PWDF: Patch-Wise Dense Fusion for 6D Pose Estimation

Emanuele Rosapepe
s346705

Gabriele Santona
s343402

Niccolo Antonelli-Dziri
s350296

Omar Hisham Elsayed Wafaay Wafaay
s355114

Politecnico di Torino

Abstract

Accurate 6D object pose estimation is essential for robotic manipulation in cluttered environments. While state-of-the-art methods like DenseFusion achieve high accuracy by processing unstructured 3D point clouds, this approach introduces significant computational overhead. We propose Patch-Wise Dense Fusion (PWDF), a lightweight architecture that operates directly on dense RGB-D feature embeddings, avoiding expensive 3D reconstruction. Our method employs dual ResNet-18 backbones to extract spatially aligned appearance and geometric features from cropped RGB and depth images, which are fused at the feature level and processed through task-specific prediction heads with learned attention-based aggregation. We evaluate 4 architectural variants on the LineMOD dataset: a geometric baseline, an explicitly masked fusion approach, and an implicit no-mask fusion variant. Our experiments demonstrate that the no-mask approach achieves 98.45% *accuracy on an* $80/20$ *split while offering superior computational efficiency by eliminating dependency on instance segmentation. On the standard 15/85 benchmark split, our masked variant achieves* 92.6% *mean ADD accuracy without synthetic data or iterative refinement, outperforming DenseFusion's per-pixel variant by* 6.4%*. These results establish PWDF as a compelling alternative for real-time embedded robotic applications requiring efficient and accurate pose estimation.*

Project link:
https://github.com/Manursp0022/6D_Pose_Estimation_light

1. Introduction

Accurate 6D object pose estimation determining the 3D translation and 3D rotation of an object relative to the camera is a cornerstone capability for intelligent robotic systems. Whether for robotic bin-picking, assembly tasks, or human–robot interaction, the ability to perceive the exact geometric state of an object in cluttered environments is a strict prerequisite for successful manipulation [3, 15]. Despite significant progress enabled by the proliferation of RGB-D sensors, the task remains challenging due to sensor noise, severe occlusions, and varying lighting conditions.

State-of-the-art methods, such as PoseCNN [13] and DenseFusion [12], have set the benchmark for performance in this domain. However, these approaches often rely heavily on converting depth maps into 3D point clouds to perform complex geometric reasoning (e.g., voting schemes or iterative refinement). While effective, processing unstructured point clouds introduces a significant computational workload. In this work, we aim to explore lightweight, crop-based architectures that minimize this computational load by operating directly on dense feature embeddings from RGB and depth patches, avoiding the need for expensive full-scene 3D reconstruction.

We evaluate our proposed approaches on a subset [2] of the LineMOD dataset [12]. Unlike standard benchmarks that utilize the split proposed in the dataset files (`train.txt`, `test.txt`), we adopt a randomized $80/20$ split strategy. Our study progresses through three distinct architectural evolutions, decoupling the detection module from the pose prediction module:

- **Geometric Baseline:** A decoupled approach utilizing a ResNet backbone for rotation regression and a geometric pinhole projection model for translation based on 2D bounding box centroids.
- **Explicitly Masked Fusion:** Inspired by the DenseFusion architecture, this approach incorporates explicit segmentation masks provided by an upstream instance segmentor (YOLOv8-Seg)[10, 11]. Unlike DenseFusion, the masks are directly multiplied with RGB and depth features and are not processed by an additional pixel-wise fusion network.

- **Implicit (No-Mask) Fusion:** A streamlined architecture that relies solely on 2D detection bounding boxes (YOLOv8-Detect)[10, 11], removing the dependency on pixel-wise segmentation masks.

In both fusion-based extensions, RGB and depth crops are processed through parallel backbones and merged via a dense feature fusion network.

Our experiments reveal that the No-Mask approach achieves the same accuracy as the masked version (reaching 98.45%), while offering superior system-level efficiency. We demonstrate that relying on explicit YOLO-based segmentation introduces a dual bottleneck: the higher computational latency of instance segmentation networks and the performance degradation caused by noisy mask boundaries. By removing the mask dependency, the proposed pipeline proves to be both more robust and faster, making it an ideal candidate for real-time embedded robotic applications.

2. Related work

RGB-based Pose Estimation. Early deep learning approaches focused on estimating the 6D pose directly from single RGB images. Methods such as PoseCNN [13] decouple the task into object center localization, depth prediction, and quaternion regression for rotation. Similarly, SSD-6D[6] and YOLO-6D[8] extend 2D detection architectures to predict 3D bounding boxes. While these methods are computationally efficient, estimating 3D translation and metric depth from a monocular image is inherently ill-posed and suffers from scale ambiguity. Consequently, they often rely on expensive post-processing steps, such as Iterative Closest Point (ICP) [1] or render-and-compare refinement techniques (e.g., DeepIM [7]), to achieve acceptable accuracy. In contrast, our approach natively integrates depth information to resolve geometric ambiguities without requiring iterative refinement.

RGB-D and Point Cloud Methods. To leverage geometric information, recent state-of-the-art methods lift depth maps into 3D point clouds. PointFusion [14] and DenseFusion [12] process RGB and point cloud features separately using CNNs and PointNet-like architectures[3], respectively, before performing pixel-wise fusion. MaskedFusion [9] further extends this paradigm by incorporating explicit segmentation masks to filter non-object points. While these methods achieve high accuracy on standard benchmark splits, processing unstructured point clouds consisting of N points requires computationally intensive gathering and grouping operations. Our work differs fundamentally by treating depth as a dense, structured representation. We employ a dense feature fusion strategy on aligned feature grids, avoiding the overhead of unstructured point cloud processing while preserving local geometric structure.

Pose Refinement and Mask Dependency. Achieving high-precision pose estimation often involves iterative refinement or a strong dependency on instance segmentation. DeepIM [7] iteratively refines the pose by matching the observed image with a rendered view of the object, incurring high latency due to repeated rendering and inference passes. Furthermore, methods such as MaskedFusion [9] are heavily dependent on the quality of upstream segmentation; errors at object boundaries can irreversibly corrupt the extracted geometric features. Our Implicit (No-Mask) architecture challenges this dependency, demonstrating that for dense interpolation tasks, the network can learn to regress accurate poses directly from detection crops. This removes the bottleneck of instance segmentation and significantly improves system-level efficiency.

3. Model

3.1. Baseline Model

Our initial baseline approach relies on a decoupled architecture for estimating the 6D pose (3D translation and 3D rotation) of objects within the scene.

Baseline: Diameter-based Pinhole Model The pipeline begins with a **YOLO**[10] detector to extract 2D bounding boxes for each object. For 3D translation estimation, we implement a modified version of the pinhole camera model. Instead of the standard depth-based projection, this variant utilizes the known physical diameter of the object to back-project its 3D position (X, Y, Z) from the image coordinates.

Simultaneously, the cropped image within the **YOLO** bounding box is fed into a **ResNet18**[5] backbone, pre-trained on ImageNet. This network is modified to output a four-dimensional vector representing the object's orientation in space via a **quaternion** representation.

Depth-Aided Refinement Evaluation using the **ADD (Average Distance)** metric revealed significant performance bottlenecks, primarily attributed to the inaccuracies of the diameter-based pinhole estimation. To address this, we extended the architecture by introducing a second **ResNet18** dedicated to depth estimation.

This refinement module takes the original image, the object's 2D bounding box, and the camera intrinsics as input to predict a dedicated depth value Z. This predicted depth is then passed to a standard pinhole model to analytically derive the X and Y coordinates. This architectural shift resulted in a substantial improvement in pose estimation accuracy and ADD scores.

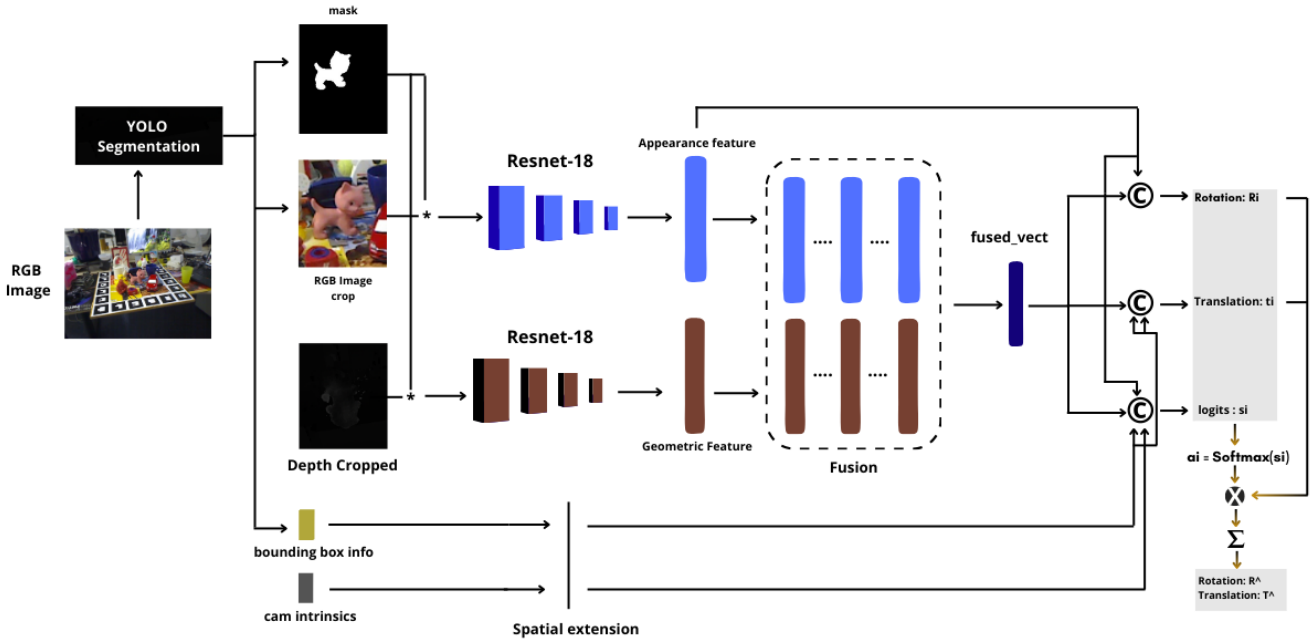


Figure 1. Overview of the proposed Patch-Wise Dense Fusion architecture.

Building on this baseline, we transition to the fully realized architecture proposed in this study (see **Figure 1**).

3.2. Patch-Wise Dense Fusion Model

3.2.1. Feature Extraction

The key technical challenge in this domain is the correct extraction of information from the color and depth channels and their synergistic fusion. Even though color and depth present a similar format in the RGB-D frame, their information resides in different spaces: the former encodes photometric appearance (texture, color), while the latter encodes geometric structure (shape, distance). Therefore, we process them separately to generate color and geometric features from embedding spaces that retain the intrinsic structure of the data sources.

Semantic Segmentation To enable the simultaneous extraction of object bounding boxes and pixel-wise masks, the architecture was transitioned from a YOLO-based object detection framework to a semantic segmentation model.

Dual-Stream backbone We use a **dual-stream backbone architecture** consisting of two parallel **ResNet-18** networks, pre-trained on ImageNet[4].

- **RGB Stream:** The masked RGB crop is fed into the first ResNet stream. The network extracts a high-level appearance feature map $F_{rgb} \in \mathbb{R}^{C \times H' \times W'}$, where $C = 512$ is the channel dimension and $H' = W' = 7$ represents the spatial resolution of the feature grid.
- **Depth Stream:** Similarly, the depth crop is treated as a single-channel intensity map and processed by the second ResNet stream to produce a geometric feature map $F_{depth} \in \mathbb{R}^{C \times H' \times W'}$. The depth crop serves as a geometric intensity map. To ensure compatibility with the standard ResNet architecture (which expects 3-channel input), we replicate the single-channel depth map along the channel dimension to form a pseudo-RGB input ($D_{in} \in \mathbb{R}^{H \times W \times 3}$), producing the geometric feature map.

By removing the final global average pooling and fully connected layers of the original ResNet, we preserve the spatial arrangement of the features. This results in two spatially aligned 7×7 grids where the position (i, j) in F_{rgb} corresponds physically to the same object region as the position (i, j) in F_{depth}.

3.3. Patch-Wise Dense Fusion

Traditional pixel-wise approaches extract features for a set of N randomly sampled 3D points (typically $N \approx 1000$). For each point, the network must perform a computationally expensive "gathering" operation to retrieve the corresponding RGB feature vector from the image plane based on projection coordinates (u, v). This process, involving non-contiguous memory access and unstructured point-cloud processing (e.g., via PointNet), acts as a bottleneck for real-time inference in standard CNN pipelines.

To address this, we propose a **Patch-Wise Dense Fusion** strategy. Instead of operating on unstructured points, our architecture processes the scene as a structured feature grid derived directly from the CNN backbones. As described

	RGB	RGB-D			
Object	Baseline	Global Pool	Standard	NoMask	NoMask-NoAtt
Ape	7.66	85.89	**95.16**	94.76	94.76
Benchvise	41.15	95.88	**98.77**	98.35	98.77
Cam	18.67	**99.59**	97.51	97.93	97.51
Can	53.33	**99.17**	98.75	98.75	98.75
Cat	30.08	99.15	**99.58**	99.15	99.15
Driller	36.13	99.58	98.73	**100.00**	99.58
Duck	9.60	94.42	**96.41**	95.62	90.44
Eggbox	99.60	**100.00**	99.60	99.60	99.60
Glue	90.57	**100.00**	99.59	99.59	99.59
Holepuncher	19.76	98.39	98.39	**98.79**	97.98
Iron	34.20	**98.70**	97.40	97.84	98.70
Lamp	59.35	**100.00**	99.59	99.59	99.59
Phone	33.33	98.78	99.19	**100.00**	97.97
Mean	41.11	97.63	98.36	**98.45**	97.85

Table 1. Ablation Study of the Patch-Wise Dense Fusion Model and comparison with the baseline model. The evaluation metric used is ADD0.1d. The split for this experiments is 80 training vs 20 evaluation

in Sec. 3.2.1, the parallel ResNet streams output two spatially aligned feature maps, F_{rgb} and F_{depth}, of dimension $C \times H' \times W'$ (where $H' = W' = 7$). In this context, a feature vector at grid position (i, j) does not represent a single dimensionless point, but rather a local *image patch* (receptive field). Crucially, thanks to the spatial preservation of the Fully Convolutional Network, the vector $F_{rgb}^{(i,j)}$ encodes the visual appearance of that patch, while $F_{depth}^{(i,j)}$ encodes the geometric structure of the **exact same region**.

We first concatenate the aligned RGB and depth feature maps along the channel dimension. However, simply stacking these features is not enough to fully integrate them. Therefore, we employ a Residual Fusion strategy. The concatenated features are first passed through a convolutional layer to mix the appearance and geometric signals into a unified representation. This mixed feature map is then refined by a residual block (a short sequence of convolutions added back to the input). This process ensures that the final feature grid merges the strengths of both modalities before passing them to the prediction heads.

Dense Prediction Heads. The fused feature grid F_{fused} is fed into three parallel convolutional heads. Unlike DenseFusion which predicts a pose for each pixel, we predict a dense field of poses for each grid cell. The network treats the grid as a set of $M = H' \times W' = 49$ independent estimators: The fused feature grid F_{fused} serves as the common basis for pose estimation. However, recognizing that rotation and translation rely on different visual cues, we employ a **task-specific feature injection** strategy before feeding the data into the parallel heads:

- **Rotation Head:** Accurately estimating orientation relies heavily on high-frequency appearance details (e.g., texture patterns, logos, edges). To preserve this information, we implement a **residual-like skip connection** that reinjects the original RGB features into the rotation stream. The head receives the concatenation of the fused grid that offer semantic informations and the appearance features ($F_{in\_rot} = \text{Concat}(F_{fused}, F_{rgb})$). The head regresses a unit quaternion map $R_{map} \in \mathbb{R}^{M \times 4}$.
- **Translation Head:** Recovering absolute 3D translation requires not just local features, but also global geometric context. In this regard, we inject the camera intrinsic parameters (K) and the object's bounding box spatial information (B). We augment the fused features by concatenating these geometric cues ($F_{in\_trans} = \text{Concat}(F_{fused}, K, B)$), allowing the network to reason about scale and perspective. The head regresses a translation map $T_{map} \in \mathbb{R}^{M \times 3}$.
- **Implicit Attention- Head:**Instead of relying on a simple average of the predictions which would include outliers, occlusions, and background noise inherent in the patch grid, we employ a learned attention-based aggregation mechanism. A dedicated convolutional head predicts a score map $S \in \mathbb{R}^{M \times 1}$ alongside the pose maps. These scores are normalized across the grid using a **Softmax** function to produce a set of attention weights w_i, such

that $\sum w_i = 1$:

$$w_i = \frac{e^{s_i/\tau}}{\sum_{j=1}^{M} e^{s_j/\tau}} \tag{1}$$

where τ is a temperature parameter used to control the sharpness of the distribution. Unlike DenseFusion [12], which enforces an explicit confidence loss, we train this weighting mechanism *implicitly* end-to-end. The final global pose is computed as the weighted sum of the patch predictions: $\hat{P}_{global} = \sum_{i=1}^{M} w_i \cdot \hat{P}_i$. By minimizing the error on the aggregated global pose, the network automatically learns to assign lower weights ($w_i \approx 0$) to unreliable patches (e.g., background or occlusions) and higher attention to discriminatory object features. Empirical results in our experiments demonstrate that this **Attention Aggregation** strategy yields a quite good improvement ($\approx 1.5\%$) compared to uniform averaging or global pooling baselines, confirming its effectiveness as a learned soft-masking mechanism but leaving space for improvement on this idea.

4. Experiments

The primary objective of our experimental evaluation is to investigate the efficacy of a 2D-driven RGB-D approach that avoids explicit 3D reconstruction. Specifically, we aim to quantify the performance of our model, which leverages depth maps and segmentation masks directly, relative to state-of-the-art (SOTA) architectures that typically rely on point cloud representations for geometric reasoning.

4.1. Dataset: LineMOD

We conduct our training and evaluation on the LineMOD dataset [12], a standard benchmark for 6D object pose estimation. LineMOD consists of 13 categories of low-texture objects situated in cluttered indoor environments. It is characterized by challenging lighting conditions and significant inter-object occlusion, making it a rigorous testbed for monocular and depth-assisted pose estimation. The experimental evaluation is conducted using two distinct partitioning strategies of the LineMOD dataset to ensure both internal validity and external comparability:

1. **80/20 Split:** A standard partition wherein 80% of the data is allocated for training, with the remaining 20% reserved for validation and performance evaluation.
2. **15/85 Split:** A restricted-training partition adopted to maintain consistency with established benchmarks in existing literature, facilitating a direct comparison with state-of-the-art methods.

4.2. 6D Pose Estimation

Having defined the two networks pipelines, let's focus on their learning objective.

Baseline Loss Function The baseline model is trained end-to-end using a multi-task loss function that balances translation accuracy and rotational alignment. Let $\mathcal{L}_{trans}$ be the Smooth L1 loss (or Mean Squared Error) between the predicted and ground truth translation vectors, and $\mathcal{L}_{rot}$ be the quaternion loss defined as:

$$\mathcal{L}_{rot} = 1 - |\langle \hat{\mathbf{q}}_{norm}, \mathbf{q}_{norm} \rangle| \tag{2}$$

where $\mathbf{q}_{norm} = \frac{\mathbf{q}}{\|\mathbf{q}\|_2}$ denotes the L_2 normalization onto the unit hypersphere. This formulation effectively computes the sine of the angle between the two quaternions in the four-dimensional space, providing a smooth gradient for backpropagation. The multi-task loss is defined as:

$$\mathcal{L}_{total} = \lambda_{trans}\mathcal{L}_{trans} + \lambda_{rot}\mathcal{L}_{rot} \tag{3}$$

Where we empirically set $\lambda_{trans} = 0.2$ and $\lambda_{rot} = 1.0$ to prioritize rotational convergence during the optimization process.

From Component-wise Loss to Point-Matching Loss Inspired by DenseFusion[12], we define the learning objective as the distance between points sampled on the object's 3D model in the ground truth pose and those same points transformed by the predicted pose. Specifically, the loss to minimize for the prediction per patch i is defined as:

$$L_i^p = \frac{1}{M} \sum_j \|(Rx_j + t) - (\hat{R}_i x_j + \hat{t}_i)\| \tag{4}$$

where x_j denotes the j^{th} point of M randomly selected 3D points from the object's 3D model, $p = [R|t]$ is the ground truth pose, and $\hat{p}_i = [\hat{R}_i|\hat{t}_i]$ is the predicted pose generated from the fused embedding of the i^{th} patch. This loss function is well-defined for asymmetric objects where the shape and texture determine a unique canonical frame.

4.3. Evaluation Metrics

In alignment with established protocols in the literature [12], we employ the **Average Distance of Model Points (ADD)** metric for asymmetric objects:

$$ADD = \frac{1}{m} \sum_{x \in M} \left\| (Rx + t) - (\widehat{R}x + \widehat{t}) \right\|_2 \tag{5}$$

Where x are the points randomly selected from the 3D model M, $\widehat{R}$ and $\widehat{t}$ is the predicted pose. For symmetric objects (*eggbox* and *glue*) we utilize the **Average Close Point Distance** (ADD-S) metric:

$$ADD - S = \frac{1}{m} \sum_{x_1 \in M} \min_{x_2 \in M} \left\| (Rx_1 + t) - (\widehat{R}x_2 + \widehat{t}) \right\|_2 \tag{6}$$

Following standard practice, a pose estimate is considered a "success" if the error is below the 10% threshold of the object's 3D diameter.

4.4. Ablation study

Having established the general architecture, we evaluate the synergy of the individual modules to the overall model performance. Specifically, we conduct a comparative ablation study using an $80/20$ split of the LineMOD dataset, benchmarking three distinct variants of the **Patch-Wise Dense Fusion** (PWDF) framework:

1. **PWDF-Standard:** The baseline architecture integrated with both the segmentation mask and the attention estimation head.
2. **PWDF-GlobalPooling:** A configuration utilizing the segmentation mask but substituting the Attention head with a Global Average Pooling (GAP) layer to test effectivness of attention head.
3. **PWDF-NoMask:** A variant maintaining the attention head while omitting the mask input to quantify its importance
4. **PWDF-NoMask-NoAttHead:** Last variant to verify how the model perform without mask and attention head.

4.4.1. Attention Head Effectiveness

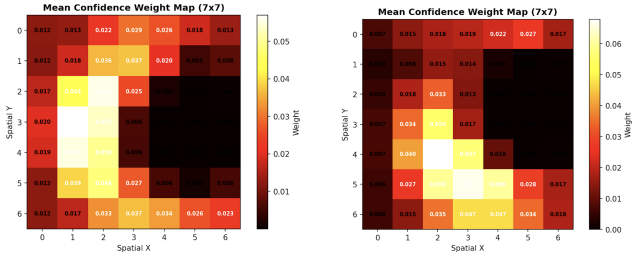
The ablation results in **Table 1** reveal a consistent pattern: configurations employing the attention head (PWDF-Standard and PWDF-NoMask) achieve the highest accuracy scores of 98.36% and 98.45% respectively, outperforming the Global Average Pooling approach (97.63%) by approximately 0.8-1%. This improvement, while modest in absolute terms, is significant given the already high baseline performance and demonstrates the value of learned spatial weighting over naive averaging.

Attention Distribution Analysis. A critical question is whether the attention head learns meaningful spatial attention or collapses to a trivial uniform distribution—effectively reducing to average pooling. Notably, our architecture does not explicitly regularize the attention weights through the loss function, as proposed in the original DenseFusion work [12]. Despite this, our analysis reveals that the network naturally learns non-uniform attention patterns.

Figure 2 presents the attention weight analysis for our best-performing model (PWDF-NoMask). We quantify the attention distribution using normalized entropy.

Spatial Attention Patterns. The mean attention weight maps (**Figure 2b**) reveal consistent spatial preferences across samples. Interestingly, the attention does not concentrate uniformly at the image center as might be expected. Instead, we observe higher weights in specific regions (left side and bottom portions of the feature map), suggesting the network has learned to identify discriminative features for pose estimation that may correspond to object edges, texture boundaries, or geometric discontinuities rather than object centers.

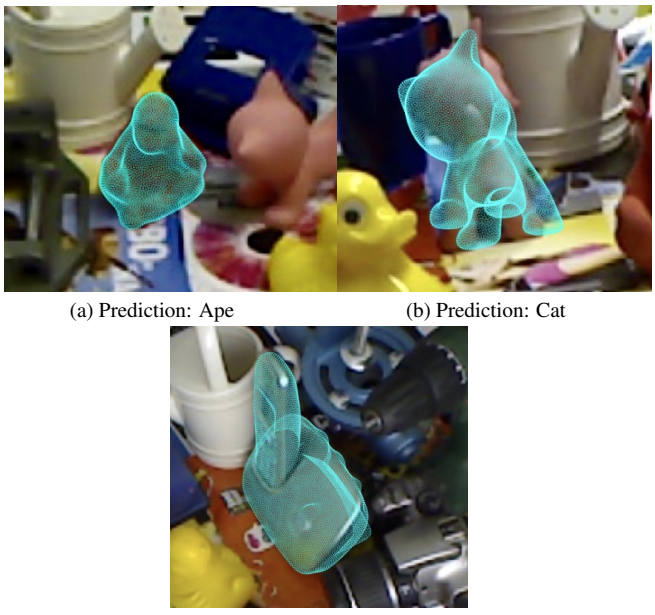
Limitations and possible Improvements. While the attention head demonstrates meaningful behavior, there is room for improvement. The normalized entropy of 0.86-0.89 suggests the attention could be more selective. Incorporating such regularization could potentially sharpen the attention distribution and further improve pose estimation accuracy.



(a) Attention head analysis for PWDF-Mask

(b) Attention head analysis for PWDF-NoMask

Figure 2. Distribution of attention weights showing right-skewed pattern with many suppressed regions.



(a) Prediction: Ape

(b) Prediction: Cat

(c) Prediction: Phone

Figure 3. Visual representation of the point clouds produced by PWDF

Object	DeepIm [7]	DenseFusion (per-pixel)[12]	DenseFusion (iterative)[12]	Ours Standard
Ape	77.0	79.5	**92.3**	73.0
Benchvise	**97.5**	84.2	93.2	87.2
Cam	93.5	76.5	**99.4**	91.2
Can	96.5	86.6	93.1	**97.4**
Cat	82.1	88.8	**96.5**	95.8
Driller	95.0	77.7	87.0	**98.4**
Duck	77.7	76.3	**92.3**	79.1
Eggbox	97.1	**99.9**	99.8	**99.9**
Glue	99.4	99.4	**100.0**	**100.0**
Holepuncher	52.8	79.0	**92.1**	90.5
Iron	98.3	92.1	97.0	**98.4**
Lamp	97.5	92.3	95.3	**98.3**
Phone	**97.7**	88.0	92.8	94.4
Mean	88.6	86.2	**94.3**	92.6

Table 2. 6D pose estimation accuracy on the LineMOD benchmark. Performance is evaluated on the restricted 15/85 training-test partition to ensure comparability with state-of-the-art methods.

4.5. Results on Standard Benchmark

4.5.1. Impact of Data Splitting Strategies

While our ablation study on the 80/20 split demonstrated the potential of the No-Mask architecture, we acknowledge that this partition strategy may yield optimistic performance estimates. Since the LineMOD dataset consists of continuous video sequences, a random 80% training split introduces high temporal correlation between training and validation samples (i.e., adjacent video frames often appear in both sets). Consequently, the network may learn to recognize specific background features or lighting conditions rather than generalizing to the object's geometry.

To address this and ensure a fair comparison with state-of-the-art methods like DenseFusion and DeepIM, we evaluated our best-performing models (PWDF-Standard and PWDF-NoMask) on the standardized **15/85 split**. Notably, unlike many competing approaches, we perform this evaluation **without using any additional synthetic data**, relying solely on the approximately 200-250 real training images per object.

4.5.2. The Role of Segmentation in Data-Scarce Regimes

The transition to the restricted 15/85 split revealed a critical divergence in performance between the masked and unmasked architectures.

- **PWDF-NoMask Failure:** The No-Mask approach, which excelled in the data-rich 80/20 split, failed to converge effectively on the 15/85 split, exhibiting severe overfitting. We hypothesize that this degradation is due to **spurious background correlations**. In the absence of a large dataset (or synthetic augmentation), a model without explicit masking tends to exploit contextual shortcuts—such as the texture of the supporting table or the relative position of neighboring objects—rather than learning the intrinsic appearance of the target object.
- **PWDF-Standard Robustness:** Conversely, the PWDF-Standard architecture (utilizing segmentation masks) proved robust to data scarcity. By filtering out background noise, the mask acts as a strong **inductive bias**, forcing the network to focus exclusively on the object's visual and geometric features. Although this could cause inaccuracy in spatial localization, the integration of camera parameters and bounding box metadata via our feature injection module provides sufficient geometric constraints to maintain high-fidelity predictions.

4.6. Comparison with State-of-the-Art

As shown in **Table 2**, the PWDF-Standard model achieves a mean ADD accuracy of **92.6%**. Crucially, our method outperforms the direct baseline **DenseFusion (per-pixel)** (86.2%) by a significant margin (+6.4%). While our accuracy is slightly lower than **DenseFusion (iterative)** (94.3%), it is important to highlight that our result is achieved:

- **Without iterative refinement** (which increases inference latency).
- **Without processing unstructured point clouds** (saving computational resources).
- **Without synthetic data training** (demonstrating high data efficiency).

In terms of computational efficiency, we evaluated the inference speed of our full pipeline on an NVIDIA GeForce RTX 4060 Laptop GPU. With a batch size of 1, the total inference time is just 25.4 ms per image (YOLO: 21.6

ms; PWDF: 3.7 ms). Increasing the batch size to 32 further improves efficiency, reducing the per-image latency to 20.4 ms (YOLO: 18.9 ms; PWDF: 1.6 ms). For comparison, DenseFusion[12] requires approximately 60 ms, making our approach 2x to 3x faster. Moreover , the complete pipeline comprises 35.73M parameters (32.32M for the pose estimation network and 3.40M for YOLOv8n-seg), requiring 142.92 MB of memory in FP32 precision or 71.46 MB in FP16. This confirms that the Patch-Wise Dense Fusion approach provides an excellent trade-off between accuracy and system complexity, offering a competitive alternative for embedded applications where computational resources are limited.

5. Conclusion

In this work, we presented Patch-Wise Dense Fusion (PWDF), a lightweight architecture designed to address the computational overhead of unstructured point cloud processing in 6D pose estimation. By operating directly on spatially aligned RGB-D feature embeddings, our method avoids expensive 3D reconstruction while preserving dense geometric structure. Experimental results demonstrate the dual advantages of our approach: the implicit "No-Mask" variant achieves 98.45% accuracy on the $80/20$ split, offering superior efficiency by eliminating segmentation dependency. Conversely, in data-scarce regimes ($15/85$ split), our masked variant exhibits high robustness, attaining 92.6% mean ADD accuracy and outperforming the DenseFusion per-pixel baseline by 6.4% without requiring iterative refinement or synthetic data. These findings establish PWDF as a compelling, high-efficiency alternative suitable for real-time embedded robotic applications.

References

[1] P.J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992. 2

[2] Eric Brachmann, Frank Michel, Alexander Krull, Michael Ying Yang, Stefan Gumhold, and Carsten Rother. Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3364–3372, 2016. 1

[3] R. Qi Charles, Hao Su, Mo Kaichun, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 77–85, 2017. 1, 2

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. IEEE, 2009. 3

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 2

[6] Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab. Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1530–1538, 2017. 2

[7] Yi Li, Gu Wang, · Xiangyang, · Xiang, and · Fox. Deepim: Deep iterative matching for 6d pose estimation. *International Journal of Computer Vision*, 128, 2020. 2, 7

[8] Debapriya Maji, Soyeb Nagori, Manu Mathew, and Deepak Poddar. Yolo-6d-pose: Enhancing yolo for single-stage monocular multi-object 6d pose estimation. In *2024 International Conference on 3D Vision (3DV)*, pages 1616–1625, 2024. 2

[9] Nuno Pereira and Luís A. Alexandre. Maskedfusion: Mask-based 6d object pose detection. *CoRR*, abs/1911.07771, 2019. 2

[10] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016. 1, 2

[11] Rejin Varghese and Sambath M. Yolov8: A novel object detection algorithm with enhanced performance and robustness. In *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*, pages 1–6, 2024. 1, 2

[12] Chen Wang, Danfei Xu, Yuke Zhu, Roberto Martin-Martin, Cewu Lu, Li Fei-Fei, and Silvio Savarese. Densefusion: 6d object pose estimation by iterative dense fusion. pages 3338–3347, 2019. 1, 2, 5, 6, 7, 8

[13] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *CoRR*, abs/1711.00199, 2017. 1, 2

[14] Danfei Xu, Dragomir Anguelov, and Ashesh Jain. Pointfusion: Deep sensor fusion for 3d bounding box estimation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 244–253, 2018. 2

[15] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast global registration. In *Computer Vision – ECCV 2016*, pages 766–782, Cham, 2016. Springer International Publishing. 1