PoseLecTr: A Novel Convolution and Attention Mechanism-Based Model for 6D Object Pose Estimation

Alexander Du
Interlake High School
s-dual@bsd405.org

Yingwu Zhu
Seattle University
zhuy@seattleu.edu

Abstract

6D object pose estimation from RGB images is a key task in computer vision. This is challenging because RGB images lack depth information, requiring deducing the third dimension from a 2D projection. Additionally, objects may be occluded by other objects in the scene, complicating the task greatly. In this work, we introduce PoseLecTr, a new model for 6D object pose estimation. Unlike traditional models that rely on grid-based structures such as Convolutional Neural Networks, PoseLecTr uses a Graph Neural Network approach. In the graph, the spatial-temporal features extracted from individual pixels serve as nodes, allowing for more detailed spatial interactions. We also introduce Legendre convolution, a novel feature that utilizes Legendre polynomials to approximate convolution kernels. This design ensures filter orthogonality and improves numerical stability in the system. Lastly, we introduce advanced spatial-attention and self-attention distillation mechanisms to expedite the spatial-temporal features extraction process while reducing redundancy. We conduct extensive experiments on the YCB-Video dataset to show that PoseLecTr outperforms nine other state-of-the-art models in pose estimation accuracy.

1. Introduction

In the fields of computer vision and robotics, precise object identification and determination of their 3D position and orientation (x, y, z, θ, ψ, ϕ) are crucial, particularly for tasks such as autonomous driving, augmented reality, and industrial automation. Traditionally, solving the six degrees of freedom (6-DOF) pose estimation challenge involved matching feature points between model projections and actual images taken from various camera viewpoints, followed by applying techniques like Perspective-n-Point along with coordinate transformations to determine the object's pose in the physical environment, as outlined by Collet, Martinez, and Srinivasa in 2011. However, this method heavily depends on clear textures to correctly identify feature points. The advent of depth-sensing technology helped address issues posed by objects without distinctive textures, as discussed by Brachmann et al. in 2014. Later developments, such as PoseCNN by Xiang et al. in 2017, have successfully incorporated deep learning into the 6-DOF pose estimation process, making significant strides by leveraging large datasets and complex feature analysis techniques, allowing for effective pose estimation using only RGB images.

Nevertheless, many deep learning techniques, particularly those based on convolutional neural networks (CNNs), often fail to capture the dependencies between extracted features due to their inherent grid-based data structure. To overcome these limitations, we propose a new graph-based encoder-decoder architecture called PoseLecTr. PoseLecTr's innovation lies in representing image features as graph nodes and modeling the relationships between these nodes using an adjacency matrix. This framework forms the basis of a graph neural network (GNN), mapping both dependencies and independencies among features. Additionally, the model enhances feature selection by integrating spatial attention mechanisms and self-attention distillation, coupled with a Legendre convolution technique that utilizes Legendre polynomials to improve accuracy during the convolution process.

2. Related Works

The goal of 6D object pose estimation is to precisely determine both the orientation and location of objects in 3D space, which is an essential capability for applications in robotics and computer vision (Xu et al. 2022). This research focuses on estimating 6-DOF poses using RGB images exclusively. With the rise of deep learning, the field has seen advancements that enable precise pose predictions from a single RGB image. Various feature-based methods have been explored (Zhu et al. 2021; Zhao et al. 2020; You et al. 2021), which leverage local features like keypoints

and edges, extracted either from specific regions or the entire image, to align with 3D models, establishing accurate 2D-3D correspondences (Marullo et al. 2023). In addition, Peng et al. (2019) introduced the Pixel-wise Voting Network (PVNet), which improves robustness against occlusion by utilizing pixel-level vectors to identify keypoint positions, thereby enhancing pose estimation accuracy.

Other researchers have investigated template-based methods (Ulrich, Wiedemann, and Steger 2011; Payet and Todorovic 2011; Massa, Russell, and Aubry 2016), where a database of templates is created from 3D models in an offline phase, which are then matched against current images in an online phase to estimate the pose (Marullo et al. 2023). Sundermeyer et al. (2020) presented an RGB-based framework utilizing an augmented autoencoder for real-time object detection and pose estimation, adept at handling object symmetries and sensor variations without requiring annotated pose data.

Other approaches have focused on direct prediction methods using convolutional neural networks (CNNs), which require extensive labeled training data but have enabled significant advancements in predicting object positions and orientations (Rad and Lepetit 2017; Oberweger, Rad, and Lepetit 2018; Josifovski et al. 2018; Xu and Chen 2018; Periyasamy et al. 2020). These deep learning models often include additional pose refinement steps using Perspective-n-Point (PnP) techniques (Marullo et al. 2023). Xiang et al.'s PoseCNN (2017) deals with 3D translations by detecting object centers and estimating distances from the camera, while 3D rotations are handled by regressing quaternions, using a unique loss function to accommodate symmetrical objects.

Recent studies have increasingly turned to non-Euclidean data, represented as graphs that capture complex interrelationships (Wu et al. 2020). Foundational work in graph neural networks (GNNs), such as Recurrent graph neural networks (RecGNNs) (Scarselli et al. 2008; Gallicchio and Micheli 2010; Li et al. 2015), set the stage for current approaches. Convolutional graph neural networks (ConvGNNs) extended traditional convolutional methods to graph structures, aggregating features from nodes and their neighbors to create more advanced representations (Wu et al. 2020). Li et al. (2018a) introduced a spectral method that adjusts the graph structure during training, while Chiang et al. (2019) proposed Cluster-GCN, which improves the efficiency of training large graphs through graph clustering techniques. Unsupervised learning methods like graph autoencoders (GAEs) (Cao, Lu, and Xu 2016; Yu et al. 2018; Li et al. 2018b) have pushed the boundaries of GNNs by encoding graphs into latent spaces and reconstructing them (Wu et al. 2020). In contrast, attention-based spatial graph convolutional networks such as ASTGCN (Guo et al. 2019) effectively model traffic patterns using both spatial-temporal attention and convolutional layers. Despite these innovations, most research still depends on grid-based CNNs for feature extraction, which may not capture higher-level feature correlations. Graph-based neural networks offer a promising approach for improving 6D pose estimation by constructing representations based on semantic similarities.

Our key contributions are as follows:

- **Graph-Based Representation**: We introduce a novel graph-based representation for image features, enabling more flexible and expressive feature interactions compared to traditional grid-based CNNs.
- **Legendre Convolution**: We propose a novel Legendre convolution layer, which significantly improves numerical stability compared to standard graph convolutional layers.
- **Spatial-Attention and Self-Attention Distillation**: We incorporate advanced spatial-attention and self-attention distillation techniques to enhance feature extraction and focus on critical spatial-temporal information.

3. Methodology

In this section, we introduce various notations and formalize the problem of 6D object pose estimation from RGB images.

We represent the image as a set of pixels arranged in a grid structure, which can be denoted as a graph $G(V, \varepsilon, A)$. Here, $V = \{v_1, v_2, v_3, \ldots, v_N\}$ is the set of N nodes corresponding to pixels, $\varepsilon \subseteq V \times V$ is a set of edges representing the connections between these nodes, and A is the adjacency matrix of the graph $G(V, \varepsilon, A)$. The adjacency matrix A is constructed based on the cosine similarity of feature vectors derived from the image, capturing the relational information between different regions of the image.

Let I_t represent the input RGB image at time t. The goal is to estimate the 6D pose, which includes the 3D rotation and 3D translation, of the object in the image. The pose can be represented as a tensor P_t, capturing the pose information across different images.

The architecture we introduce, named **PoseLecTr**, is composed of several key components: a feature extraction stage, a Legendre-based convolution layer, a spatiotemporal encoding layer, and a final decoding layer. In the first stage, the feature extraction mechanism transforms the raw image data into complex high-dimensional features. The convolution process in the second layer is approximated through

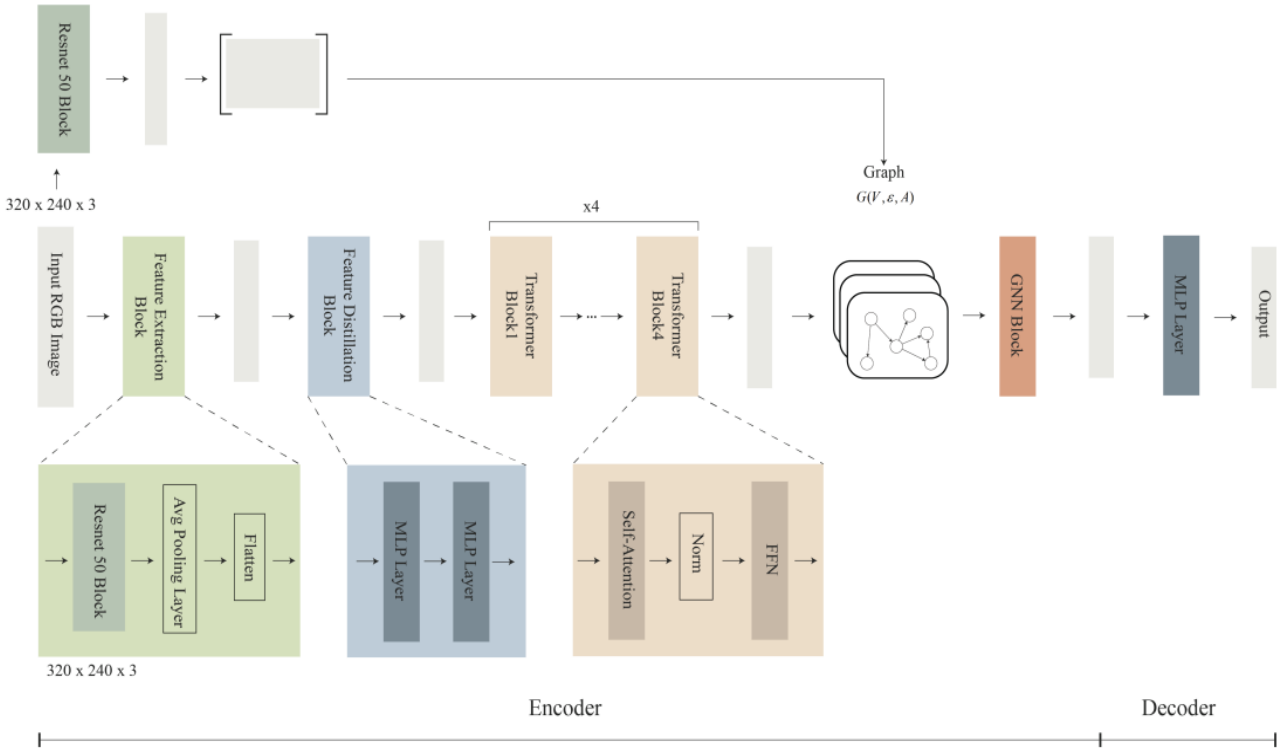

Figure 1. Architecture of **PoseLecTr**

the use of Legendre polynomials, chosen for their properties of orthogonality, numerical stability, and recursive efficiency. The spatiotemporal encoder, which uses a self-attention mechanism, is responsible for capturing intricate spatiotemporal relationships, essential for interpreting both the object's movement and spatial structure. The decoder is tasked with producing the resulting 6D pose estimation. Figure 1 illustrates the PoseLecTr framework, which includes the feature extraction module, several spatiotemporal encoding layers stacked in sequence, and the final output layer. A detailed explanation of each component follows.

3.1. Feature Extraction Layer

The feature extraction layer is modified to utilize the third convolutional block of the ResNet50 network, which helps reduce redundant features, compress the model size, and decrease the number of parameters. Specifically, the image is processed through the initial layers of ResNet50, up to the third convolutional block, to extract meaningful feature maps. By using this intermediate layer, as opposed to deeper layers, the model retains essential spatial and structural information while reducing unnecessary details, thereby decreasing computational complexity and avoiding overfitting. The extracted feature maps are then flattened and passed through a fully connected layer to obtain the final embedding, with an embedding dimension. The feature extraction layer converts the raw input image I_t into high-dimensional feature representations. This is achieved through a series of convolutional layers followed by fully connected layers. Specifically, the image is passed through convolutional layers to extract feature maps, which are then flattened and further processed through a fully connected layer to obtain the embedding, denoted as F_t with an embedding dimension d:

$$F_t = FC(Flatten(Conv(I_t))) \tag{1}$$

where $F_t \in R^{N \times d}$ represents the feature matrix, N is the number of pixels, and d is the embedding dimension.

3.2. Legendre Convolution Layer

The data embedding layer transforms the input into a high-dimensional representation. Initially, the raw input $\chi = (X_1, X_2, \ldots, X_T) \in R^{T \times N \times c}$ is processed through a fully connected layer to obtain $\chi = (X_1, X_2, \ldots, X_T) \in R^{T \times N \times d}$, where d is embedding dimension, T is the number of input, N is the number of pixels, c is the image channels. Subsequently, a spatial-temporal embedding mechanism is designed to integrate essential knowledge into the model. This mechanism includes the spatial graph Laplacian embedding for encoding the spatial relationships within the image and the temporal embedding for capturing the dynamic changes in object pose over time.

To characterize the interconnections between nodes in the graph, we utilize the regularized Laplacian matrix denoted as $L = I - D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$, where A represents the adjacency matrix, I is the identity matrix, and D is the matrix of node degrees in diagonal form. As L is a symmetric positive semidefinite matrix, its eigenvectors are orthogonal in pairs, and the eigenvalues are distinct and non-negative. Therefore, L can be simplified to the form of:

$$L = U\,\Delta\,U^T \tag{2}$$

where U is a matrix composed of eigenvectors and Δ is a matrix composed of eigenvalues.

In the Graph Convolutional Network (GCN) model, Y_{output} is define as:

$$Y_{output} = \sigma\left(U\ \mathrm{g}_\theta(\Delta)\,U^T\chi\right) \tag{3}$$

This formula represents a layer in the standard GCN, where $\sigma()$ denotes the activation function, and θ is the parameter value for iterative updates. Here, $\mathrm{g}_\theta(\Delta)$ signifies the convolution kernel operating on a graph.

The number of convolution parameters is equivalent to the number of graph nodes, denoted as n. For large graphs, the computational efficiency of this method is significantly diminished. Additionally, the algorithm necessitates the computation of $\mathrm{g}_\theta(\Delta)U^T$ each time, resulting in a high time complexity of $O(n^3)$.

To address these challenges, we employ Legendre polynomials to approximate the convolution kernel. Legendre polynomials are defined as follows:

$$P_0(x) = 1$$

$$P_n(x) = \frac{1}{2^n \times n!} \cdot \frac{d^n}{dx^n}[(x^2-1)^n],\ n = 1, 2, \ldots \tag{4}$$

There are several advantages to utilizing Legendre polynomials for approximation:
1. Orthogonality: Legendre polynomials of different orders exhibit zero inner product, enabling independent calculation of each coefficient during polynomial fitting or series expansion, thereby streamlining the computation process.
2. Numerical stability: Legendre polynomials typically demonstrate robust numerical stability in the fitting process, mitigating the occurrence of Runge's phenomenon.
3. Recursion: Legendre polynomials feature a straightforward recursive relationship, facilitating efficient generation of higher-order Legendre polynomials.

$$P_0(x) = 1 \tag{5}$$

$$P_1(x) = x \tag{6}$$

$$(n{+}1)P_{n+1}(x) = (2n{+}1)xP_n(x) - nP_{n-1}(x),\ n = 1, 2\ldots \tag{7}$$

After replacing the convolution kernel $\mathrm{g}_\theta(\Delta)$ with the Legendre polynomial, a new convolution kernel form is obtained:

$$\mathrm{g}_\theta(\Delta) = \sum_{k=0}^{n-1} \alpha_k P_k(\widetilde{\Delta}) \tag{8}$$

3.3. Encoder Layer

We have developed a encoder layer based on a self-attention mechanism to capture the intricate and dynamic dependencies required for accurate 6D object pose estimation from RGB images. The core of the encoder layer comprises two main components: the **spatial feature self-attention module** and the **self-attention distillation module**. These modules work together to effectively model the complex relationships present in the data. These modules collectively enable the encoder layer to effectively understand and model complex relationships in embedding data.

$$Q \in R^{L_Q \times d}, \quad K \in R^{L_K \times d}, \quad V \in R^{L_V \times d}$$

and d is input dimension. The self-attention mechanism has demonstrated effectiveness in various scenarios.

3.4. Spatial Features Self-attention

We design spatial feature self-attention to capture dynamic spatial features in RGB images for accurate 6D object pose estimation. Formally, for input $\chi = (X_1, X_2, \ldots, X_T) \in R^{T \times N \times d}$, the $query$, key, and $value$ matrices of self-attention operations can be represented as:

$$Q_t = X_t\ \mathrm{g}\ W_Q$$

$$K_t = X_t\ \mathrm{g}\ W_K$$

$$V_t = X_t\ \mathrm{g}\ W_V$$

where $W_Q, W_K, W_V \in R^{d \times d'}$ are learnable parameters and d' is the dimension of the $query$, key, and $value$ matrices. The self-attention mechanism SA can be expressed as:

$$SA\,(Q, K, V) = Softmax(\frac{QK^T}{\sqrt{d}})\,V \tag{9}$$

As can be seen from the formula, the spatial dependencies between nodes are different in different X_i.

The key to the attention mechanism is the mapping function, which encodes the relative importance of the input elements, mapping the values to probabilities. The commonly used Softmax function generates dense attention. However, in the task of 6D pose estimation from RGB images, the distribution of feature relevance can be highly imbalanced across different regions of the image. Therefore, we design

a new spatial feature attention module SFA, utilizing the Sparsemax function [21], which is expressed as:

$$SFA\,(Q, K, V) = Sparsemax(\frac{QK^T}{\sqrt{d}})\,V \qquad (10)$$

In contrast to the Softmax function, the Sparsemax function can generate sparse probability distributions. Sparsemax has the ability to selectively emphasize the most significant features while disregarding less relevant ones, thereby offering clearer decision boundaries. Moreover, as not all image regions exhibit high correlation, Sparsemax reduces the model's sensitivity to noise by assigning zero probability to these less relevant regions. Furthermore, when dealing with complex image data for 6D pose estimation, Sparsemax is more adept at handling the long-tail effects, ensuring the model focuses on the most critical spatial features.

3.5. Self-attention Distillation

While we employed the Sparsemax function to minimize the redundancy in feature mapping within the encoder, certain redundant features persist. To address this, we utilize a self-attention distillation module to identify the crucial features with dominance and create a centralized self-focused feature map in the subsequent layer. This module effectively reduces the dimensionality of the input features, with the distillation process progressing from layer l to layer $l+1$ as follows:

$$X_{l+1} = AvgPool(ReLU(Conv1d([X_l]_{SFA}))) \qquad (11)$$

where $[\,]_{SFA}$ represents the spatial features self-attention block, $Conv1d()$ denotes a one-dimensional convolution operation. This includes multi-head spatial features self-attention and basic operations, where $Conv1d(\cdot)$ executes a one-dimensional convolution filter (kernel width = 3) in the time dimension using the $ReLU(\cdot)$ activation function. Following the stacking of one layer, we introduce an average-pooling layer with a stride of 3. Ultimately, the final feature representation of the encoder is obtained.

3.6. Abbreviations and Acronyms

Our methodology incorporates a traditional decoding architecture comprising two sequentially layered multi-headed attention mechanisms. Additionally, a bypass route, facilitated by 1 × 1 convolutional operations, is applied after each encoder layer to convert the outputs, labeled as χ_{output}, into a bypassed format $\chi_{sk} \in R^{T\times N\times d_{sk}}$, with d_{sk} representing the bypassed dimensionality.

4. Experiments

4.1. Dataset Description

In our experiment, we tested PoseLecTr against 9 other SoTA frameworks on the YCB-Video [48] dataset. In our ablation studies, we evaluated our contributions against three different datasets: LINEMOD [46], Occluded LINEMOD [47], and YCB-Video [48]. These datasets present a variety of challenging scenarios, including dense clutter, multiple objects, both static and dynamic environments, desktop configurations, and robotic applications. The objects featured across these datasets display a wide range of attributes, such as being textureless, glossy, symmetrical, and of different sizes.

We investigated a model-free approach for the task of 6D pose estimation. In the model-free setup, we selected reference images from the dataset's training portion that contain novel objects, with ground-truth pose annotations available. Throughout all tests, except for the ablation studies, we maintained the same trained model and settings for inference, with no further fine-tuning applied.

4.2. Experimental Details

Baselines: We compared PoseLecTr to the following 9 baseline models that fall into three categories:

A. CNN-Based Models PoseCNN[48], DenseFusion[49], PVNet[50], PoseRBPF[51], DeepIM[52]. PoseCNN directly predicts 3D translation and rotation from RGB images, setting benchmarks in integrating RGB and depth data for precise 6D pose estimation. DenseFusion combines dense pixel-wise information from both RGB and depth maps, ensuring highly accurate 6D pose predictions through fusion of both modalities. PVNet focuses on regressing 2D keypoints from images and utilizes a PnP algorithm to compute the final pose, ensuring keypoint localization for improved accuracy. PoseRBPF uses recurrent neural networks combined with particle filters to manage temporal dynamics in pose estimation, enhancing robustness against occlusion and temporal inconsistencies. DeepIM refines an initial pose estimate iteratively by aligning rendered images with observed ones, forming a feedback loop that boosts pose estimation accuracy across iterations.

B. GNN-based Models GC-Pose[53] utilizes graph convolutional networks to model spatial relationships between object parts as a graph, extracting features from RGB images to capture the geometric structure of the object. G2LNet[54] employs a GNN to learn spatial relationships between keypoints, using these to predict the 6D pose of the object. GC-Pose and G2LNet leverage GNNs to model spatial relationships between object parts. GC-Pose constructs a graph of object parts, while G2LNet learns spatial relationships between detected keypoints for pose prediction.

C. Self-attention-based Models TransPose[55] uses a transformer-based architecture to capture long-range dependencies in the image data, employing self-attention mechanisms to focus on relevant image regions for 6D pose prediction. PoseFormer[56] combines CNNs with trans-

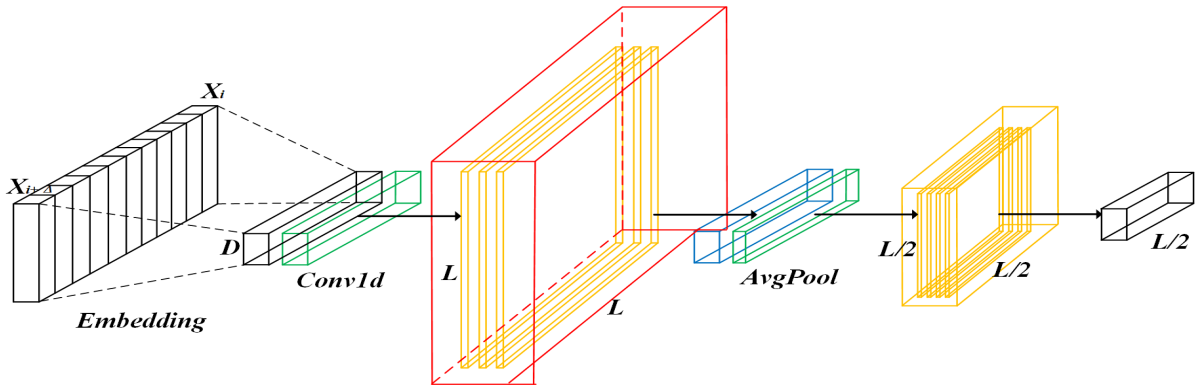

Figure 2. Illustrates the structure within the encoder of the **PoseLecTr**. (1) This diagram represents one of several identical sections of the encode. (2) The primary section shown processes the entire input sequence. Subsequently, the next section handles only half of the input segments, and this pattern continues in the following sections. (3) The output of the encoder is formed by merging the feature maps from all sections.

Object\Method	PoseCNN	DenseFusion	PVNet	PoseRBPF	DeepIM	GC-Pose	G2LNet	TransPose	PoseFormer	**PoseLecTr**
002_master_chef_can	89	93.9	89.3	86.5	90.2	91.5	92.7	94	95.2	**96.4**
003_cracker_box	88.5	96.5	87.8	85	89.9	91	92.5	94.7	95.8	**97.8**
004_sugar_box	94.3	**97.6**	91.2	89.7	92.3	93.5	94.8	96	95.2	97.5
005_tomato_soup_can	89.1	95	88.9	86.8	90.7	91.9	93.2	94.5	95.7	**98.5**
006_mustard_bottle	92	95.8	90.5	88.6	91.8	93	94.2	95.5	96.7	**97.3**
007_tuna_fish_can	78	86.5	79.3	77.5	81.8	83.2	84.5	85.8	87.9	**91.2**
008_pudding_box	81.4	**97.9**	83.7	80.9	84.2	85.6	87.1	89.3	90.5	91.8
009_gelatin_box	90.4	97.8	92.1	89.8	93	94.2	95.6	96.7	97.9	**98.7**
010_potted_meat_can	81.7	77.8	80.3	78.2	82.1	83.5	84.9	86.3	87.5	**90.2**
011_banana	82.6	94.9	84.5	83.2	86.5	87.8	89.1	91.3	92.6	**96.9**
All Frames	86.66	93.37	86.76	84.62	88.25	89.52	90.86	92.41	93.61	**95.63**

Table 1. Pose estimation accuracy of RGB-D methods evaluated using ADD on the YCB-Video dataset.

formers, using CNNs for local feature extraction and transformers to capture global context, providing high-precision 6D pose estimations. TransPose and PoseFormer integrate transformers to capture dependencies and contextual information. TransPose focuses on relevant regions for pose prediction using self-attention, while PoseFormer combines CNNs for local features with transformers for global context.

4.3. Hyper-parameter Setting

Our proposed methods are optimized using the Adam optimizer, with the learning rate initialized at 1e-4 and decaying by half every epoch. The total number of epochs is set to 8 with appropriate early stopping. The comparison methods are configured as recommended, with a batch size of 32.

4.4. Evaluation Metrics

In our research, we applied two distinct indicators[48] to measure performance:

$$ADD = \frac{1}{m} \sum_{x \in M} \|(Rx + t) - (\widehat{R}x + \widehat{t})\| \quad (12)$$

$$ADDS = \frac{1}{m} \sum_{x_1 \in M} \min_{x_2 \in M} \|(Rx_1 + t) - (\widehat{R}x_2 + \widehat{t})\| \quad (13)$$

The ADD metric quantifies the accuracy of the predicted attitude by measuring the mean Euclidean distance between the corresponding three-dimensional points of the model. If the average distance between these points is less than 10%, the posture is considered correct. Similarly, for symmetric objects, where multiple gestures may correspond to the same physical configuration, ADD-S metrics are used. Instead of calculating point-to-point distance, ADD-S measures the average minimum distance between each transformation point in the predicted attitude and the nearest point on the model transformed by the ground true state. If this average minimum distance is less than 10%, the posture is considered correct.

Object\Method	PoseCNN	DenseFusion	PVNet	PoseRBPF	DeepIM	GC-Pose	G2LNet	TransPose	PoseFormer	**PoseLecTr**
002_master_chef_can	85.3	92.2	87.1	84.4	89.6	89.6	89	93.7	92.5	**93.9**
003_cracker_box	86.6	94.5	85.2	83.6	87.2	88.2	89.1	93.2	**95**	**95**
004_sugar_box	91.9	95.7	87.3	87.8	91.2	92	91.7	95.8	**96.8**	95.5
005_tomato_soup_can	87.3	94.5	87.8	83.6	89.9	90.1	90	93.2	93.6	**96.7**
006_mustard_bottle	91.6	91.9	89.2	86.1	90.3	92.3	92.6	95	93.5	**96.3**
007_tuna_fish_can	77.1	83	75.7	75.2	78.6	81	82.7	85.6	85.9	**88**
008_pudding_box	77.5	**96.6**	83.2	79.3	81.7	85.1	85.2	87.3	89	89
009_gelatin_box	90	94.9	88.6	89.7	91	93	93.2	93.7	95.5	**95.8**
010_potted_meat_can	77.9	74.4	78.9	76.9	79.9	80.2	83.5	82.6	85.1	**88.2**
011_banana	80.1	93.5	81.1	79.6	84.5	84.4	88.2	88.1	89.4	**95.2**
All Frames	84.53	91.12	84.41	82.62	86.39	87.59	88.52	90.82	91.63	**93.36**

Table 2. Pose estimation accuracy of RGB-D methods evaluated using ADD-S on the YCB-Video dataset.

4.5. Experiment Platform

Every test was carried out using a computer equipped with an NVIDIA GeForce 4080 GPU and 64GB of RAM. $PoseLecTr$ was executed under Ubuntu 18.04, utilizing PyTorch 1.16.10 and Python 3.9.0.

4.6. Performance Comparison

In Table 1, comparisons with benchmark models on various datasets are presented. Bolding indicates the best performance, while underlining indicates the second best performance.

For the object $002\_master\_chef\_can$, PoseLecTr outperforms all methods in these two tables, achieving 96.4% and 93.9% accuracy in Tables 1 and 2. On $003\_cracker\_box$, PoseLecTr consistently shows superior performance of 97.8% and 95%, while DenseFusion again follows with 96.5% and 94.5%, respectively. For more challenging objects, such as $007\_tuna\_fish\_can$, PoseLecTr is a clear improvement over other methods, with a clear gap in both tables: 91.2% in Table 1 and 88% in Table 2. When considering the overall performance of all frames, PoseLecTr shows the highest accuracy in both tables, scoring 95.63% in Table 1 and 93.36% in Table 2. This shows that PoseLecTr consistently delivers the most accurate results across a variety of object types and frameworks. TransPose and PoseFormer follow closely behind, in Table 1 and Table 2, TransPose is 92.41% and 90.82% respectively, while PoseFormer is 93.61% and 91.63% respectively.

4.7. Ablation Study

To further investigate the efficacy of PoseLecTr's components, we conducted a comparative analysis with several modifications of PoseLecTr, that results in three different variants:

PoseLecTr+: In this variant, Legendre convolution is replaced by Chebyshev convolution to verify the effect of different modes of graph convolution on the result.

PoseLecTr*: This version removes the mechanism of spatial features self-attention, resulting in a scenario where each node in the graph is influenced by every other node, leading to a uniform distribution of attention across the network.

PoseLecTr#: This adjustment excludes the module dedicated to distilling self-attention, impeding the processing of extended data, in cases where the graph size is substantial.

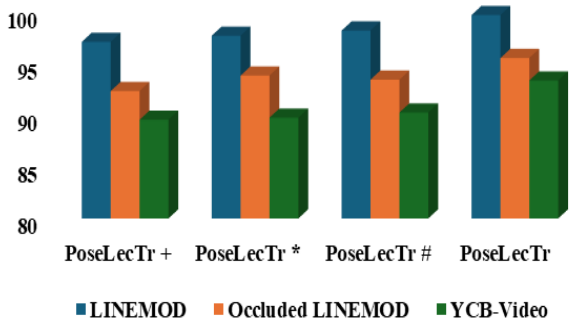

Figure 3. Impact of **PoseLecTr** variants on performance in LINEMOD, Occluded LINEMOD and YCB-Video.

Figure 3 shows how these variants compare on the LINEMOD datasets, where PoseLecTr+ is labeled as A, PoseLecTr* is labeled as B, PoseLecTr# is labeled as C, and PoseLecTr is labeled as D.

Based on these results, we can draw the following conclusions:

1. **Effect of Graph Convolution Type (PoseLecTr+)**: Replacing Legendre convolution with Chebyshev convolution in PoseLecTr+ results in slightly lower performance for all data sets. On the LINEMOD dataset, accuracy dropped from 99.7% (PoseLecTr) to 96.4%, with similar declines on Occluded LINEMOD (94.34% vs. 95.54%) and YCB-Video (91.56% vs. 93.36%). This shows that Legendre convolution captures local and global contexts more efficiently than Chebyshev convolution.

2. **Effects of Spatial Feature Self-attention (PoseLecTr*)**: When the spatial self-attention mechanism is removed from PoseLecTr*, the accuracy of the Occluded LINEMOD (92.54%) and YCB-Video (89.96%) datasets decreases more significantly. The importance of this feature to occlusion and complex scene processing is highlighted. However, the performance of the LINEMOD dataset (96.9%) remained relatively robust, suggesting that the absence of spatial self-attention affects more challenging datasets to a greater extent, where spatial dependence is more critical.

3. **Effects of Self-attentional Distillation (PoseLecTr#)**: Excluding self-attentional distillation from PoseLecTr# modules results in performance degradation, especially when blocking LINEMOD (92.64%) and YCB-Video (89.56%). This suggests that self-focused distillation contributes to processing larger graph sizes and extended data. Without this component, the model's ability to handle a wide range of spatial relationships is compromised, affecting its ability to generalize in more complex environments.

5. Conclusion

In this paper, we presented **PoseLecTr**, a novel graph-based framework for 6D object pose estimation from RGB images. By incorporating Legendre convolution and attention mechanisms, PoseLecTr advances both accuracy and efficiency in pose estimation tasks. The introduction of Legendre convolution, with its orthogonality and numerical stability properties, streamlines the convolution process and enables more efficient feature extraction compared to traditional convolution methods. Additionally, the spatial feature self-attention and self-attention distillation modules enhance the model's ability to focus on critical spatial-temporal features, thus improving the robustness and accuracy of pose estimation. Our evaluation on benchmark datasets such as LINEMOD and YCB-Video demonstrates that PoseLecTr consistently outperforms existing state-of-the-art models, particularly in complex scenes with textureless, symmetric, or highly occluded objects.

References

[1] Amini, A.; Selvam Periyasamy, A.; and Behnke, S. YOLOPose: Transformer-based multi-object 6D pose estimation using keypoint regression. In *International Conference on Intelligent Autonomous Systems*, 392–406, Springer, 2022.

[2] Atwood, J.; and Towsley, D. Diffusion-convolutional neural networks. *Advances in neural information processing systems*, 29, 2016

[3] Brachmann, E.; Krull, A.; Michel, F.; Gumhold, S.; Shotton, J.; and Rother, C. Learning 6d object pose estimation using 3d object coordinates. In *Computer Vision–ECCV 2014: 13th European Conference*, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part II 13, 536–551. Springer, 2014.

[4] Bruna, J.; Zaremba, W.; Szlam, A.; and LeCun, Y. Spectral networks and locally connected networks on graphs. arXiv preprint arXiv:1312.6203, 2013.

[5] Cao, S.; Lu, W.; and Xu, Q. Deep neural networks for learning graph representations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

[6] Chen, W.; Jia, X.; Chang, H. J.; Duan, J.; and Leonardis, A. G2l-net: Global to local network for real-time 6d pose estimation with embedding vector features. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4233–4242, 2020.

[7] Chiang, W.-L.; Liu, X.; Si, S.; Li, Y.; Bengio, S.; and Hsieh, C.-J. Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 257–266, 2019.

[8] Collet, A.; Martinez, M.; and Srinivasa, S. S. The moped framework: Object recognition and pose estimation for manipulation. *The international journal of robotics research*, 30(10): 1284–1306, 2011.

[9] Dai, H.; Kozareva, Z.; Dai, B.; Smola, A.; and Song, L. Learning steady-states of iterative algorithms over graphs. In *International conference on machine learning*, 1106–1114. PMLR, 2018.

[10] De Cao, N.; and Kipf, T. MolGAN: An implicit generative model for small molecular graphs. arXiv preprint arXiv:1805.11973, 2018.

[11] Deng, X.; Mousavian, A.; Xiang, Y.; Xia, F.; Bretl, T.; and Fox, D. PoseRBPF: A Rao–Blackwellized particle filter for 6-D object pose tracking. *IEEE Transactions on Robotics*, 37(5): 1328–1342, 2021.

[12] Gallicchio, C.; and Micheli, A. Graph echo state networks. In *The 2010 international joint conference on neural networks (IJCNN)*, 1–8. IEEE, 2010.

[13] Guo, S.; Lin, Y.; Feng, N.; Song, C.; and Wan, H. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, 922–929, 2019.

[14] Henaff, M.; Bruna, J.; and LeCun, Y. Deep convolutional networks on graph-structured data. arXiv preprint arXiv:1506.05163, 2015.

[15] Hinterstoisser, S.; Lepetit, V.; Ilic, S.; Holzer, S.; Bradski, G.; Konolige, K.; and Navab, N. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *Computer Vision–ACCV 2012: 11th Asian Conference on Computer Vision*, Daejeon, Korea, November 5-9, 2012, Revised Selected Papers, Part I 11, 548–562. Springer, 2013.

[16] Josifovski, J.; Kerzel, M.; Pregizer, C.; Posniak, L.; and Wermter, S. Object detection and pose estimation based on convolutional neural networks trained with synthetic data. In *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, 6269–6276. IEEE, 2018.

[17] Li, R.; Wang, S.; Zhu, F.; and Huang, J. Adaptive graph convolutional neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018a.

[18] Li, Y.; Tarlow, D.; Brockschmidt, M.; and Zemel, R. Gated graph sequence neural networks. arXiv preprint arXiv:1511.05493, 2015.

[19] Li, Y.; Vinyals, O.; Dyer, C.; Pascanu, R.; and Battaglia, P. Learning deep generative models of graphs. arXiv preprint arXiv:1803.03324, 2018b.

[20] Li, Z.; Li, X.; Chen, S.; Du, J.; and Li, Y. SaMfENet: Self-Attention Based Multi-Scale Feature Fusion Coding and Edge Information Constraint Network for 6D Pose Estimation. *Mathematics*, 10(19): 3671, 2022.

[21] Martins, A.; and Astudillo, R. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *International conference on machine learning*, 1614–1623. PMLR, 2016

[22] Marullo, G.; Tanzi, L.; Piazzolla, P.; and Vezzetti, E. 6D object position estimation from 2D images: A literature review. *Multimedia Tools and Applications*, 82(16): 24605–24643, 2023.

[23] Massa, F.; Russell, B. C.; and Aubry, M. Deep exemplar 2d-3d detection by adapting from real to rendered views. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 6024–6033, 2016.

[24] Micheli, A. Neural network for graphs: A contextual constructive approach. *IEEE Transactions on Neural Networks*, 20(3): 498–511, 2009.

[25] Oberweger, M.; Rad, M.; and Lepetit, V. Making deep heatmaps robust to partial occlusions for 3d object pose estimation. In *Proceedings of the European conference on computer vision (ECCV)*, 119–134, 2018.

[26] Payet, N.; and Todorovic, S. From contours to 3d object detection and pose estimation. In *2011 International Conference on Computer Vision*, 983–990. IEEE, 2011.

[27] Peng, S.; Liu, Y.; Huang, Q.; Zhou, X.; and Bao, H.

Pvnet: Pixel-wise voting network for 6dof pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4561–4570, 2019.

[28] Periyasamy, A. S.; Capellen, C.; Schwarz, M.; and Behnke, S. ConvPoseCNN2: prediction and refinement of dense 6D object pose. In *International Joint Conference on Computer Vision, Imaging and Computer Graphics*, 353–371. Springer, 2020.

[29] Rad, M.; and Lepetit, V. Bb8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth. In *Proceedings of the IEEE international conference on computer vision*, 3828–3836, 2017.

[30] Scarselli, F.; Gori, M.; Tsoi, A. C.; Hagenbuchner, M.; and Monfardini, G. The graph neural network model. *IEEE transactions on neural networks*, 20(1): 61–80, 2008.

[31] Sundermeyer, M.; Marton, Z.-C.; Durner, M.; and Triebel, R. Augmented autoencoders: Implicit 3d orientation learning for 6d object detection. *International Journal of Computer Vision*, 128(3): 714–729, 2020.

[32] Ulrich, M.; Wiedemann, C.; and Steger, C. Combining scale-space and similarity-based aspect graphs for fast 3D object recognition. *IEEE transactions on pattern analysis and machine intelligence*, 34(10): 1902–1914, 2011.

[33] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[34] Wang, C.; Xu, D.; Zhu, Y.; Mart´ın-Mart´ın, R.; Lu, C.; Fei-Fei, L.; and Savarese, S. Densefusion: 6d object pose estimation by iterative dense fusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 3343–3352, 2019.

[35] Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; and Philip, S. Y. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1): 4–24, 2020.

[36] Wu, Z.; Pan, S.; Long, G.; Jiang, J.; and Zhang, C. Graph wavenet for deep spatial-temporal graph modeling. arXiv preprint arXiv:1906.00121, 2019.

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

[38] Xu, B.; and Chen, Z. Multi-level fusion based 3d object detection from monocular images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2345–2353, 2018.

[39] Xu, Y.; Lin, K.-Y.; Zhang, G.; Wang, X.; and Li, H. Rnnpose: Recurrent 6-dof object pose refinement with robust correspondence field estimation and pose optimization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 14880–14890, 2022.

[40] Yan, S.; Xiong, Y.; and Lin, D. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[41] You, J.-K.; Hsu, C.-C. J.; Wang, W.-Y.; and Huang, S.-K. Object pose estimation incorporating projection loss and discriminative refinement. *IEEE Access*, 9: 18597–18606, 2021.

[42] Yu, W.; Zheng, C.; Cheng, W.; Aggarwal, C. C.; Song, D.; Zong, B.; Chen, H.; and Wang, W. Learning deep network representations with adversarially regularized autoencoders. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2663–2671, 2018.

[43] Zhao, H.; Wei, S.; Shi, D.; Tan, W.; Li, Z.; Ren, Y.; Wei, X.; Yang, Y.; and Pu, S. Learning symmetry-aware geometry correspondences for 6d object pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 14045–14054, 2023.

[44] Zhao, W.; Zhang, S.; Guan, Z.; Zhao, W.; Peng, J.; and Fan, J. Learning deep network for detecting 3d object keypoints and 6d poses. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, 14134–14142, 2020.

[45] Zhu, Y.; Wan, L.; Xu, W.; and Wang, S. ASPP-DF-PVNet: atrous Spatial Pyramid Pooling and Distance-Filtered PVNet for occlusion resistant 6D object pose estimation. *Signal Processing: Image Communication*, 95: 116268, 2021.

[46] Stefan Hinterstoisser, Stefan Holzer, Cedric Cagniart, Slobodan Ilic, Kurt Konolige, Nassir Navab, and Vincent Lepetit. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In *International Conference on Computer Vision (ICCV)*, pages 858–865, 2011.

[47] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6D object pose estimation using 3d object coordinates. In *13th European Conference on Computer Vision (ECCV)*, pages 536–551, 2014. 6, 7

[48] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes. In *Robotics: Science and Systems (RSS)*, 2018. 2, 6, 7

[49] Wang, Yisheng, et al. DenseFusion: 6D Object Pose Estimation by Iterative Dense Fusion. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[50] Peng, Song, et al. "PVNet: Pixel-wise Voting Network for 6DoF Pose Estimation." *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[51] Drost, Bertram, et al. "PoseRBPF: A Rao-Blackwellized Particle Filter for 6D Object Pose Estima-

tion." *Proceedings of Robotics: Science and Systems (RSS)*, 2017.
[52] Li, Yisheng, et al. "DeepIM: Deep Iterative Matching for 6D Pose Estimation." *European Conference on Computer Vision (ECCV)*, 2018.
[53] Bogdan, Panait, et al. "GC-Pose: Geometric and Correspondence-Based Graph Neural Network for 6D Pose Estimation." *Proceedings of IEEE Conference on Robotics and Automation (ICRA)*, 2020.
[54] Wen, Bowen, et al. "G2LNet: Global to Local Networks for Real-time 6D Pose Estimation with Embeddings." *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.
[55] Yang, Zhenyu, et al. "TransPose: Transformer for 6D Object Pose Estimation." *Proceedings of IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.
[56] Zhou, Yufei, et al. "PoseFormer: When Transformers Meet 6D Object Pose Estimation." *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
[57] P. Yin and J. Ye and G. Lin and Q. Wu "Graph neural network for 6D object pose estimation" *Knowledge-Based Systems*, Vol 218, 2021.
[58] C. Zheng and S. Zhu and M. Mendieta and T. Yang and C. Chen and Z. Ding "3D Human Pose Estimation with Spatial and Temporal Transformers" *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.