coders, which effectively extract prior features for ob-jects with simple geometric structures (e.g., bottes). However, these encoders demonstrate insufficient pre-cision in extracting prior features for objects with complex geometric structures (e.g., cameras). To ad-dress this limitation, we employ the GSENet [28] net-work, which enhances the model's feature extraction capability. GSENet effectively captures 3D points along the edges of cameras while preserving key point information, thereby improving the extraction of prior features for objects with complex geometries.

The architecture of the proposed multi-scale prior point cloud module is shown in Figure 1(b). The input to this module consists of the cropped RGB image $I_0 \in R^{H_0 \times W_0 \times 3}$ and the known 3D model (point cloud) $P \in R^{N_r \times 3}$ of the object category. First, the known 3D model $P \in R^{N_r \times 3}$ is fed into GSENet to extract the prior point cloud $P_r \in R^{N_r \times 3}$. The loss is then calculated using the Chamfer Distance, as described by the following equation:

$$\begin{aligned} D_{cd}\left(M_c^i M_c^g\right) = & \sum_{x \in M_c^i} \min_{y \in M_i^g} \left\|x - y\right\|_2^2 \\ & + \sum_{y \in M_i^g} \min_{x \in M_c^i} \left\|x - y\right\|_2^2 \end{aligned} \tag{3}$$

Here, M_c^i and M_c^g represent the 3D model (point cloud) of instance i in category c and the corresponding 3D global shape (point cloud) extracted by GSENet, respectively. The first term of the Chamfer Distance represents the sum of the minimum distances from each point in M_c^i to M_c^g, while the second term represents the sum of the minimum distances from each point in M_c^g to M_c^i. A smaller Chamfer Distance indicates better alignment between the prior shape and the instance point cloud.

However, existing methods, after extracting prior shapes, only consider single-scale and single-feature extraction for prior point clouds, failing to fully utilize the fusion of multi-scale image texture features and prior point cloud features. This limitation leads to significant discrepancies between the prior point cloud and the surface texture of the object, as well as between the prior point cloud and the target

object point cloud. To address this issue, this paper employs ResNet-18 [10] and PointFeatureNet to extract multi-scale image features and prior point cloud features. Specifically, for the target object image, ResNet-18 is used to extract four different scales of image features $X_i(i=1,2,3,4)$, as expressed by the following equation:

$$X_i = \mathrm{Re}\,sNet-18\left(I_o\right) \quad i=1,2,3,4. \tag{4}$$

Here, I_o represents the cropped RGB image of the target object, and X_i denotes the feature map extracted from the i-th layer of ResNet-18. Each feature map X_i corresponds to a different scale of information, capturing more detailed texture features. This provides rich feature support for the subsequent multi-scale feature fusion operations.

The architecture of the proposed multi-scale prior point cloud module is shown in Figure 2. Specifically, for the prior point cloud $P_r \in R^{B\times N_r\times 3}$, the coordinate distance between each central point and the remaining points is first calculated. This operation generates a distance matrix $D \in R^{N_r\times N_r}$, where $D_{i,j} = d(p_i,p_j)$ represents the distance between the i-th central point and the j-th point. Based on the distance matrix D the K-nearest points are selected for each central point P_i using the K-nearest neighbors (KNN) algorithm, as de-scribed by the following equation:

$$\begin{aligned} &d\left(p_i^{(B)},p_j^{(B)}\right)=\left\|p_i^{(B)}-p_j^{(B)}\right\|_2 \; p_i^{(B)},p_j^{(B)}\in P_r \\ &N_k^{(B)}\left(x_i^{(B)}\right)=\arg top-kd\left(x_i^{(B)},x_j^{(B)}\right) \\ &N_k\left(x_i^{(B)}\right)=\left\{x_{j1}^{(B)},x_{j2}^{(B)},\ldots,x_{jk}^{(B)}\right\} \\ &d\left(x_i^{(B)},x_{j1}^{(B)}\right)\le d\left(x_i^{(B)},x_{j2}^{(B)}\right)\le\cdots\le d\left(x_i^{(B)},x_j^{(B)}\right) \end{aligned} . \tag{5}$$

Here, $d\left(p_i^{(B)},p_j^{(B)}\right)$ represents the distance between the point p_i and the point p_j, $\|\ \|$ denotes the Euclidean norm, $p_i^{(B)}$ is the central point of the B-th point cloud, and $p_j^{(B)}$ refers to the coordinates of the remaining points in the B-th point cloud. $N_k\left(x_i^{(B)}\right)$ represents the set of K-nearest neighbors of the i-th point in the point cloud.

At the same time, a directed graph $G=(V,E)$ is constructed, where $V=\{1,\ldots N_r\}$ represents the set of vertices and $E\subseteq V\times V$ represents the set of edges. The graph G is constructed using the K-Nearest Neighbor (KNN) algorithm. This graph includes self-loops, meaning that each node also points to itself. By defining the local neighborhood for each point through the KNN algorithm, the spatial relationships within the point cloud data are explicitly established. This enables the network to capture the geometric structure in-formation between points and their neighboring points.

Next, for each point, its 3D coordinates are concatenated with the relative coordinates of its k-nearest neigh-bors, ultimately forming point cloud features $F_{K_1}\in{}^{\sim B\times C\times N_r\times k}$ enriched with local geometric information. In this way, the extracted features not only capture the geometric relationships between each point and its neighborhood but also provide abundant local structural information. This process enables the network to effectively capture the local geometric features and global shape information of the prior point cloud, providing a stronger foundation for feature input into subsequent modules. The process can be described by the following equation:

$$F_{K_1}=KG\left(P_R\right). \tag{6}$$

Here, $d(p_i,p_j)$ represents the distance between point p_i and point $p_jB = 16$ denotes the batch size of the point clouds during training, $C = 6$ refers to the number of feature channels, $N_r = 1024$ 4 represents the number of points in each point cloud, and $k = 20$ in-dicates the number of neighboring points for each point.

After obtaining the features $F_{K_1}\in{}^{\sim B\times C\times N_r\times k}$, they are fed into a convolutional module. First, $F_{K_1}\in{}^{\sim B\times C\times N_r\times k}$ is passed through a 1×1 convolution kernel to generate the convolved features $e_{i,jm}\in{}^{\sim B\times C_1\times N_r\times k}$. The convolved features are then de-fined through the following nonlinear activation function:

$$\begin{aligned} e_{i,jm}&=h_0\left(x_i,x_j\right) \\ &=\mathrm{Re}\,LU\left(\theta_m\cdot\left(x_i-x_j\right)+\phi_m\cdot x_i\right). \end{aligned} \tag{7}$$

Here, $h_\theta:{}^{\sim F}\times{}^{\sim F}\rightarrow{}^{\sim F'}$ is a nonlinear function with learnable parameters θ, where θ_m and ϕ_m are learnable weight parameters.

$x_i - x_j$ captures local spatial information (the relative positions of neighboring points with respect to the central point), while x_i captures global shape

Figure 2
The Architecture of PointFeatureNet.

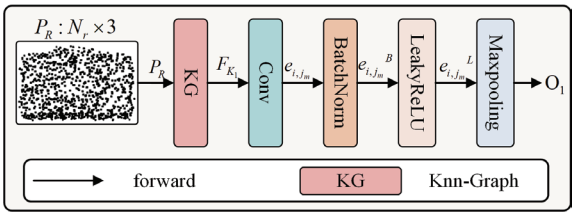


Figure 3
The relationship between the center point and its neighboring points before the convolution operation.

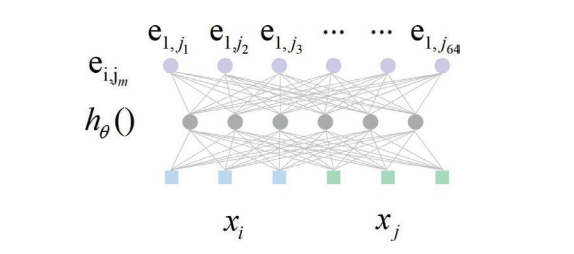
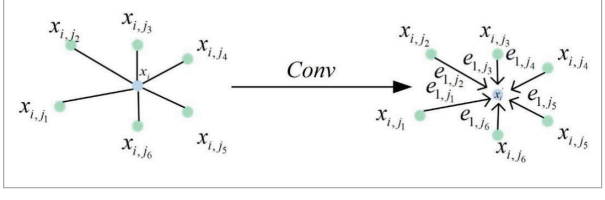


Figure 4
Relationship between the center point and its neighboring points aher the convolution operation.



information (the coordinates of the central point). $Leaky\mathrm{Re}LU$ is the activation function, introduced to incorporate nonlinearity. This nonlinear function effectively combines local features and global features by extracting the relative geometric information $x_i - x_j$ of neighboring points with respect to the central point and the global shape information x_i.

The convolution process is illustrated in Figure 3, which shows the changes in the relationship between the central point and its neighboring points before and after the convolution operation. Before the convolution, as shown in Figure 4(a), the relative coordinate iformation of the neighborhood points is modeled through the adjacency relations constructed by the KNN graph. After the convolution, as shown in Figure 4(b), the central point's features are made more expressive through nonlinear transformations and aggregation operations. Finally, the convolved features can be expressed as:

$$e_{i,jm} = Conv\left(F_{K_1}\right). \tag{8}$$

The features $e_{i,jm} \in {}^{\sim B\times C_1\times N_r\times k}$ conv are then passed through a normalization layer for normalization. This layer normalizes the features within each batch, thereby reducing numerical fluctuations during gradient updates. The resulting normalized features $e_{i,jm}{}^B \in {}^{\sim B\times C_1\times N_r\times k}$ are obtained as follows:

$$e_{i,jm}{}^B = Batch\ Norm\left(e_{i,jm}\right). \tag{9}$$

Subsequently, the normalized features are passed through an activation function (LeakyReLU) to enhance the feature representation capability of the net-work, resulting in the activated features $e_{i,jm}{}^L \in {}^{\sim B\times C_1\times N_r\times k}$, which are expressed as follows:

$$e_{i,jm}{}^L = Leaky\,\mathrm{Re}\,LU\left(e_{i,jm}{}^B\right). \tag{10}$$

Finally, max pooling is applied to aggregate the neighborhood features $x_{im} = \max_{j:(i,j)\in E} e_{ijm}$. This operation extracts the most salient features within the neighborhood, enhancing the network's ability to perceive geometric information while reducing redundant information. This process generates point cloud features with enhanced local geometric and global shape characteristics, providing richer feature representations for subsequent modules. The process is expressed as follows:

$$O_1 = \max e_{i,jm}^L = \begin{bmatrix} \max\left(e_{1,1}, e_{2,1}, \ldots, e_{k,1}\right) \\ \max\left(e_{1,2}, e_{2,2}, \ldots, e_{k,2}\right) \\ \vdots \\ \max\left(e_{1,C_1}, e_{2,C_1}, \ldots, e_{k,C_1}\right) \end{bmatrix}. \tag{11}$$

Here, the max pooling operation $\max(\ )$ is used to select the maximum value for each feature dimension within the neighborhood. Specifically, the max pooling operation selects the maximum value from the features of 20 neighbors for each feature dimension, effectively filtering out the strongest responses

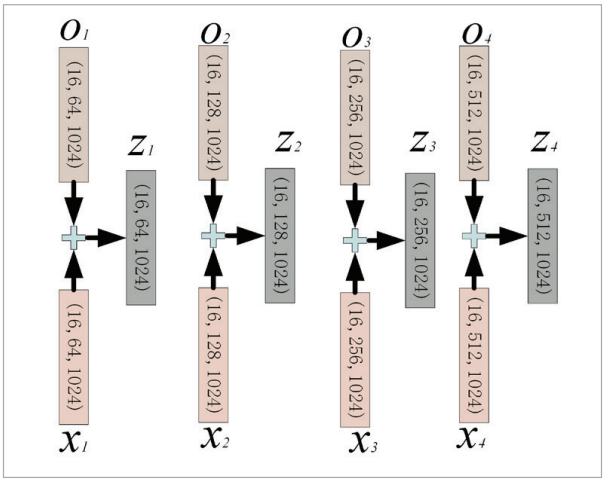
in the local region. For each central point, the result of the max pooling operation can be regarded as extracting the most geometrically significant features from its neighborhood. After max pooling, the module does not retain all the features of the neighbors but instead selects the most representative features within the neighborhood. These retained features capture the most critical geometric information of the point in the local space, effectively representing the structural relationship between the central point and its neighborhood. Ultimately, the combination of the central point's features with the most significant features from its neighborhood constitutes a compact and representative feature representation. This approach avoids the interference of redundant information during subsequent processing, enabling the model to focus on the key geometric features of the point cloud. Meanwhile, max pooling preserves the "most significant" feature information from the central point and its neighborhood, optimizing the data structure and improving the model's computational efficiency and robustness in feature representation. The resulting feature $O_1 \in {}^{\sim B\times C_1\times N_r}$ is obtained, and similar operations are applied to extract corresponding features O_2, O_3, O_4.

Next, the image features and the prior point cloud features are fused at the corresponding scales by performing elementwise addition. This integration combines the coordinate information of the point cloud with the texture information of the image, resulting in a new feature representation. This fusion method effectively combines the data from both modalities, enabling the model to simultaneously leverage the texture information from the image and the geometric information from the point cloud. This enhances the model's ability to comprehensively understand the scene and objects. The element-wise addition is illustrated in Figure 5. The fusion process is expressed as follows:

$$Z_i = X_i + O_i \quad i = 1,2,3,4\,. \tag{12}$$

Here, Z_i represents the fused features at the i–th layer, while X_i and O_i denote the image features and point cloud features at the corresponding scale, respec-tively. Through this approach, the network is able to simultaneously capture multiscale texture information and geometric information of the target object.

Figure 5
The process of element-wise addition.



Next, the fused features are concatenated, as expressed in the following equation:

$$F = concat\left(Z_1, Z_2, Z_3, Z_4\right). \tag{13}$$

Here, Z_1, Z_2, Z_3 and Z_4 represent the fused features at different scales, and concat denotes the concatenation operation.Subsequently, a multi-layer perceptron (MLP) is applied to process the concatenated features F, further enriching the point cloud information. The process is expressed as follows:

$$P_R = MLP\left(F\right). \tag{14}$$

Here, F represents the concatenated features, MLP denotes the multi-layer perceptron, and $P_R \in {}^{\sim N_r\times 3}$ is the fused point cloud. This approach enables the fused point cloud to better adapt to the geometric shape of the target object, thereby improving the accuracy and generalization capability of pose estimation.

2.2. Lightweight Multi-feature Fusion Module

This process is illustrated in Figure 1(c). First, we use a multi-layer perceptron (MLP) and PointNet++ [31] to extract features from the image and the target object point cloud features. After this stage, the features of the image $F_{rgb} \in {}^{\sim C_{rgb}\times N_o}$, the target object point cloud $F_o \in {}^{\sim C_o\times N_o}$, and the fused point cloud $F_o \in {}^{\sim C_o\times N_o}$ are obtained, where C_{rgb}, C_o, C_R denote the respective channel dimensions of these features.

Next, the features F_{rgb}, F_o, F_R are input into the prior adaptation module, which is inspired by the method proposed in [44]. This module fuses these features to obtain the geometric features of the fused point cloud $F_t \in {}^{\sim C_t \times N_r}$. Here, C_t represents the channel dimension of the fused point cloud after passing through the prior adaptation module, and C_{tM} denotes the channel dimension after being processed by the multi-layer perceptron (MLP).

Subsequently, F_t is fed into a multi-layer perceptron (MLP) to obtain the features F_{tM}.

$$F_{tM} = MLP\left(F_t\right). \tag{15}$$

Next, F_{rgb} and F_o are concatenated to obtain the geometric features of the target object point cloud $F_{rgb,o} \in {}^{\sim C_{rgb} \times N_r}$, Subsequently, $F_{rgb,o} \in {}^{\sim C_{rgb,o} \times N_r}$ is fed into a multi-layer perceptron (MLP) to produce the features $F_{(rgb,o)M} \in {}^{\sim C_{(rgb,o)M} \times N_r}$, where $C_{(rgb,o)}$ represents the channel dimension of the concatenated features F_{rgb} and F_o, and $C_{(rgb,o)M}$ represents the channel dimension of the target object point cloud features after being processed by the MLP.

$$F_{rgb,o} = Concat\left(F_{rgb}, F_o\right) \tag{16}$$

$$F_{(rgb,o)M} = MLP\left(F_{rgb,o}\right). \tag{17}$$

Next, F_t, F_{tM}, $F_{(rgb,o)}$ are concatenated to obtain the fused point cloud feature $F_{f,r} \in {}^{\sim C_{f,r} \times N_r}$, integrating the geometric information of the target object into the shape-fused point cloud. Simultaneously F_o, $F_{(rgb,o)}$, F_{tM} are concatenated to produce the current point cloud feature $F_{f,c} \in {}^{\sim C_{f,c} \times N_T}$, thereby incorporating the geometric features of the fused point cloud into the target object point cloud. This feature fusion not only enhances the robustness of the fused point cloud but also improves the model's generalization capability when handling different instances within the same category. Here, $C_{f,r}$ represents the channel dimension of the fused point cloud feature after concatenation, and $C_{f,c}$ represents the channel dimension of the target object point cloud feature after concatenation.

$$F_{f,r} = Concat\left(F_r, F_{tM}, F_{(rgb,o)M}\right) \tag{18}$$

$$F_{f,c} = Concat\left(F_o, F_{(rgb,o)M}, F_{tM}\right). \tag{19}$$

Subsequently, the difference feature $F_{f,r}$ is obtained by subtracting $F_{f,r}$ from $F_{f,c}$ where C_d represents the channel dimension of the difference feature. The subtraction operation captures the differences between the fused point cloud and the target object point cloud, enabling the network to better understand their distinct features, thereby optimizing subsequent point cloud reconstruction and pose estimation.

$$F_d = F_{f,r} - F_{f.c} \tag{20}$$

Finally, $F_{f,r}, F_{f,c}, F_{f,d}$ are individually fed into a multi-layer perceptron (MLP), an average pooling layer (Pool), and another MLP, respectively, to obtain the features $G_R \in {}^{\sim C_R \times N_r}$, $G_d \in {}^{\sim C_d \times N_r}$, $G_c \in {}^{\sim C_c \times N_r}$. Here, C_R, C_d, C_c represent the channel dimensions of the fused point cloud feature, the difference feature, and the target object point cloud feature, respectively. These three features serve as inputs to the subsequent network for reconstructing the NOCS shape of unknown objects within the same category.

$$G_R = MLP\left(Pool\left(F_{f,r}\right)\right) \tag{21}$$

$$G_d = MLP\left(Pool\left(F_{f,c}\right)\right) \tag{22}$$

$$G_c = MLP\left(Pool\left(F_d\right)\right). \tag{23}$$

3.4. NOCS Coordinate Space Reconstruction Module

This process is illustrated in Figure 1(d). We use a multi-layer perceptron (MLP) to transform the features $F_{f,r}$ and $F_{f,c}$ into the corresponding features G_R and G_c, respectively. Additionally, the feature F_d is converted into the corresponding feature G_d through average pooling. Subsequently, the three features G_R, G_c and G_d are concatenated with the fused point cloud feature $F_{f,r}$ and the deformation matrix $D_m \in {}^{\sim N_r \times 3}$ is computed using a multi-layer perceptron.

$$D_m = MLP\left(Concat\left(F_{f,r}, G_c, G_d, G_R\right)\right). \tag{24}$$

Subsequently, the features G_R, G_c and G_d are concatenated with the target object point cloud feature $F_{f,r}$, to regress the mapping matrix $D_m \in {}^{N_o \times N_r}$. This process can be expressed as follows:

$$M_m = Soft\max\left(MLP\left(\left(Concat\left(F_{f,c}, G_c, G_d, G_R\right)\right)\right)\right). \quad (25)$$

Finally, the NOCS shape of the unknown object within the category is reconstructed as follows:

$$T = P_R + D_m \quad (26)$$

$$C_p = M_m \times T\ . \quad (27)$$

Here, "+" and "×" denote matrix addition and matrix multiplication, respectively P_R represents the points in the fused point cloud, with a dimension of $N_r \times 3 \times 3$, where N_r is the number of points in the point cloud. The deformation matrix D_m is used to deform the fused point cloud. The transformation relationship between the deformed shape and the target object's NOCS shape $C_p \in {}^{N_o \times 3}$ is described by the mapping matrix $M_m \in {}^{N_o \times N_r}$.

3.5. Point Cloud Registration Module

This process is illustrated in Figure 1(e). After obtaining the current NOCS shape C_p, the 6D pose of an unknown object within the category can be estimated through point cloud registration. The core objective of point cloud registration is to model the geometric relationship between the target object point cloud and C_p, thereby deriving the rotation matrix $R \in SO(3)$, translation vector $t \in {}^3$, and scaling factor $s \in {}$. We adopt the Umeyama algorithm, which aligns point clouds by minimizing the mean squared error between the target object point cloud P_o and C_p. The process is formulated as follows:

$$\mathcal{L}_{MSE} = \frac{1}{N_r}\sum_{i=1}^{N_r}\left\| sRP_{o,i} + t - C_{p,i}\right\|^2 . \quad (28)$$

Here, Nr represents the number of points in the point cloud, $P_{o,i}$ is the coordinate of the i–th point in the target object point cloud, and $C_{p,i}$ is the coordinate of the i–th point in the reconstructed point cloud $R \in SO(3)$ denotes the rotational relationship between the reconstructed point cloud and the target object point cloud, $t \in {}^3$ represents the translation vector from the reconstructed point cloud to the target object point cloud, and $s \in {}$ is the scaling factor used to adjust the scale of the point cloud. $\| \ \|$ indicates the squared Euclidean distance between two points.

First, the centroids of the target object point cloud and the reconstructed point cloud are calculated. The calculation is expressed as follows:

$$\mu_{P_o} = \frac{1}{N_r}\sum_{i=1}^{N_r} P_{o,i} \quad (29)$$

$$\mu_{C_p} = \frac{1}{N_r}\sum_{i=1}^{N_r} P_{p,i}\ . \quad (30)$$

Here, μ_{P_o} represents the centroid coordinates of the target object point cloud, μ_{C_p} represents the centroid coordinates of the reconstructed point cloud, $P_{o,i}$ denotes the coordinates of the i–th point in the target object point cloud, and $C_{p,i}$ denotes the coordinates of the i–th point in the reconstructed point cloud.

Next, the point clouds are centralized by subtracting their respective centroids. The process is expressed as follows:

$$\tilde{P}_{o,i} = P_{o,i} - \mu_{P_o} \quad (31)$$

$$\tilde{C}_{p,i} = C_{p,i} - \mu_{C_p}\ . \quad (32)$$

Here, $\tilde{P}_{o,i}$ represents the i–th point of the decentralized target object point cloud, and $\tilde{C}_{p,i}$ represents the i–th point of the decentralized reconstructed point cloud.

After decentralizing the point clouds, the covariance matrix is computed. The calculation is expressed as follows:

$$\Sigma = \frac{1}{N_r}\sum_{i=1}^{N_r}\tilde{P}_{o,i}\tilde{C}_{p,i}^{\mathrm{T}}\ . \quad (33)$$

Here, $\sum$ represents the covariance matrix, which is used to express the correlation between the target object point cloud and the reconstructed point cloud. $()^{\mathrm{T}}$ denotes the matrix transpose operation.

Next, a Singular Value Decomposition (SVD) is performed on the covariance matrix. The formula is expressed as follows:

$$\sum = USV^{\mathrm{T}}. \tag{34}$$

Here, U and V represent orthogonal matrices, and S represents a diagonal matrix, with the values on its diagonal being the singular values.

Finally, the rotation matrix R, scaling factor s, and translation vector t are computed. The formula is expressed as follows:

$$R = VU^{\mathrm{T}} \tag{35}$$

$$s = \frac{Tr(S)}{\sum_{i=1}^{N_r} \left\| \tilde{P}_{o,i} \right\|^2} \tag{36}$$

$$t = \mu_{C_p} - sR\mu_{P_o}. \tag{37}$$

Using the above formulas, the optimal alignment parameters (R,t,s) between the observed point cloud and the target point cloud are obtained, thereby achieving the estimation of the 6D pose of the unknown object.

3.6. Loss Function

To ensure the smoothness of point cloud deformation, we draw upon the method proposed in [7] and apply the Laplacian loss to constrain excessive deformations. Specifically, the Laplacian loss is used to regulate the differences between the original prior point cloud and the fused point cloud.

Our MSPF-LMFF produces two intermediate outputs: the deformation field D_m and the corresponding matrix A which are used to calculate the final 6D pose and 3D dimensions. To optimize MSPF-LMFF, we adopt the same training strategy as described in [28]. First, we optimize the deformation matrix D_m by calculating the Chamfer Distance between the reconstructed 3D point cloud model PR and the ground truth 3D point cloud model P_o. The formula is expressed as follows:

$$L_{cd} = \sum_{p_i \in \mathrm{P}_r} \min_{p_j \in P_o} \left\| p_i - p_j \right\|_2 + \sum_{p_i \in P_o} \min_{p_j \in P_r} \left\| p_i - p_j \right\|_2. \tag{38}$$

Next, to ensure the smoothness of point cloud deformation, we refer to the method proposed in [26] and apply the Laplacian loss to constrain excessive deformation. Specifically, the Laplacian loss is used to regulate the differences between the original prior point cloud P_r and the fused point cloud P_R.

$$L_{laplacian} = \sum_{N_o} \left\| L_{pc}(P_r) - L_{pc}(P_R) \right\|. \tag{39}$$

Here, L_{pc} represents the Laplacian operator, while P_r and P_R denote the original prior point cloud and the fused point cloud, respectively

Next, to optimize the correspondence matrix A, we introduce a smooth loss L_1 to constrain the error between the predicted coordinates NOCS $\hat{M}$ and the ground truth NOCS coordinates M. This loss function calculates the average loss over all coordinates, expressed as follows:

$$\hat{L}_{corr}(m, m_{gt}) = \begin{cases} 5(m - m_{gt})^2, & \text{if } |m - m_{gt}| \le 0.1 \\ |m - m_{gt}| - 0.05, & \text{otherwise} \end{cases} \tag{40}$$

$$L_{corr}(\hat{M}, M) = \frac{1}{N_v} \sum_{m \in \hat{M}} \hat{L}_{\mathrm{corr}}(m, m_{gt}). \tag{41}$$

Here, m_{gt} represents the ground truth NOCS values from M, and m denotes the predicted NOCS values. Therefore, the correspondence loss is calculated as the average matching loss over all NOCS values.

To penalize large deformations caused by the deformation field D_m and to constrain the sparsity of the correspondence matrix A, we impose two regularization terms on each row d_i of the deformation field D_m and each row A_i of the correspondence matrix A, respectively, as follows:

$$L_{entropy} = \frac{1}{N_o} \sum_i -A_i \log A_i \tag{42}$$

$$L_{def} = \frac{1}{N_r} \sum_{d_i \in D_m} \left\| d_i \right\|_2. \tag{43}$$

Finally, the overall loss function is the weighted sum of the five individual loss functions, where λ_1, λ_2, λ_3, λ_4, λ_5 are the regularization coefficients.

$$L_{laplacian} = \sum_{N_o} \left\| L_{pc}\left(P_r\right) - L_{pc}\left(P_R\right) \right\| . \quad (39)$$

4. Experiment

In this section, we conducted extensive experiments on the NOCS-REAL [39] and NOCS-CAMERA [39] da-tasets to evaluate the performance of MSPF-LMFF and compare it with state-of-the-art methods. Additionally, we assessed the generalization capability of our MSPF-LMFF on the Wild6D [8] dataset, which provides more challenging real-world scenarios compared to the NOCS datasets. To further validate the advantages of MSPF-LMFF, we performed comprehensive ablation studies and provided visualizations of the pose estimation results, qualitatively demonstrating the effectiveness of our approach.

4.1. Dataset

Our experiments are rigorously conducted based on the following three benchmark datasets: NOCS-CAMERA, NOCS-REAL, and WILD6D. The NOCS-CAMERA dataset consists of 300,000 carefully designed images, generated using advanced rendering techniques to simulate objects in real-world scenes, ensuring data richness and diversity. By leveraging virtual rendering, NOCS-CAMERA provides a highly controllable image environment, which facilitates the training and evaluation of object pose estimation algorithms.

In contrast, the NOCS-REAL dataset contains 4,300 images captured by real cameras, with 2,750 images collected from six different real-world scenes. These images reflect the authentic appearance of objects and include complex lighting conditions and background information, making this dataset an ideal choice for evaluating algorithm performance in real-world environments. The complexity of the NOCS-REAL dataset poses greater challenges for pose estimation and provides a more rigorous testing environment for practical application scenarios.

Both datasets cover six common categories of everyday objects: bottles, bowls, cameras, cans, laptops, and mugs. These object categories provide a comprehensive testing benchmark for object pose estimation algorithms.

The WILD6D dataset consists of images captured from real everyday scenes rather than being generated in controlled laboratory environments. This results in a greater diversity of objects and higher scene complexity, significantly increasing the difficulty of object pose estimation. Objects in the dataset are captured from multiple view-points, covering a wide range of angles and lighting conditions, further enhancing the requirements for testing the generalization capability of the algorithms.

4.2. Preprocessing

We load images, depth maps, masks, coordinate maps, and annotation information from the specified data directories. The data sources include a synthetic dataset (NOCS-CAMERA) and a real-world dataset (NOCS-REAL). After loading, the image and depth map data undergo preprocessing to ensure data quality and consistency. Simultaneously, point cloud data is sampled to ensure that each sample contains a fixed number of points.

In training mode, data augmentation is applied by introducing color jittering to the images and random translation and jittering to the point cloud data. These augmentation operations help increase data diversity, thereby enhancing the model's robustness, particularly when dealing with varying viewing angles, lighting conditions, and partial occlusions. Subsequently, the system loads predefined 3D models and category-averaged shapes, which will be used for subsequent pose estimation tasks. For symmetric objects (e.g., cups, bowls, etc.), we apply pose normalization by rotating these objects to a canonical pose, effectively reducing ambiguity in pose estimation. This approach enables the model to handle pose estimation for symmetric objects more accurately.

Finally, the preprocessing pipeline returns processed point cloud data, images, selected indices, category labels, 3D models, shape priors, transformation matrices, and normalized object coordinates (NOCS). These data provide a unified input format for the 3D object pose estimation task, facilitating subsequent model training and inference.

For point cloud sampling from .obj model files, all sampled model points undergo standardization to ensure they reside within a unified coordinate range and are centered at the origin. Additionally, models of different categories (e.g., cups, bottles, etc.) are realigned

and normalized to ensure consistency and comparability across categories. The entire dataset is divided into training (train) and validation (val) subsets, with point cloud data in each subset organized and stored according to category labels. This data processing pipeline supports the use of the Farthest Point Sampling (FPS) method to control the number of sampled points, ensuring sample representativeness and uniform distribution. All processed point cloud data and their corresponding labels are saved in HDF5 format, which not only facilitates large-scale data storage but also improves data reading efficiency.

4.3. Evaluation Metrics

We use two criteria to evaluate the performance of MSPF-LMFF and compare it with state-of-the-art methods. First, we report the mean Average Precision (mAP) of the 3D Intersection over Union (IoU) at different thresholds to jointly assess the accuracy of rotation, translation, and size estimation. Second, we use the n°m cm metric to directly compare the errors in rotation and translation. These two metrics are applied to the NOCS-REAL, NOCS-CAMERA, and Wild6D datasets.

IoU_x: We use the 3D Intersection over Union (IoU) and centimeter-level accuracy (n°m cm) to quantitatively evaluate the performance of MSPF-LMFF. Specifically, a predicted pose is considered accurate only when the IoU value between the predicted 3D bounding box and the ground truth bounding box exceeds a predefined threshold. We adopt IoU_{50} and IoU_{75} as evaluation standards, where the predicted pose is deemed accurate if the IoU value reaches or exceeds 50% and 75%, respectively.

n°m cm: This metric is used to evaluate the model's performance based on rotation and translation errors of the pose. A prediction is considered successful if the difference between the predicted and ground-truth rotations is less than n°, and the difference between the predicted and ground-truth translations is less than m centimeters. In this paper, we use n°m cm, 5°5cm, 10°2cm, 10°5cm, and 10°10cm as evaluation metrics.

4.4. Implementation Details

In our experiments, we used instance segmentation masks generated by Mask R-CNN [51] to extract N_o = 1024 points from the depth image, forming the current point cloud. The unfused prior point cloud was obtained by extracting N_r = 1024 points from the GSENet network. In Section 3.2, the feature dimensions of the RGB image were X_1[16,64,1024], X_2[16,128,1024], X_3[16,256,1024], X_4[16,512,1024]; while the dimensions of the prior point cloud were O_1[16,64,1024], O_2[16,128,1024], O_3[16,256,1024], O_4[16,512,1024]. In Section 3.3, the dimensions of F_r, F_{rgb}, and F_o were all [16,64,1024]; the dimensions of F_t and F_{rgb}, were [16128,1024]; the dimensions of F_{tM} and $F_{(rgb,o)M}$ were [16,128,1024]; and the dimensions of $F_{f,r}$, $F_{f,c}$, and F_d were [16,320,1024]; The dimensions of G_R, G_d, and G_c were [16,128,1], [16,320,1], and [16,128,1], respectively. In Section 3.4, the dimensions of F_D and F_M were both [16,896,1024]. The regularization coefficients $\{\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5\}$ were set to {5.0,1.0,1.0,0.01,0.0001}.

4.5. Performance on NOCS-REAL Dataset

We used the Adam optimizer [16] to train the network, with the learning rate set to 1e-3, and trained for up to 100 epochs. The learning rate was decayed by factors of 0.6, 0.3, 0.1, and 0.01 at every 20th epoch. All experiments were conducted on a computer equipped with an NVIDIA GeForce RTX 4090 GPU, with a batch size of 16. The NOCS-REAL dataset presents greater challenges compared to the NOCS-CAMERA dataset due to its real-world complexity and limited training data. This dataset contains only 18 object instances (3 per category) and uses 4,300 images for training. Since the limited data in NOCS-REAL is insufficient to fully support network training, we utilized the synthetic dataset NOCS-CAMERA (which includes 275K training images) to assist training. During training, we randomly selected images from NOCS-CAMERA and NOCS-REAL with a 1:3 ratio for joint training. Table 1 presents the performance of our MSPF-LMFF method compared to 13 state-of-the-art methods on the NOCS-REAL dataset [5]. The experimental results show that our method surpasses existing methods on multiple metrics. For instance, on the IoU75, 5°2cm, 5°5cm, 10°2cm, and 10°5cm evaluation metrics, our method achieved average accuracies of 69.1%, 52%, 60.5%, 70.2%, and 81.5%, respectively. These results exceed those of CD-POSE [37] by 0.5%, 12.2%, 15.6%, 8.4%, and 9.9%, respectively. On the IoU50 metric, our MSPF-LMFF achieved an average accuracy of 88.9%, outperforming CD-POSE by 7.9%. Figure 6 shows some qualitative results on

the NOCS-REAL dataset. Both the quantitative and qualitative analyses demonstrate that our method exhibits high accuracy and robustness in real world scenarios. Figure 7 illustrates the 6D object poses estimated by MSPF-LMFF under complex occlusion conditions.

Table 1

The quantitative results of our method and the state-of-the-art (SOTA) methods on the NOCS-REAL dataset are presented, evaluated using IoU and n°m cm metrics.

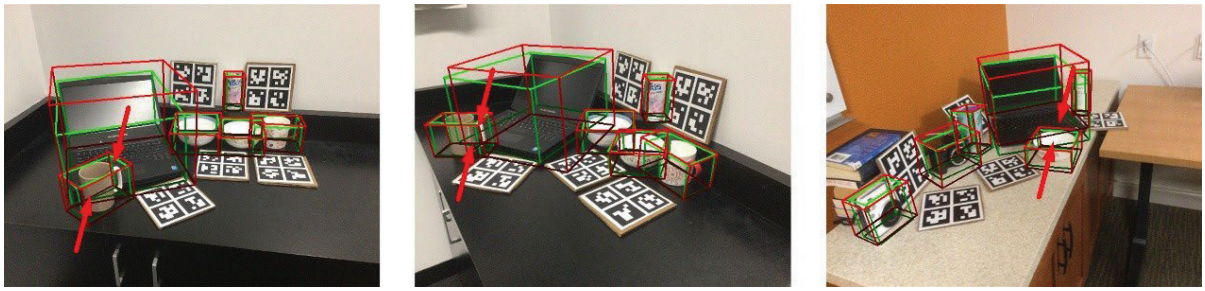
Method	Data	Prior	NOCS-REAL						
			$3D_{50}$	$3D_{75}$	5°2*cm*	5°5*cm*	10°2*cm*	10°5*cm*	10°10*cm*
NOCS [*CVPR*'19]	*RGB-D*	×	78.0	30.1	7.2	10.0	13.8	25.2	-
SPD [*ECCV*'20]	*RGB-D*	√	77.3	53.2	19.3	21.4	43.2	54.1	-
DPN [*ICCV*'21]	*RGB-D*	×	79.8	62.2	29.3	35.9	50.0	66.8	-
SS–Conv [*NeurIPS*'21]	*RGB-D*	×	79.8	65.6	36.6	43.4	52.6	63.5	-
CR–Net [*IROS*'21]	*RGB-D*	√	79.3	55.9	27.8	34.3	47.2	60.8	-
SGPA[*ICCV*'21]	*RGB-D*	√	80.1	61.9	35.9	39.6	61.3	70.7	-
CenterSnap [*ICRA*'22]	*RGB-D*	×	80.2	-	-	29.1	-	64.3	-
SD–Pose [*WACV*'23]	*RGB-D*	×	83.2	67.0	34.2	39.4	53.0	64.6	-
*Diff*9*D* ['24]	*RGB-D*	×	76.48	38.16	25.52	30.46	32.48	40.94	-
SAR–Net [*CVPR*'22]	*D*	×	79.3	62.4	31.6	42.3	50.3	68.3	-
GPV–Pose [*CVPR*'22]	*D*	×	83.0	64.4	32.0	42.9	-	73.3	-
GPT–Cope ['24]	*D*	×	82.0	70.4	45.9	53.8	63.1	77.7	79.8
CD–Pose ['24]	*D*	√	81.0	68.6	39.8	44.9	61.8	71.6	-
Our	*RGB-D*	√	88.9	69.1	52.0	60.5	70.2	81.5	82.2

Figure 6

Figure 6 presents a comparison of the 6D object poses estimated by MSPF-LMFF and the GPT-COPE method, along with the NOCS coordinate space generated by our approach. All images are sourced from the NOCS-REAL dataset. The predict-ed poses are indicated by red lines, while the ground truth poses are represented by green lines.

Figure 7

Figure 7 shows the 6D object poses estimated by MSPF-LMFF under complex occlusion conditions. All images are from the NOCS-REAL dataset. The predicted poses are indicated by red lines, while the ground truth poses are represented by green lines.



4.6. Performance on NOCS-CAMERA Da-taset

Table 2 presents the performance of our MSPF-LMFF method compared to 9 state-of-the-art methods on the NOCS-CAMERA dataset [5] NOCS-CAMERA is a synthetic dataset with a large number of training images (~275K), resulting in higher testing accuracy compared to NOCS-REAL. Following the approach in [28], we conducted both training and testing on the NOCS-CAMERA dataset. Our MSPF-LMFF achieved accuracy rates of 92.4%, 84.9%, 70.8%, 73.8%, 83.2%, 88.8%, and 89.9% on the evaluation metrics IoU50, IoU75, 5°2cm, 5°5cm, 10°2cm, 10°5cm, and 10°10cm, respectively. However, it is important to emphasize that MSPF-LMFF does not rely on attention mechanisms; instead, it utilizes a lightweight multi-feature fusion approach. In contrast, CD-POSE and GPT-COPE both incorporate attention mechanisms, which significantly increase computational overhead and inference time.

Overall, the comparative results on the NOCS-REAL and NOCS-CAMERA datasets demonstrate the superiority of our approach. Figure 8 shows some

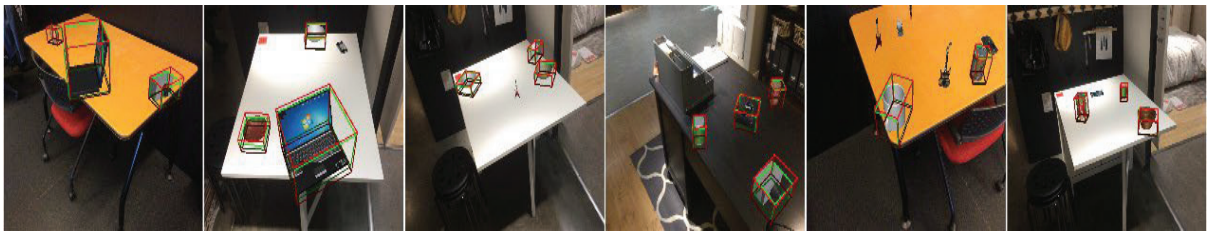
Table 2

The quantitative results of our method and the state-of-the-art (SOTA) methods on the NOCS-CAMERA dataset are presented, evaluated using IoU and n°m cm metrics.

Method	Training Data	prior	NOCS-CAMERA						
			$3D_{50}$	$3D_{75}$	5°2*cm*	5°5*cm*	10°2*cm*	10°5*cm*	10°10*cm*
NOCS [*CVPR*'19]	*RGB-D*	×	83.9	69.5	32.3	40.9	48.2	64.6	-
SPD [*ECCV*'20]	*RGB-D*	√	93.2	83.1	54.3	59.0	73.3	81.5	-
SGPA[*ICCV*'21]	*RGB-D*	√	93.2	88.1	70.7	74.5	82.7	88.4	-
CterSnap [*ICRA*'22]	*RGB-D*	×	92.5	-	-	66.2	-	81.3	-
SD–Pose [*WACV*'23]	*RGB-D*	×	93.5	88.4	64.9	69.1	80.5	86.6	-
*Diff*9*D* ['25]	*RGB-D*	×	79.8	55.8	50.5	57.1	72.1	81.5	-
SAR–Net [*CVPR*'22]	*D*	×	86.8	79.0	66.7	70.9	75.3	80.3	-
GPT–COPE ['24]	*D*	×	92.5	86.9	70.4	76.5	81.3	88.7	89.9
GPV–Pose ['22]	*D*	×	92.9	86.6	67.4	76.2	-	87.4	-
CD–Pose ['24]	*D*	√	82.2	87.7	68.6	73.0	81.6	87.3	-
Our	*RGB-D*	√	92.4	84.9	70.8	73.8	83.2	88.8	89.9

Figure 8

The 6D object poses estimated by MSPF-LMFF. All images are sourced from the NOCS-CAMERA dataset. The predicted poses are represented by red lines, while the ground truth poses are represented by green lines.



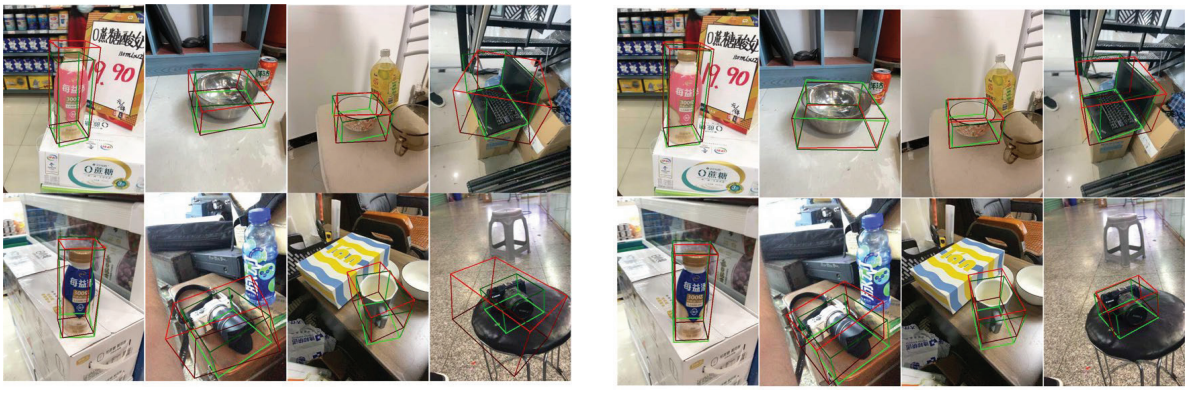
qualitative results on the NOCS-CAMERA dataset. The experimental results indicate that our method not only achieves high accuracy on the synthetic dataset but also improves inference speed, without relying on complex attention mechanisms.

4.7. Performance on wild6d Dataset

To further validate the generalization capability of the MSPF-LMFF model, we directly evaluated the model, originally trained on the NOCS-REAL dataset, on the test set of the Wild6D dataset without any additional finetuning. We compared its performance with five state-of-the-art methods (SPD [33], CR-Net [40], SGPA [3], GPV-Pose [6], and GPT-COPE [50]). The test results are presented in Table 3. Specifically, MSPF-LMFF achieved an mAP of 68.5% under IoU50, outperforming SGPA by 4.9% and GPT-COPE by 2.4%. Additionally, MSPF-LMFF achieved mAPs of 19.3%, 24.2%, and 37.8% under the metrics of 5°2cm, 5°5cm, and 10°5cm, respectively, surpassing SGPA but falling slightly behind GPT-COPE. To provide a more intuitive demonstration of our model's performance. As demonstrated in Figure 9, MSPF-LMFF achieves faster inference speeds than attention-based methods, making it more applicable for real-world deployment. To provide a more intuitive demonstration of the model's performance, we present visual comparisons with SGPA in Figure 9, where MSPF-LMFF generates high-quality 6D object pose estimations even in challenging real-world conditions. The visualization results highlight MSPF-LMFF's superior ability to handle occlusions, clutter, and shape variations compared to SGPA. Additional Comparisons with SGPA in Figure 9: Our vi-

Figure 9

The visualization results on the Wild6D dataset are shown in (a) and (b), which display the results of SGPA and the proposed MSPF-LMFF, respectively. Green and red represent the ground truth and predicted results, respectively. It can be observed that MSPF-LMFF outperforms SGPA.

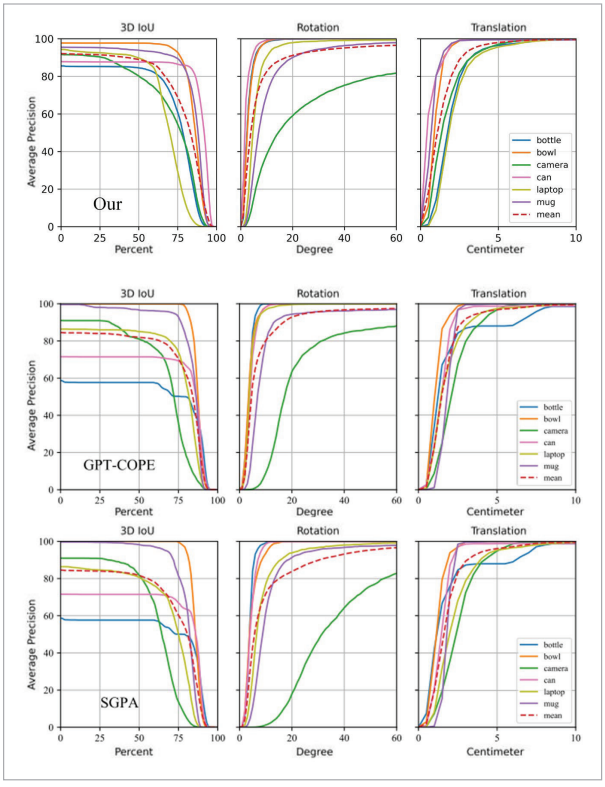


(a) (b)

Table 3
Comparison of our MSPF-LMFF with state-of-the-art methods on the WILD 6D dataset.

Method	prior	Wild6D			
		$3D_{50}$	$5°2cm$	$5°5cm$	$10°5cm$
SPD [*ECCV'20*]	√	32.5	2.6	3.5	13.9
CR–Net [*IRS°21*]	√	49.5	16.1	19.2	36.4
SGPA[*ICCV'21*]	√	63.6	26.2	29.2	39.5
GPV–Pose [*CVPR'22*]	×	67.8	14.1	21.5	41.1
GPT–COPE ['24]	√	66.1	29.8	35.6	42.3
Our	√	68.5	19.3	24.2	37.8

Figure 10
A quantitative comparison of our model with SGPA [3] and GPT-COPE on the NOCS-REAL dataset [39] is presented. The mAP (%) is shown for different thresholds of 3D IoU, rotation, and translation errors.



sualization results show that MSPF-LMFF provides more stable pose predictions, even for objects with complex structures. Unlike SGPA, which struggles with partially occluded objects, MSPF-LMFF leverages prior information more effectively, resulting in fewer pose estimation errors. The superior alignment of predicted poses with ground truth further confirms that our method achieves more accurate and consistent predictions across various object categories. These enhancements and analyses have been incorporated into the revised manuscript, and we sincerely appreciate your insightful suggestions, which have helped us further refine the discussion on generalization capability.

4.8. Comparison of Objects with Different Structures

To further analyze the performance of the MSPF-LMFF model when handling objects with different structures, Figure 8 presents a detailed per-category comparison of MSPF-LMFF with the correspondence-based methods SGPA and GPT-COPE in terms of 3D IoU, rotation accuracy, and translation accuracy. The results show that MSPF-LMFF outperforms SGPA and GPT-COPE in terms of average accuracy for certain metrics and categories, particularly in 3D IoU estimation. Notably, our method demonstrates excellent performance when processing bottle and can instances (represented in blue and pink in the Figure 8, respectively). These instances correspond to relatively simple object categories, where existing methods typically perform well. While MSPF-LMFF achieves comparable performance to SGPA and GPT-COPE when handling complex geometric objects such as cameras, it significantly outperforms both methods on simpler

object categories. This result strongly demonstrates the advantages of our model across different object structures, especially its superior performance when processing simple object categories.

4.9. Ablation Study

To evaluate the effectiveness of the individual components proposed in MSPF-LMFF, we conducted detailed ablation experiments on the NOCS-REAL and NOCS-CAMERA datasets [39]. Starting from a baseline model, we incrementally added the proposed components, including the MSPF block for enriching point cloud feature information and the LMFF block for feature fusion. The baseline model does not include any multi-scale feature fusion module or lightweight multi-feature fusion module. The experimental results are shown in Tables 4 and 5. Compared to the baseline model, the mean average precision (mAP) significantly improved after incorporating the multi-scale feature fusion module, indicating that this module effectively extracts multi-scale point cloud features and helps the model better capture texture information. Subsequently, the mAP further increased after adding the lightweight multi-feature fusion module, although the improvement was relatively modest due to the differences between the fused point cloud and the target point cloud. Finally, when both modules were applied together, the model achieved the highest mAP, demonstrating the best evaluation performance.

a Baseline

Our baseline model is a modified version of T-S [28], where the hidden dimensions in the feature extraction layers are adjusted to accommodate the modified feature dimensions. The training strategy of the baseline model is consistent with that of our MSPF-LMFF. The results in Table 4 show that the baseline model performs poorly in the 6D pose estimation task, serving as a comparative benchmark for the evaluation of subsequent components.

b Effect of MSPF

To evaluate the impact of the MSPF module on model performance, we incorporated PointFeatureNet and ResNet-18 into the baseline model and adjusted the feature dimensions in the MLP layers. The results in Tables 4-5 demonstrate that adding the MSPF module significantly improves model performance. On the NOCS-CAMERA dataset: IoU50 increased by 32.7%, IoU75 increased by 47%, 5°2cm increased by 59.6%, 5°5cm increased by 61%, 10°2cm increased by 69.5%, 10°5cm increased by 68.4%, and 10°10cm in-creased by 68.4%. On the NOCS-REAL dataset: IoU50 increased by 2.7%, IoU75 increased by 1.7%, 5°2cm in-creased by 1.2%, 5°5cm increased by 0.7%, 10°2cm in-creased by 1.1%, 10°5cm increased by 2%, and 10°10cm increased by 0.7%. These results strongly demonstrate the effectiveness of the MSPF module in enhancing the model's ability to capture texture features.

c Effect of LMFF

To assess the efficacy of the LMFF module, we incorporated it into the baseline model and fine-tuned the dimensions of the MLP hidden layers to align with the feature fusion requirements. The integration of the LMFF module led to a notable enhancement in the model's performance, with particularly substantial improvements observed on the NOCS-CAMERA dataset. Specifically, the module yielded increases of 32.3% in IoU50, 42.0% in IoU75, 57.1% in 5°2cm, 57.4% in 5°5cm, 67.1% in 10°2cm, 63.7% in 10°5cm, and 67.7% in 10°10cm. On the NOCS-REAL dataset, the improvements were 0.4% in IoU50, 0.6% in IoU75, 0.9% in 5°2cm, 0.5% in 5°5cm, 1.2% in 10°2cm, 1.1% in 10°5cm, and 0.4%

Table 4

Ablation Study of Different Modules on NOCS-REAL Evaluated Using IoU and n°m cm Metrics.

Method	MSPF	LMFF	NOCS-REAL						
			$3D_{50}$	$3D_{75}$	5°2cm	5°5cm	10°2cm	10°5cm	10°10cm
1	—	—	74.7	58.9	45.9	49.1	66.3	73.1	76.5
2	—	√	75.1	59.5	46.8	49.6	67.5	74.2	76.9
3	√	—	76.0	60.6	47.1	49.8	67.4	75.1	77.2
4	√	√	88.9	69.1	52.0	60.5	70.2	81.5	82.2

Table 5
Ablation Study of Different Modules on NOCS-REAL Evaluated Using IoU and n°m cm Metrics.

Method	MSPF	LMFF	NOCS-CAMERA						
			$3D_{50}$	$3D_{75}$	5°2*cm*	5°5*cm*	10°2*cm*	10°5*cm*	10°10*cm*
1	—	—	59.6	37.5	5.1	7.3	12.6	19.4	20.2
2	—	√	91.9	79.5	62.2	64.7	79.7	83.1	87.9
3	√	—	92.3	84.5	64.7	68.3	82.1	87.8	88.6
4	√	√	92.4	84.9	70.8	73.8	83.2	88.8	89.9

in 10°10cm. These findings underscore the LMFF module's significant contribution to the enhancement of point cloud feature fusion. The data presented in Tables 4-5 further corroborate that the inclusion of the LMFF module markedly boosts the model's performance.

d Effect of PFN (PointFeatureNet)

To evaluate the effectiveness of the PointFeatureNet module, we decomposed the four-layer PointFeatureNet and ResNet-18 modules into 1, 2, 3, and 5 layers, respectively, and compared their performance. The results in Tables 6-7 show that in the NOCS-REAL dataset, the performance of the 1-layer, 2-layer, 3-layer, and 5-layer configurations was inferior to that of the 4-layer configuration. Therefore, we selected the 4-layer configuration as the final validation layer. In the NOCS-CAMERA dataset, the experimental results for the 1-layer and 2-layer configurations were lower than those of the 4-layer configuration. For the 3-layer configuration, the

Table 6
Ablation Study of Different Layers in the PointFeatureNet Module on NOCS-REAL, Represented by IoU and n°m cm.

Method	PFN RESNet	NOCS-REAL						
		$3D_{50}$	$3D_{75}$	5°2*cm*	5°5*cm*	10°2*cm*	10°5*cm*	10°10*cm*
1	PFN1+ResNet1	75.1	59.6	46.8	51.3	67.9	75.2	77.5
2	PFN2+ResNet2	76.9	63.1	49.2	56.5	68.2	76.8	78.1
3	PFN3+ResNet3	80.6	65.5	50.6	58.9	69.5	78.1	81.5
4	PFN4+ResNet4	88.9	69.1	52.0	60.5	70.2	81.5	82.2
5	PFN5+ResNet5	79.1	62.8	43.2	49.9	63.1	75.7	80.1

Table 7
Ablation Study of Different Layers in the PointFeatureNet Module on NOCS-CAMERA, Represented by IoU and n°m cm.

Method	PFN RESNet	NOCS-CAMERA						
		$3D_{50}$	$3D_{75}$	5°2*cm*	5°5*cm*	10°2*cm*	10°5*cm*	10°10*cm*
1	PFN1+ResNet1	62.3	39.7	8.6	10.1	15.3	24.2	26.9
2	PFN2+ResNet2	75.4	46.6	35.5	46.6	54.1	59.3	62.7
3	PFN3+ResNet3	89.2	87.5	62.1	67.4	75.9	85.2	87.6
4	PFN4+ResNet4	92.4	84.9	70.8	73.8	83.2	88.8	89.9
5	PFN5+ResNet5	92.5	85.6	69.2	76.8	81.3	84.6	86.1

3D75 metric was 2.9% higher than that of the 4-layer configuration, while the remaining metrics were lower. In the 5-layer configuration, the 3D75 metric was 0.7% higher, and the 5°5cm metric was 3.0% higher than those of the 4-layer configuration. Consequently, we ultimately selected the 4-layer configuration as the final validation layer.

e Effect of Training Strategy

To validate the effectiveness of different training strategies, we conducted experiments without using prior point clouds, training the model solely on depth maps and images on the NOCS-REAL dataset. The results in Tables 8 and 9 demonstrate that the use of prior point clouds significantly improves the accuracy of pose estimation. On the NOCS-REAL dataset, using prior point clouds increased IoU50 by 6.4%, IoU75 by 7.8%, 5°2cm by 2.3%, 5°5cm by 8.1%, 10°2cm by 2.9%, 10°5cm by 5.8%, and 10°10cm by 0.6%. On the NOCS-CAMERA dataset, improvements were observed in 5°2cm by 5.2%, 10°2cm by 1.5%, 10°5cm by 0.3%, and 10°10cm by 3.8%.

Table 8
Ablation Study of Prior Point Clouds on the NOCS-REAL Dataset, Represented by IoU and n°m cm.

Method	Prior point	NOCS-REAL						
		$3D_{50}$	$3D_{75}$	5°2*cm*	5°5*cm*	10°2*cm*	10°5*cm*	10°10*cm*
1	—	82.5	61.3	49.7	52.4	67.3	75.7	81.6
2	√	88.9	69.1	52.0	60.5	70.2	81.5	82.2

Table 9
Ablation Study of Prior Point Clouds on the NOCS-CAMERA Dataset, Represented by IoU and n°m cm.

Method	Prior point	NOCS-CAMERA						
		$3D_{50}$	$3D_{75}$	5°2*cm*	5°5*cm*	10°2*cm*	10°5*cm*	10°10*cm*
1	—	92.5	86.5	65.6	79.2	81.7	88.5	85.4
2	√	92.4	84.9	70.8	73.8	83.2	88.8	89.9

f Effect of Parameters

To investigate the impact of learning rate, maximum training epochs (Max Epoch), and batch size on model performance, we set different learning rates (0.001, 0.0001, and 0.00001). In the experiments, we fixed the maximum training epochs and batch size and observed the model's performance under different learning rates. According to the experimental results in Table 10, a learning rate of 0.0001 yielded the best performance in terms of convergence speed and final accuracy, achieving a better balance between training stability and performance improvement compared to other learning rate settings.

To evaluate the effect of maximum training epochs, we set different numbers of epochs (4, 8, 12, and 16). In the experiments, we fixed the learning rate and batch size and observed the performance under different maximum training epochs. As shown in Table 11, the model achieved the best performance with a maximum of 16 epochs, exhibiting faster convergence and reaching the highest accuracy.

To assess the influence of batch size, we tested different batch sizes (8, 16, 32, and 64). In the experiments, we fixed the learning rate and maximum training epochs and compared the model's performance under different batch sizes. According to the results in Table 12, a batch size of 16 provided the best performance, enabling faster convergence during training and achieving an optimal balance in final accuracy.

In summary, the model achieved the best performance across all metrics with a learning rate of 0.0001, a maximum of 16 training epochs, and a batch size of 16.

Table 10
Ablation Study of Different Learning Rates on the NOCS-REAL Dataset.

Learning Rate	NOCS-REAL						
	$3D_{50}$	$3D_{75}$	5°2*cm*	5°5*cm*	10°2*cm*	10°5*cm*	10°10*cm*
0.001	65.2	64.4	44.3	46.9	65.4	73.5	71.8
0.0001	88.9	69.1	52.0	60.5	70.2	81.5	82.2
0.00001	74.9	59.6	48.9	51.4	69.5	76.8	79.2

Table 11
Ablation Study of Different Max Epochs on the NOCS-REAL Dataset.

Max Epoch	NOCS-REAL						
	$3D_{50}$	$3D_{75}$	5°2*cm*	5°5*cm*	10°2*cm*	10°5*cm*	10°10*cm*
4	60.7	53.1	43.1	45.5	64.3	72.8	75.4
8	73.7	60.7	50.2	53.2	70.3	77.8	79.2
12	74.9	59.6	48.9	51.4	70.2	76.8	79.6
16	88.9	69.1	52.0	60.5	70.2	81.5	82.2

Table 12
Ablation Study of Different Batch Sizes on the NOCS-REAL Dataset.

Batch size	NOCS-REAL						
	$3D_{50}$	$3D_{75}$	5°2*cm*	5°5*cm*	10°2*cm*	10°5*cm*	10°10*cm*
8	66.8	54.6	42.6	46.2	66.6	75.5	78.2
16	88.9	69.1	52.0	60.5	70.2	81.5	82.2
32	84.5	62.3	52.9	61.2	69.1	78.2	81.4
64	79.6	68.5	49.8	55.1	69.5	79.2	80.5

4.10. Runtime Analysis and Parameters

On a system equipped with an NVIDIA GeForce RTX 4090 GPU, our MSPF-LMFF model achieved an average processing speed of 20 frames per second (FPS) on the NOCS-REAL [39] dataset and 23 FPS on the NOCS-CAMERA [39] dataset. To ensure the fairness and comprehensiveness of the evaluation, we compared the runtime of MSPF-LMFF with three correspondence-based methods (CR-Net [40], SGPA [3], and CD-POSE [51]) on the same machine. We observed that models using attention mechanisms or transformer architectures required 11 hours for a full training cycle, while the MSPF-LMFF model completed training in just 2 hours, significantly reducing training time. As shown in Table 6, our method demonstrated superior inference speed compared to these methods. Notably, CD-POSE exhibited significantly slower runtime on the NOCS-REAL dataset due to its use of attention mechanisms. Additional-ly, CD-POSE experiments were conducted on a less powerful NVIDIA GeForce RTX 2080Ti GPU. Diff9D [23] also showed slower runtime on both the NOCS-REAL and NOCS-CAMERA datasets, primarily due to its reliance on attention mechanisms, which increased computational complexity and impacted

inference speed. Our MSPF-LMFF, running on the more powerful NVIDIA GeForce RTX 4090 GPU and without using attention mechanisms, demonstrated a clear advantage in speed.

Furthermore, our MSPF-LMFF model also exhibited advantages in terms of parameter count. Specifically, MSPF-LMFF has 81.5M parameters, which is fewer than SGPA (89.0M) and CD-POSE (87.3M), and comparable to CR-Net (81.7M), while delivering superior performance. This indicates that our model not only excels in speed but also achieves better efficiency and resource utilization. In summary, MSPF-LMFF optimizes the model architecture to balance speed and parameter count, outperforming existing methods across various experimental settings.

Table 13

Comparison of Inference Time and Parameter Counts for Different Correspondence-Based Methods on the NOCS-REAL and NOCS-CAMERA Datasets [14].

Method	Training Data	SPEED(FPS)		Parameters
		NOCS-REAL	NOCS-CAMERA	
CR–Net ['21]	*RGB-D*	14.0	15.0	81.7M
SGPA[*ICCV*'21]	*RGB-D*	14.0	17.0	89.0M
CD–Pose ['24]	*D*	7.0	-	87.3M
*Diff*9*D* ['25]	*RGB-D*	17.2	17.2	-
Our	*RGB-D*	20.0	23.0	81.5M

4.11. Lightweight Network Comparison

To comprehensively compare the effectiveness of lightweight networks in category-level 6D pose es-timation, we conducted a detailed comparative anal-ysis of the latest lightweight networks, as shown in Tables 14 and 15. Among them, the LFF [44] model uses only depth maps as input, which gives it an advantage in inference speed and parameter count, resulting in lower computational costs and shorter inference times. However, despite its computational efficiency, LFF exhibits significantly lower accuracy on the NOCS-REAL dataset compared to our method. In contrast, our method combines RGB images and depth maps as inputs to fully leverage image texture features and geometric depth information, thereby enhancing the model's expressive power and pose estimation accuracy. Although this fusion strategy increases inference time and parameter count, it achieves a significant improvement in accuracy, making our method superior to LFF in terms of overall performance.

Table 14

Comparison of Inference Time for Different Lightweight Methods on the NOCS-REAL Dataset.

Method	Input	SPEED(FPS)	Parameters
LFF ['23]	*D*	32.0	35.2
Our	*RGB-D*	20.0	81.5M

Table 15

Quantitative Results of Our Method and State-of-the-Art Lightweight Methods on NOCS-REAL, Represented by IoU and n°m cm.

Method	Training Data	prior	NOCS-REAL						
			$3D_{50}$	$3D_{75}$	5°2*cm*	5°5*cm*	10°2*cm*	10°5*cm*	10°10*cm*
LFF ['23]	*D*	√	82.5	70.9	36.6	44.5	61.9	76.9	-
Our	*RGB-D*	√	88.9	69.1	52.0	60.5	70.2	81.5	82.2

5. Conclusion

In this study, we propose the MSPF-LMFF model, which effectively addresses issues such as high computational complexity and large model size in existing 6D object pose estimation methods. Through multi-scale prior point cloud fusion and lightweight multi-feature fusion strategies, MSPF-LMFF not only achieves high accuracy and robustness but also significantly reduces training time and inference latency. Experimental results demonstrate that the proposed method outperforms state-of-the-art methods on multiple datasets. Although significant progress has been made, there are still many potential research directions worthy of further exploration.

First, multimodal learning, as a cutting-edge technology in the field of computer vision, has shown great potential in various application areas. In the future, we plan to further integrate data from different modalities, such as visual, depth, and LiDAR, leveraging these multidimensional information sources to enhance the performance of the MSPF-LMFF model. For instance, in fields such as electricity price prediction [11] and trajectory prediction [29], multimodal fusion has proven to effectively improve prediction accuracy and model generalization capabilities. Additionally, the application of multimodal technologies in power markets and intelligent transportation systems [15] provides important insights for further optimizing the MSPF-LMFF model.

Furthermore, future research will explore integrating the MSPF-LMFF model with multimodal learning frameworks, particularly in handling complex environments and dynamic changes (e.g., joint learning of visual and depth information). By fusing point cloud and image data and leveraging multimodal information, the model's robustness and adaptability in uncertain scenarios can be further enhanced, providing more effective solutions for practical applications such as autonomous driving [17] and robotic grasping [34]. Additionally, we consider incorporating self-supervised learning methods into the MSPF-LMFF model to utilize unlabeled data for further improving the model's generalization capabilities. Combining advanced technologies such as graph neural networks (GNNs) can also enable efficient processing of large-scale point cloud data in large-scale scenarios, demonstrating significant application potential.

Appendix A.

Nomenclature:

I	Cropped red, green, blue (RGB)image.
F_o	Features extracted from the scene point cloud.
F_{rgb}	Features extracted from the RGB image
P_r	Prior point cloud
P_R	Fusion point cloud
x_i	Image features at the ith scale
O_i	Prior Point cloud features at the ith scale
F_r	Features extracted from the prior point cloud
F_t	Features generated by Prior adaptation
$F_{rgb,o}$	Features generated by F_{rgb} and F_o
F_{tM}	F_t Features generated by MLP
$F_{(rgb,o)M}$	$F_{(rgb,o)}$ Features generated by MLP
$F_{f,r}$	Features generated by F_r, F_{tM} and $F_{(rgb,o)M}$ concatenation
$F_{f,c}$	Features generated by Fo, FtM and $F_{(rgb,o)M}$ concatenation
F_d	Features generated by $F_{f,r}$ and $F_{f,c}$ subtraction
G_R	$F_{f,r}$ Features generated by pool
G_d	F_d Features generated by pool
G_c	$F_{f,c}$ Features generated by pool
F_D	Features generated by G_R , G_d , G_c and $F_{f,r}$ and concatenation
F_M	Features generated by G_R , G_d , G_c and $F_{f,c}$ and concatenation
F	Features formed by concatenating a prior point cloud features and image features and then stitching them together
N_o	Number of Scene Point Clouds
N_r	Number of a prior point clouds before deformation
N_R	Number of deformed a fusion point clouds

Acknowledgement

This work was supported in part by Scientific and Technological Research Program of Chongqing Mu-nicipal Education Commission (KJQN202301517, KJQN202301543), in part by Shanxi Province Applied Basic Research Program, China (No.202203021211116).

References

1. Castaño, A., González, P., González, J. A., Molina, A., Serra-no, M. A., Garijo, M. Matching Distributions Algorithms Based on the Earth Mover's Distance for Ordinal Quantification. IEEE Transactions on Neural Networks and Learning Systems, 2022, 35(1), 1050-1061. doi: 10.1109/TNNLS.2022.3179355. https://doi.org/10.1109/TNNLS.2022.3179355
2. Chen, H., Tian, W., Wang, P., Wang, F., Xiong, L., Li, H. EPRO-PNP: Generalized End-to-End Proba-bilistic Perspective-N-Points for Monocular Object Pose Estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024, Early Ac-cess, 1-12. doi: 10.1109/TPAMI.2024.3354997. https://doi.org/10.1109/TPAMI.2024.3354997
3. Chen, K., Dou, Q. SGPA: Structure-Guided Prior Adaptation for Category-Level 6D Object Pose Es-timation. Proceedings of the IEEE In-ternational Conference on Computer Vision (ICCV), 2021, 2773-2782. doi: 10.1109/ICCV51070.2023.00317. https://doi.org/10.1109/ICCV51070.2023.00317
4. Chen, W., Jia, X., Chang, H. J., Chen, H., Zhou, Y., Li, S., Hu, R., Lu, C. FS-Net: Fast Shape-Based Network for Category-Level 6D Object Pose Estimation with Decoupled Rota-tion Mechanism. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, 1581-1590. doi: 10.1109/TGRS.2016.2584107. https://doi.org/10.1109/TGRS.2016.2584107
5. Dang, Z., Wang, L., Guo, Y., Salzmann, M. Learn-ing-Based Point Cloud Registration for 6D Object Pose Estimation in the Real World. Proceedings of the European Conference on Computer Vision (ECCV), 2022, 19-37. doi: 10.48550/arXiv.2203.15309. https://doi.org/10.1007/978-3-031-19769-7_2
6. Di, Y., Zhang, R., Lou, Z., Manhardt, F., Tombari, F., Ji, X. GPV-Pose: Category-Level Object Pose Esti-mation via Geometry-Guided Point-Wise Voting. Proceedings of the IEEE/CVF Conference on Com-puter Vision and Pattern Recognition, 2022, 6781-6791. doi: 10.1109/CVPR52688.2022.00666. https://doi.org/10.1109/CVPR52688.2022.00666
7. Fang, H.-S., Wang, C., Gou, M., Lu, C. GraspNet-1Billion: A Large-Scale Benchmark for General Ob-ject Grasping. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020, 11444-11453. doi: 10.1109/CVPR42600.2020.01146. https://doi.org/10.1109/CVPR42600.2020.01146
8. Fu, Y., Wang, X. Category-Level 6D Object Pose Estimation in the Wild: A Semi-Supervised Learning Approach and a New Da-taset. Advances in Neural Information Pro-cessing Systems, 2022, 35, 27469-27483. doi: 10.48550/arXiv.2206.15436.
9. He, K., Gkioxari, G., Dollár, P., Girshick, R. Mask R-CNN. Proceedings of the IEEE Interna-tional Con-ference on Computer Vision (ICCV), 2017, 2980-2988. doi: 10.1109/ICCV.2017.322. https://doi.org/10.1109/ICCV.2017.322
10. He, K., Zhang, X., Ren, S., Sun, J. Deep Re-sidual Learning for Image Recognition. Pro-ceedings of the IEEE Conference on Computer Vi-sion and Pattern Recognition (CVPR), 2016, 770-778. doi: 10.1109/CVPR.2016.90. https://doi.org/10.1109/CVPR.2016.90
11. He, M., Jiang, W., Gu, W. TriChronoNet: Advanc-ing Electricity Price Prediction with Multi-Module Fusion. Applied Energy, 2024, 371, 123626. doi: 10.1016/j.apeergy.2024.123626. https://doi.org/10.1016/j.apenergy.2024.123626
12. He, Y., Fan, H., Chen, Q., Sun, J. FFB6D: A Full Flow Bidirectional Fusion Network for 6D Pose Esti-mation. Proceedings of the IEEE Con-ference on Computer Vision and Pattern Recognition (CVPR), 2021, 3003-3013. doi: 10.1109/CVPR46437.2021.00302. https://doi.org/10.1109/CVPR46437.2021.00302
13. He, Y., Sun, W., Huang, H., Liu, J., Fan, H., Sun, J. PVN3D: A Deep Point-Wise 3D Key-points Voting Network for 6DoF Pose Estima-tion. Proceedings of the IEEE Conference on Computer Vision and Pat-tern Recognition (CVPR), 2020, 11632-11641. doi: 10.1109/CVPR42600.2020.01165. https://doi.org/10.1109/CVPR42600.2020.01165
14. Jiang, H., Dang, Z., Gu, S., Xie, J., Salz-mann, M., Yang, J. Center-Based Decoupled Point-Cloud Reg-istration for 6D Object Pose Es-timation. Proceed-ings of the IEEE International Conference on Com-puter Vision (ICCV), 2023, 3427-3437. doi: 10.1109/ICCV51070.2023.00317. https://doi.org/10.1109/ICCV51070.2023.00317
15. Jiang, W., Zhang, Y., Han, H., Li, D., Chen, Q., Wang, P. Mobile Traffic Prediction in Con-sumer Applications: A Multimodal Deep Learn-ing Ap-proach. IEEE Transactions on Consumer Electron-ics, 2024, 70(1), 3425-3435. doi: 10.1109/TCE.2024.3361037. https://doi.org/10.1109/TCE.2024.3361037

16. Kingma, D. P., Ba, J. Adam: A Method for Sto-chastic Optimization. Proceedings of the In-ternational Conference on Learning Representa-tions (ICLR), 2015, 1-9. doi: 10.48550/arXiv.1412.6980.

17. Li, D., Zhang, J., Liu, G. Autonomous Driv-ing Decision Algorithm for Complex Multi-Vehicle Inter-actions: An Efficient Approach Based on Global Sorting and Local Gaming. IEEE Transactions on Intelligent Transportation Systems, 2024, 25(7), 6927-6937. doi: 10.1109/TITS.2023.3346048. https://doi.org/10.1109/TITS.2023.3346048

18. Li, Z., Ji, X. Pose-Guided Auto-Encoder and Feature-Based Refinement for 6-DoF Object Pose Regression. Proceedings of the IEEE In-ternational Conference on Robotics and Auto-mation (ICRA), 2020, 8397-8403. doi: 10.1109/ICRA40945.2020.9196953. https://doi.org/10.1109/ICRA40945.2020.9196953

19. Lin, F., Liu, H., Zhou, H., Wang, X., Chen, Y., Zhang, J. Loss Distillation via Gradient Matching for Point Cloud Completion with Weighted Chamfer Distance. IEEE/RSJ Interna-tional Conference on Intel-ligent Robots and Systems (IROS), 2024, 511-518. doi: 10.1109/IROS58592.2024.10801828. https://doi.org/10.1109/IROS58592.2024.10801828

20. Lin, J., Wei, Z., Li, Z., Chen, H., Zhang, Y., Liu, W. DualPoseNet: Category-Level 6D Ob-ject Pose and Size Estimation Using Dual Pose Network with Re-fined Learning of Pose Con-sistency. Proceedings of the IEEE/CVF Interna-tional Conference on Com-puter Vision (ICCV), 2021, 3560-3569. doi: 10.1109/ICCV48922.2021.00354. https://doi.org/10.1109/ICCV48922.2021.00354

21. Lin, J., Wei, Z., Zhang, Y., Li, Z., Chen, H., Liu, W. VI-Net: Boosting Category-Level 6D Object Pose Estimation via Learning Decoupled Rotations on the Spherical Representations. Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2023, 14001-14011. doi: 10.1109/ICCV51070.2023.01287 https://doi.org/10.1109/ICCV51070.2023.01287

22. Lin, X., Yang, W., Gao, Y., Chen, H., Zhang, L., Wang, J. Instance-Adaptive and Geometric-Aware Keypoint Learning for Category-Level 6D Object Pose Estimation. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024, 21040-21049. doi: 10.1109/CVPR52733.2024.01988. https://doi.org/10.1109/CVPR52733.2024.01988

23. Liu, J. Diff9D: Diffusion-Based Domain-Generalized Category-Level 9-DoF Object Pose Es-timation. IEEE Transactions on Pattern Anal-ysis and Machine Intelligence, 2025. doi: 10.1109/TPAMI.2025.3552132. https://doi.org/10.1109/TPAMI.2025.3552132

24. Liu, J., Chen, Y., Ye, X., Zhang, L., Wang, H., Li, P. IST-Net: Prior-Free Category-Level Pose Estimation with Implicit Space Transfor-mation. Proceedings of the IEEE/CVF Interna-tional Conference on Com-puter Vision (ICCV), 2023, 13978-13988. doi: 10.1109/ICCV51070.2023.01285. https://doi.org/10.1109/ICCV51070.2023.01285

25. Liu, J., Sun, W., Liu, C., Zhang, X., Fan, S., Wu, W. HFF6D: Hierarchical Feature Fusion Network for Robust 6D Object Pose Tracking. IEEE Transactions on Circuits and Systems for Video Technology, 2022, 32(11), 7719-7731. doi: 10.1109/TCSVT.2022.3181597. https://doi.org/10.1109/TCSVT.2022.3181597

26. Liu, P., Zhang, Q., Zhang, J., Wang, F., Cheng, J. MFPN-6D: Real-Time One-Stage Pose Estimation of Objects on RGB Images. Proceed-ings of the IEEE International Conference on Robotics and Au-tomation (ICRA), 2021, 12939-12945. doi: 10.1109/ICRA48506.2021.9561878. https://doi.org/10.1109/ICRA48506.2021.9561878

27. Liu, X., Wang, G., Li, Y., Zhao, H., Chen, Q., Sun, W. CATRE: Iterative Point Clouds Alignment for Cat-egory-Level Object Pose Re-finement. European Conference on Computer Vision (ECCV), 2022, 499-516. doi: 10.1007/978-3-031-20086-1_29. https://doi.org/10.1007/978-3-031-20086-1_29

28. Liu, Y., Sun, W., Liu, C., Zhang, X., Fu, Q. Robotic Continuous Grasping System by Shape Transform-er-Guided Multi-Object Category-Level 6-D Pose Estimation. IEEE Transactions on Industrial Infor-matics, 2023, 32(10), 6728-6740. doi: 10.1109/TII.2023.3244348. https://doi.org/10.1109/TII.2023.3244348

29. Lu, Y., Wang, W., Bai, R., Zhang, H., Li, M. Hyper-Relational Interaction Modeling in Mul-ti-Modal Tra-jectory Prediction for Intelligent Connected Vehicles in Smart Cities. Information Fusion, 2025, 114, 102682. doi: 10.1016/j.inffus.2024.102682. https://doi.org/10.1016/j.inffus.2024.102682

30. Peng, S., Zhou, X., Liu, Y., Lin, H., Huang, Q., Bao, H. PVNet: Pixel-Wise Voting Network for 6DoF Object Pose Estimation. IEEE Transac-tions on Pat-tern Analysis and Machine Intelli-gence, 2022, 44, 3212-3223. doi: 10.1109/TPAMI.2020.3047388. https://doi.org/10.1109/TPAMI.2020.3047388

31. Qi, C. R., Yi, L., Su, H., Guibas, L. J. Point-Net++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. Proceedings of the Advances in Neural In-

formation Processing Systems (NeurIPS), 2017, 30, 5105-5114. doi: 10.48550/arXiv.1706.02413.

32. Sundermeyer, M., Marton, Z-C., Durner, M., Brucker, M., Triebel, R. Implicit 3D Orienta-tion Learning for 6D Object Detection from RGB Images. Proceedings of the European Con-ference on Computer Vision (ECCV), 2018, 699-715. doi: 10.48550/arXiv.1902.01275. https://doi.org/10.1007/978-3-030-01231-1_43
33. Tian, M., Ang, M. H., Lee, G. H. Shape Pri-or De-formation for Categorical 6D Object Pose and Size Estimation. Proceedings of the Euro-pean Confer-ence on Computer Vision (ECCV), 2020, 530-546. doi: 10.1007/978-3-030-58589-1_32. https://doi.org/10.1007/978-3-030-58589-1_32
34. Tong, L., Song, K., Tian, H., Li, J., Zhao, Y., Chen, P., Wang, X. A Novel RGB-D Cross-Background Ro-bot Grasp Detection Dataset and Background-Adaptive Grasp Network. IEEE Transactions on In-strumentation and Measure-ment, 2024. doi: 10.1109/TIM.2024.3413164. https://doi.org/10.1109/TIM.2024.3413164
35. Tummala, S., Kadry, S., Bukhari, S. A. C., Al-Ahmari, A., El-Sappagh, S., Hussain, S. Clas-sification of Brain Tumor from Magnetic Reso-nance Imaging Using Vision Transformers En-sembling. Current Oncology, 2022, 29(10), 7498-7511. doi: 10.3390/curroncol29100590. https://doi.org/10.3390/curroncol29100590
36. Umeyama, S. Least-Squares Estimation of Transformation Parameters Between Two Point Patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1991, 13(4), 376-380. doi: 10.1109/34.88573. https://doi.org/10.1109/34.88573
37. Wang, C., Xu, D., Zhu, Y., Martín-Martín, R., Lu, C., Fei-Fei, L., Savarese, S. DenseFusion: 6D Object Pose Estimation by Iterative Dense Fusion. Pro-ceedings of the IEEE Conference on Computer Vi-sion and Pattern Recognition (CVPR), 2019, 3343-3352. doi: 10.1109/CVPR.2019.00346. https://doi.org/10.1109/CVPR.2019.00346
38. Wang, G., Manhardt, F., Tombari, F., Ji, X. GDR-Net: Geometry-Guided Direct Regression Network for Monocular 6D Object Pose Estima-tion. Pro-ceedings of the IEEE Conference on Computer Vi-sion and Pattern Recognition (CVPR), 2021, 16611-16621. doi: 10.1109/CVPR46437.2021.01634. https://doi.org/10.1109/CVPR46437.2021.01634
39. Wang, H., Sridhar, S., Huang, J., Mottaghi, R., Chen, W., Liao, X., Bai, Y. Normalized Ob-ject Coor-dinate Space for Category-Level 6D Object Pose and Size Estimation. Proceedings of the IEEE/CVF Conference on Computer Vi-sion and Pattern Recognition (CVPR), 2019, 2642-2651. doi: 10.1109/CVPR.2019.00275. https://doi.org/10.1109/CVPR.2019.00275
40. Wang, J., Chen, K., Dou, Q. Category-Level 6D Object Pose Estimation via Cascaded Rela-tion and Recurrent Reconstruction Networks. Proceedings of the IEEE/RSJ International Con-ference on Intelli-gent Robots and Systems (IROS), 2021, 4807-4814. doi: 10.1109/IROS51168.2021.9636212. https://doi.org/10.1109/IROS51168.2021.9636212
41. Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bron-stein, M. M., Solomon, J. M. Dynamic Graph CNN for Learning on Point Clouds. ACM Transactions on Graphics, 2019, 38(5), 1-12. doi: 10.1145/3326362. https://doi.org/10.1145/3326362
42. Wu, Y., Zand, M., Etemad, A., Greenspan, M. Vote from the Center: 6DoF Pose Estimation in RGB-D Images by Radial Keypoint Voting. Proceed-ings of the European Conference on Computer Vi-sion (ECCV), 2022, 335-352. doi: 10.1007/978-3-031-20080-9_20. https://doi.org/10.1007/978-3-031-20080-9_20
43. Yang, B., Wang, X., Li, Y., Chen, Q., Zhao, L. Mo-dality Fusion Vision Transformer for Hy-perspectral and LiDAR Data Collaborative Classification. IEEE Journal of Selected Topics in Applied Earth Obser-vations and Remote Sensing, 2024. doi: 10.1109/JSTARS.2024.3415729. https://doi.org/10.1109/JSTARS.2024.3415729
44. Yang, H., Sun, W., Liu, J., Fan, H., Zhang, X., Wu, W. Depth-Based Lightweight Feature Fusion Net-work for Category-Level 6D Pose Estimation. 2023 China Automation Congress (CAC), 2023, 393-398. doi: 10.1109/CAC59555.2023.10451707. https://doi.org/10.1109/CAC59555.2023.10451707
45. Zhang, R., Di, Y., Lou, Z., Manhardt, F., Tombari, F., Ji, X. RBP-Pose: Residual Bound-ing Box Projec-tion for Category-Level Pose Es-timation. Proceed-ings of the European Confer-ence on Computer Vi-sion (ECCV), 2022, 655-672. doi: 10.1007/978-3-031-19769-7_38. https://doi.org/10.1007/978-3-031-19769-7_38
46. Zhang, Z., Liu, M., Shen, J., Wang, Y., Li, H., Chen, Q. Lightweight Whole-Body Human Pose Es-timation with Two-Stage Refinement Training Strat-egy. IEEE Transactions on Hu-man-Machine Sys-tems, 2024, 54(1), 121-130. doi: 10.1109/THMS.2024.3349652. https://doi.org/10.1109/THMS.2024.3349652
47. Zheng, L., Tse, T. H. E., Wang, C., Li, Y., Zhang, Q., Sun, Y. GeoReF: Geometric Align-ment Across Shape Variation for Category-Level Object Pose Refinement. Pro-

ceedings of the IEEE/CVF Confer-ence on Computer Vision and Pattern Recognition (CVPR), 2024, 10693-10703. doi: 10.1109/CVPR52733.2024.01017. https://doi.org/10.1109/CVPR52733.2024.01017

48. Zheng, L., Wang, C., Sun, Y., Li, H., Huang, Z., Gu, N., Zhang, Q. HS-Pose: Hybrid Scope Feature Extraction for Category-Level Object Pose Estima-tion. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023, 17163-17173. doi: 10.1109/CVPR52729.2023.01646. https://doi.org/10.1109/CVPR52729.2023.01646
49. Zheng, Y., Jiang, W. Evaluation of Vision Transformers for Traffic Sign Classification. Wireless Communications and Mobile Compu-ting, 2022, 2022(1), 3041117. doi: 10.1155/2022/3041117. https://doi.org/10.1155/2022/3041117
50. Zou, L., Huang, Z., Gu, N., Liu, X., Wang, W., Li, Y. GPT-COPE: A Graph-Guided Point Transformer for Category-Level Object Pose Es-timation. IEEE Transactions on Circuits and Systems for Video Technology, 2023. doi: 10.1109/TCSVT.2023.3309902. https://doi.org/10.1109/TCSVT.2023.3309902
51. Zou, L., Huang, Z., Gu, N., Liu, X., Wang, W., Li, Y. Learning Geometric Consistency and Discrepan-cy for Category-Level 6D Object Pose Estimation from Point Clouds. Pattern Recogni-tion, 2024, 145, 109896. doi: 10.1016/j.patcog.2023.109896. https://doi.org/10.1016/j.patcog.2023.109896
52. Zou, L., Huang, Z., Gu, N., Wang, G. 6D-ViT: Cat-egory-Level 6D Object Pose Estimation via Trans-former-Based Instance Representation Learning. IEEE Transactions on Image Pro-cessing, 2022, 31, 6907-6921. doi: 10.1109/TIP.2022.3216980. https://doi.org/10.1109/TIP.2022.3216980