DOPE-Plus: Enhancements in Feature Extraction and Data Generation for 6D Pose Estimation

Jeffrey Chen
Rackham Graduate School (of UMich)
Department of Robotics
Ann Arbor, United States
jeffzc@umich.edu

Yuqiao Luo
Rackham Graduate School (of UMich)
Department of ECE
Ann Arbor, United States
joeluo@umich.edu

Longzhen Yuan
Rackham Graduate School (of UMich)
Department of ECE
Ann Arbor, United States
longzhen@umich.edu

***Abstract*—This study explored enhancements to the DOPE framework by improving both its network architecture and synthetic data generation pipeline for 6D object pose estimation. We proposed replacing the original VGG-19-based feature extractor with a Vision Transformer (ViT), aiming to leverage its superior representation capabilities. In parallel, we developed a refined data generation pipeline, resulting in an augmented HOPE dataset [1] and a new fully synthetic dataset of a customized object, Block. These datasets were used to train and evaluate our modified DOPE model on two target objects: Cookies and Block. Experimental results demonstrate that incorporating ViT improves pose estimation performance over the original VGG-19 backbone, suggesting the potential for further advancements through the integration of more powerful feature extractors. This project's public repository is available at: https://github.com/jypipi/DOPE-Plus.**

I. Introduction

As robotics continues to advance, researchers are increasingly exploring ways to equip robots with the capabilities needed to perform everyday tasks. Many of these tasks require fundamental operations such as object fetching, which depend on accurate pose estimation of target objects. This study investigated the DOPE (Deep Object Pose Estimation) proposed by J. Tremblay et al. in 2018 [2], and further extended the feature extraction and data generation pipelines. The original DOPE framework employed VGG-19 as the feature extractor. In our work, we replaced it with a Vision Transformer (ViT) [3], [4], motivated by its superior feature extraction capabilities, particularly in capturing relationships between multiple objects. Meanwhile, we enhanced the data synthesis pipeline proposed by [2] to augment and generate two new datasets for network training. Our goal is to improve the accuracy of 6D object pose estimation and to validate the effectiveness of our enhancements for object perception in real-world scenarios.

A. Original DOPE

DOPE (Deep Object Pose Estimation) is a one-shot, instance-based, deep neural network-based system designed to estimate the 3D poses of known objects in cluttered scenes from a single RGB image, in near real time and without the need for post-alignment. The system employs a straightforward deep network architecture trained entirely on synthetic data. It predicts the 2D image coordinates of the projected 3D bounding box corners, which are then used with the perspective-n-point (PnP) algorithm to recover the 3D pose.

1) Network architecture: The DOPE network is a convolutional deep neural network that detects objects' 3D keypoints using multi-stage architechture. Firstly, image features are extracted by the first ten layers of the VGG-19 convolutional neural network (with pre-trained parameters). Then two $3 * 3$ convolutional are applied to the features to reduce the feature dimensions from 512 to 128. Second, these 128-dimensional features are fed into the first stage, which consists of three $3*3*128$ convolutional layers and one $1*1*512$ layer, followed by a $1*1*9$ to produce belief maps and and $1*1*16$ to produce vector fields. There are 9 believe maps, 8 of them are for the projected vertices of the 3D objects and one for its centroid. Vector fields indicate that the direction from vertices to their corresponding centroids, to construct the bounding boxes of objects after detection. There are overall 6 identical stages as the first stage, except for the follow-up stages accept image features, belief maps and vector fields as their input, therefore they have five $7 * 7 * 128$ and one $1 * 1 * 128$ layers to align data before converting to belief maps and vector fields. The network could leverages increasing larger receptive fields as data go through the neural network. This enables the network reduce the ambiduities in early stages and thus produce context relationships in later stages.

2) Data Generation: As more data is required to train a deep network with high performance, it can be difficult to gather enough data for training. In addition, unlike 2D labeling, making 3D pose labels manually is much more difficult. DOPE proposed a method to generate data, which allows scientists to gather enough number of data rapidly, and greatly alleviate the workload of labeling manually.

The overall data synthesis strategy is to generate two kinds of dataset: "domain randomized (DR)" and "photorealistic (photo)". The domain randomized data are generated by putting the target object into a virtual environment, which is composed of different distractor objects and a random background. The objects shown in DR images do not necessarily obey physical principles. Photorealistic data are generated by putting target objects into 3D backgrounds with physical constraints. In other words, they are impacted by the effects of gravity and collision.

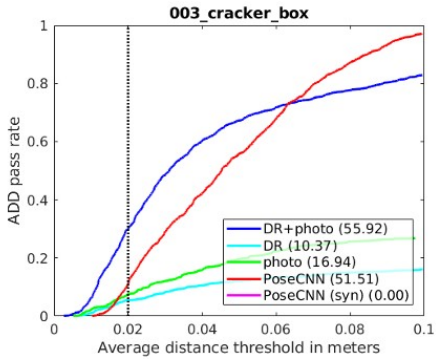


Fig. 1: Performance Comparison between DOPE and poseCNN on Cracker Box

Training on these synthesized data demonstrated a good experiment result. Figure 1 (reproduced from [2]) compares its performance with traditional poseCNN. It also shows that using a combination of DR and photo images gives a much better result than using them alone.

II. Related Work

Many efforts has been made to recognize object pose accurately. One of the most impactful work is poseCNN [5]. It can estimate the 6D pose from a single RGB image, and presents a decent result. Wang Y. et al. proposed DenseFusion [6], which further introduced depth information to enhance the model performance. In DOPE, the 3D model of the target object must be known beforehand to enable pose estimation, which helps improve precision. This approach is particularly suitable for applications such as fetching known objects in controlled environments. In general, various sensing modalities have been employed across different methods to maximize the robustness and accuracy of 6D object pose estimation.

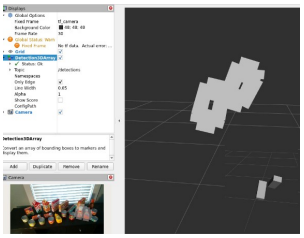
III. Reproduction & Algorithmic Extension

A. Original Paper Reproduction

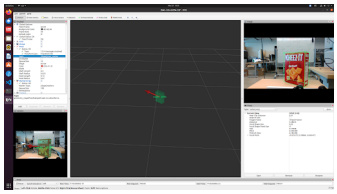
We reviewed the original DOPE paper and codebase to thoroughly understand the proposed network architecture, dataset compositions, and training techniques. To reproduce the original work, we configured our local computing environment with the necessary dependencies for the original framework. We successfully ran DOPE with the Robotics Operation System (ROS), establishing real-time data streaming and model inference in ROS Noetic. As shown in Figure 2, we deployed the pose estimation pipeline using pre-trained models and visualized inference results for two objects, Tomato Ketchup and Cracker Box, with both open-sourced datasets and Intel RealSense RGB-D camera streams, which were consistent with those reported in [2].

B. Network Architecture

One of the key algorithmic extensions we introduced involves replacing the original VGG-19 feature extractor with a Vision Transformer (ViT). This modification was driven



(a) Tomato Ketchup



(b) Cracker Box

Fig. 2: Pose Estimation Reproduction with RViz Visualization

```
    # === 1. ViT backbone using timm ===
    self.vit = timm.create_model(
        'vit_base_patch16_224', pretrained=True
            , features_only=True
    )

    vit_out_channels = self.vit.feature_info
        [-1]['num_chs']

    self.feature_proj = nn.Sequential(
        nn.Conv2d(vit_out_channels, 256,
            kernel_size=1),
        nn.ReLU(inplace=True),
        nn.Conv2d(256, 128, kernel_size=1),
        nn.ReLU(inplace=True),
    )
```

Listing 1. ViT Backbone and Feature Projection

by ViT's superior capability to capture long-range dependencies and global contextual information, thanks to its self-attention mechanism. Unlike VGG-19, which relies on local convolutional filters with relatively small receptive fields, ViT processes the entire image as a sequence of patches, enabling it to model holistic scene relationships more effectively. To accommodate this change, substantial adjustments were made throughout the model backbone, including feature dimensionality alignment and downstream compatibility with the transformer-based output structure. Basically, we changed one of the core codebases *model.py*, and some key changes are shown in Listing 1.

We created a pre-trained ViT feature extractor using the timm library. It accepts images of dimension 244 × 244 with a patch size 16 × 16, as a result, interpolating is needed to make sure the input data has a size of 244 × 244. Then we take the output from ViT only in the final layer. At the next stage, two convolutional layers are employed to reduce the number of channel to 128, hence the dimension matches the following network structure (the belief map stages). Corresponding changes also need to be made in forward function and in detector.py. Because now the weight parameters has been changed to adopt ViT structure, so in the inference stage code also need to be modified in order to adopt this new parameter file. Figure 3 demonstrated this structure in a clear flow chart.

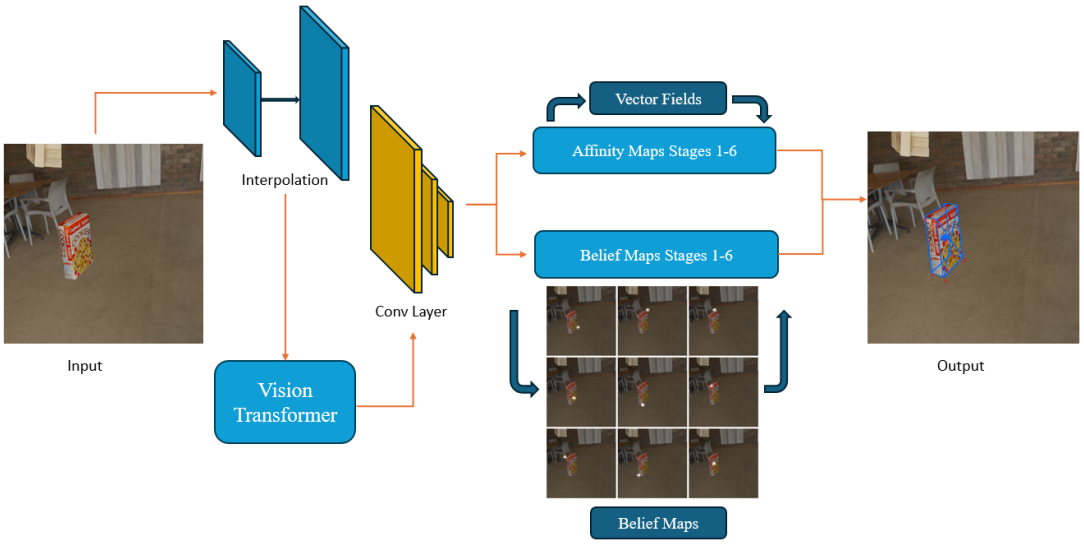


Fig. 3: Enhanced ViT-DOPE Network Architecture

C. Data Generation

We enhanced the original data generation pipeline [2] using BlenderProc to produce two distinct synthetic RGB datasets, each corresponding to a specific target object: Cookies and Block. The Cookies object is part of the publicly available HOPE dataset [1], while the Block is a newly introduced, custom-designed object. Our pipeline incorporates randomized camera poses, object poses, and 360-degree HDRI backgrounds, while ensuring that these variations remain physically reasonable. These improvements aim to create a more diverse and robust synthetic dataset, helping to mitigate the common sim-to-real domain gap in deep learning applications. The enhanced pipeline consists of four main stages: (1) textured 3D CAD modeling, (2) real-world HDRI background generation, (3) image synthesis, and (4) ground truth annotation pre-processing.

1) Textured 3D CAD Modeling and Real-World Background Generation: To obtain a precise 3D textured model of the customized object, we first used SolidWorks to create an accurate geometric model with correct dimensions. Blender was then employed to add textures and enrich visual details, including colors and physical material properties, as shown in Figure 6(b).

For real-world HDRI background generation, we captured raw 360-degree images of the desired physical environments using the Insta360 X3 camera. These images were subsequently pre-processed and converted into HDRI backgrounds using Adobe Photoshop, as illustrated in Figure 6(a).

2) Image Synthesis: With all necessary elements prepared, we proceeded to the image synthesis stage. We developed a Python script to randomize the poses of cameras, target objects, and distractors. To emulate typical indoor scenarios encountered in onboard SLAM and manipulation tasks, we assumed that both the camera and the target object remained upright, with randomized yaw angles and small perturbations in pitch and roll. In contrast, distractor objects were randomized with full degrees of freedom as a form of data augmentation, without adhering to physical stability constraints.

3) Ground Truth Annotation Pre-Processing: With the existing pipeline provided by [2], ground truth annotations for each frame were automatically generated. However, when constructing a comprehensive dataset for training and validation, it was necessary to combine synthetic and real images from various sources. In this case, the annotation files (e.g., JSON files) often differed in format and configuration. To streamline data preparation and ensure compatibility with downstream tasks, we developed an additional Python script to pre-process and standardize the ground truth annotations.

D. Innovative Enhanced Datasets

We augmented the original HOPE dataset and created a new dataset for the customized Block object by generating synthetic domain-randomized (DR) images, referred to as HOPE-Syn&Real and the Synthetic Block Dataset, respectively.

1) HOPE Data Augmentation (HOPE-Syn&Real Dataset): We generated additional synthetic data based on the HOPE dataset [1]. The original dataset consists of 28 grocery items, with approximately 300 real images per object. We selected Cookies as the target object for subsequent training tasks. To enrich the existing dataset, we synthesized additional 12,000 domain-randomized (DR) images of this object using the enhanced data generation pipeline developed upon [2], and combined them with the existing real images to form the HOPE-Syn&Real dataset. To verify the quality of the synthesized images, we employed a validation method adapted from the original codebase to visualize the ground truth annotations, as shown in Figure 5.

2) Synthetic Block Dataset: In addition to augmenting the HOPE dataset, we created a fully synthetic dataset for our customized Block object using the aforementioned methods

and strategies. This dataset consists of over 19,300 domain-randomized images, with random variations in block poses, instance counts, backgrounds, and distractor objects. Furthermore, as shown in Figure 6(c), lighting conditions and shadows were simulated and rendered to further enhance realism and dataset diversity.

(a) Raw RGB Image (b) Ground Truth Annotations

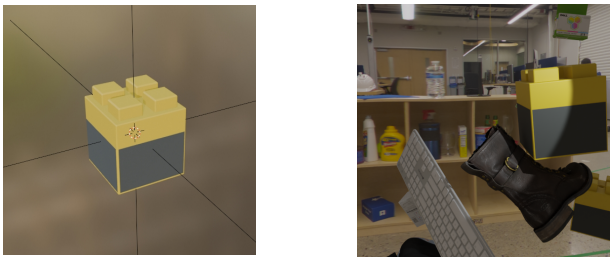
Fig. 4: Sampled HOPE Dataset Real Image and Ground Truths

Fig. 5: Sampled Generated Data and Visualized Ground Truth in the HOPE-Syn&Real Dataset. (Left column: generated RGB images, Right column: visualized ground truths)

IV. Experiments and Results

Since DOPE is an instance-based network that requires prior knowledge of the target object, each trained model is limited to recognizing a single object. In our experiments, we selected Cookies from the HOPE dataset and Block (a custom object we introduced) as the two target objects. For each object, we trained both the original DOPE model and our modified ViT-DOPE variant. The models were then evaluated based on their final pose estimation results with both visualizations and quantitative metrics, including mean Average Precision (mAP), Average Distance of Model Points (ADD), and predicted keypoints accuracy.

(a) Sampled HDRI Background



(b) 3D Textured Model (c) Sampled Synthetic Image

Fig. 6: 3D Textured Block Model (Blender), Sampled HDRI Background, and Synthetic Domain Randomized Image in the Synthetic Block Dataset

A. Experimental Setup

To facilitate training on the HOPE dataset, we implemented a pre-processing step to convert each object's pose, which was originally represented as a transformation matrix, into a ground truth belief map. In this dataset, object poses are provided as 4×4 transformation matrices that encode the object's 6D pose relative to the camera, along with the corresponding camera intrinsics. To generate the belief maps, we first reconstructed the 3D bounding boxes of the objects based on their real-world dimensions. These 3D coordinates were then projected onto the 2D image plane using the camera's intrinsic matrix (as illustrated in Figure 4(b)). After computing the 2D centroids from the projected bounding boxes, we rendered the ground truth belief maps and integrated them into the training pipeline as supervision signals for our model.

B. Training

To quantify and compare model performance, we trained a total of four models: one original DOPE and one ViT-DOPE model for each object (Cookies and Block). The HOPE-Syn&Real Dataset and the Synthetic Block Dataset were used to train the Cookies and Block models, respectively. Each dataset was split into training and validation subsets, with the validation sets comprising approximately 5%–7% of the total images. Due to project timeline constraints and the absence of open-source photorealistic data generation scripts in the original DOPE codebase, neither dataset contained photorealistic images. As a result, the Cookies models were trained on both domain-randomized (DR) and real images, whereas the Block models were trained exclusively on DR data.

To ensure a reasonable comparison, all models were trained using identical hyperparameters, optimizers, and learning rate schedules, as detailed in Table I. Specifically, the AdamW optimizer and a Cosine Annealing learning rate schedule were employed, with an initial learning rate of 0.00005. The only difference in the training procedure was the number of epochs: the Cookies models were trained for 200 epochs, whereas the Block models were trained for 400 epochs. The evolution of the learning rate over the course of training is illustrated in Figure 7.

Hyperparameter	Value
Batch Size	64
Initial LR	0.00005
Weight Decay	0.001
Epoch (Cookies)	200
Epoch (Block)	400

TABLE I: Key Hyperparameters and Values

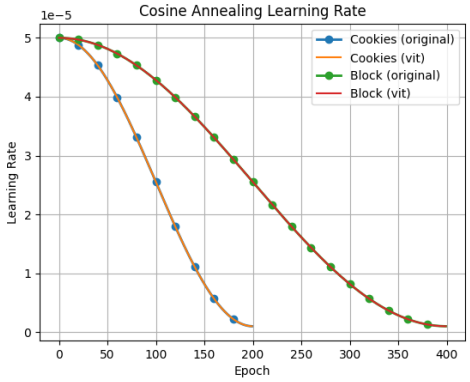
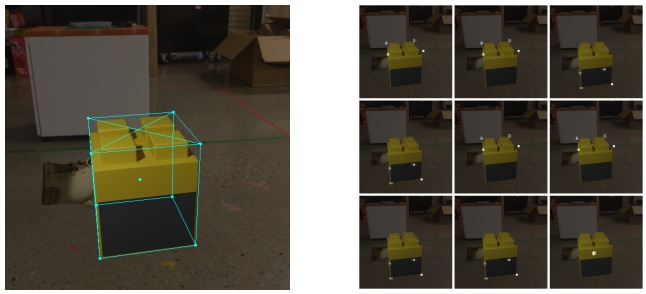


Fig. 7: Learning Rates vs. Epochs

C. Results and Analysis



(a) Ground Truth (b) Predicted Belief Maps

Fig. 8: Ground Truth and Key Point Beliefs of a Block object in a Sampled Frame

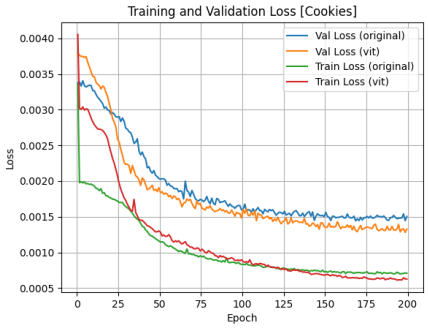
The losses and performance metrics of all models converged to stable values with minor fluctuations after training. We define predicted keypoints accuracy as the percentage of keypoints correctly predicted within a specified pixel threshold across a batch of images.

1) Cookies: Figure 9 quantifies the performance of the Cookies models on both the training and validation subsets of HOPE-Syn&Real. As shown in Figure 9(a), all losses dropped rapidly during the initial phase of training and gradually stabilized around epoch 125, indicating a reasonable and expected loss convergence trend. In Figure 9(b), the validation accuracy is consistently higher than the training accuracy. This phenomenon suggests either noise in the training set or the presence of strong regularization during training. The former is unlikely, as the images were randomly split into training and validation sets. The latter appears more plausible, as a weight decay rate of 0.001 and additional data augmentation techniques (such as blur and contrast adjustments) were applied during training to mitigate overfitting. Such regularization can lead to higher validation accuracy relative to training accuracy.

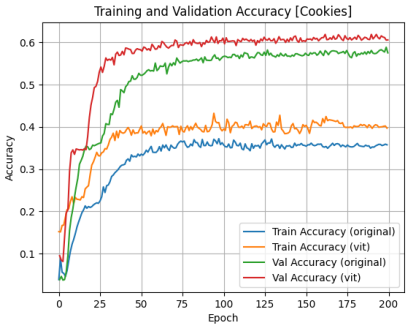
When comparing the original DOPE network (using a VGG-19 feature extractor) with our ViT-DOPE architecture, both the training and validation accuracies of ViT-DOPE are approximately 5% higher. This improvement highlights the effectiveness of our architectural enhancements and aligns with the primary goals of this project. Nevertheless, as demonstrated in Figures 9(b) and (d), despite the improvements, the overall estimation performance of both the original DOPE and ViT-DOPE models remains relatively low. Several factors may contribute to this: the batch size was halved due to GPU resource constraints; the dataset lacked photorealistic images; and the total number of synthetic images was limited compared to the original DOPE training process. Specifically, the original DOPE model was trained with approximately 60,000 domain-randomized and 60,000 photorealistic images per object [2], whereas our HOPE-Syn&Real Dataset comprises only about 10.25% of that amount.

Additionally, the absolute values of predicted keypoint accuracies and ADD metrics may have been affected by the relatively strict pixel threshold used for evaluation, where predictions were considered correct only if they fell within approximately 10 pixels of the ground truth. Despite these limitations, the observed trends demonstrate a promising future for DOPE-based architectures. Our results suggest that with stronger feature extractors and further dataset enhancements, significant improvements in 6D pose estimation accuracy are achievable.

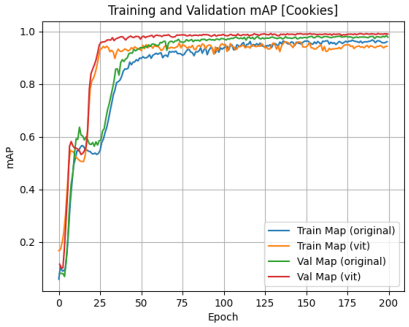
2) Block: Figure 10 showcases the performance of the Block models on the training and validation splits of the Synthetic Block Dataset. As demonstrated, although the losses remained low after training, the predicted keypoint accuracies plateaued around 20%, and the mAP values stabilized below 0.4, indicating a failure of both the DOPE and ViT-DOPE models for this object. While the absence of photorealistic data may have contributed, the primary cause is attributed to DOPE's core architectural limitations in handling object symmetry [2]. As illustrated in Figure 8, the trained models successfully predicted the nine keypoints of the Block object but failed to generate accurate pose estimations due to the object's geometric and textured symmetry. This also explains the low and oscillating trends observed in the accuracy metrics



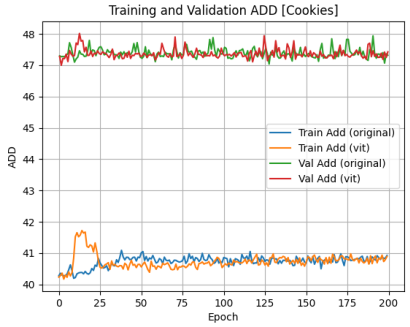
(a) Training and Validation Loss



(b) Predicted Keypoints Accuracy

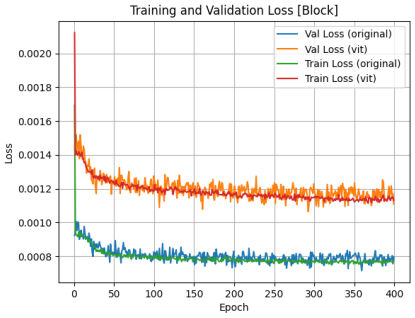


(c) mAP

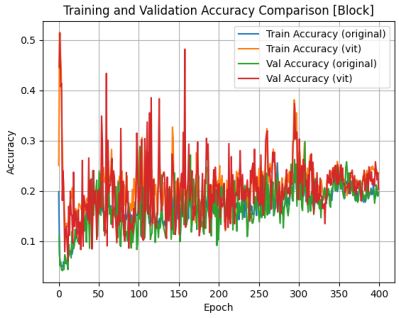


(d) ADD

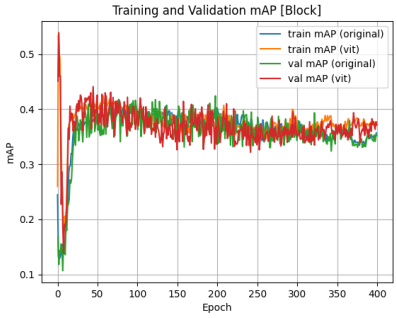
Fig. 9: Cookies Models Performance (Training and Validation)



(a) Training and Validation Loss



(b) Predicted Keypoints Accuracy



(c) mAP

Fig. 10: Block Models Performance (Training and Validation)

in Figure 10.

However, the models' success in locating keypoints of the desired object demonstrated by the belief maps shown in Figure 8(b) highlights the strength of the feature extractors and encourages potential future algorithmic extensions to better handle symmetry.

V. Conclusions

Our study successfully extends the original DOPE framework through enhancements in both feature extraction and data generation. By replacing VGG-19 with a vision transformer and increasing the randomization and realism of the datasets, we achieved improvements in object pose estimation performance. These results demonstrate the potential of ViT for precise feature extraction and highlight the effectiveness of synthetic data in facilitating model training, enabling high model accuracy while significantly reducing the need for manual ground truth annotation. For future research, integrating other advanced feature extractors into the DOPE architecture is a promising direction, as further improvements in estimation performance are likely achievable with more powerful backbones. Additionally, developing algorithmic extensions to better accommodate object symmetry is another important avenue for increasing the adaptability and robustness of the DOPE network.

REFERENCES

[1] Y. Lin, J. Tremblay, S. Tyree, P. A. Vela, and S. Birchfield, "Multi-view fusion for multi-level robotic scene understanding," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 6817–6824.

[2] J. Tremblay, T. To, Y. Xiang, D. Fox, and S. Birchfield, "Deep object pose estimation for semantic robotic grasping of household objects," *arXiv preprint arXiv:1809.10790*, 2018.

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

[4] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[5] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," in *Proceedings of Robotics: Science and Systems (RSS)*, Pittsburgh, Pennsylvania, USA, 2018.

[6] Y. Wang, Y. Xiang, D. Xu, T. Zhang, T. Khurana, Y. Xiao, A. Deng, and D. Fox, "Densefusion: 6d object pose estimation by iterative dense fusion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3343–3352.