PoET: Pose Estimation Transformer for Single-View, Multi-Object 6D Pose Estimation – Supplementary Material –

Thomas Jantos
Control of Networked Systems Group
University of Klagenfurt, Austria
thomas.jantos@aau.at

Mohamed Amin Hamdad
Infineon Technologies Austria AG
Villach, Austria
mohamedamin.hamdad@infineon.com

Wolfgang Granig
Infineon Technologies Austria AG
Villach, Austria
wolfgang.granig@infineon.com

Stephan Weiss
Control of Networked Systems Group
University of Klagenfurt, Austria
stephan.weiss@aau.at

Jan Steinbrener
Control of Networked Systems Group
University of Klagenfurt, Austria
jan.steinbrener@aau.at

1 Introduction

In this supplementary material, we present additional results for our PoET framework that allow other works to perform detailed comparisons in the future. While the paper discusses the core results and main findings for our approach, we conduct and present the results of an extensive ablation study, which investigates the influence of the network architecture, the rotation representation, data augmentation, transformer modifications and the backbone on PoET's performance. Moreover, we provide results for additional metrics and visualize qualitative results for a better understanding. Besides that, we provide additional details regarding our implementation and the dataset preparation. Finally, we evaluate PoET on the benchmark dataset Linemod-Occluded (LM-O) [1] and investigate different quaternion losses.

2 Implementation Details & Data Preparation

As discussed in the main paper, PoET can be trained on top of any object detector. For the YCB-V dataset [2] a Scaled-YOLOv4 [3] is used as the backbone object detector as it offers a good trade-off between speed and accuracy. An MS-COCO [4] pre-trained Scaled-YOLOv4 is fine tuned for 10 epochs on the YCB-V dataset for the object detection task. For the LM-O dataset we train PoET on top of a publicly available, pre-trained Mask R-CNN [5][1] network. During the training of PoET, the weights of the object detector backbone are frozen.

In order to utilize the benefits of batch processing, the number of input object queries has to be constant across all images. This is usually not the case, as the number of objects present in an image can vary significantly. Therefore, the number of object queries n_q is fixed for a specific dataset to the maximum number of objects present in any of its images. If fewer objects are present in an image, the remainder of the object queries are filled up with dummies. Such a dummy object is assigned a bounding box $(c_x, c_y, w, h) = (-1, -1, -1, -1)$, a dummy class and a dummy query embedding.

[1] https://github.com/ylabbe/cosypose

6th Conference on Robot Learning (CoRL 2022), Auckland, New Zealand.

In case that the object detector predicts more objects than the number of allowed object queries, the top-n_q predictions are chosen based on the classification score. Due to the transformer's attention mechanism, it should learn not to focus on dummy embedding feature vectors. For the loss calculation and the evaluation, dummy object queries are disregarded, as object queries are matched to ground-truth objects based on bounding box center distance, predicted class and generalized intersection over union (GIoU) using an adjusted Hungarian matcher similar to [6]. Dummy embeddings are not needed in inference, as only single images are processed. During training, the ground-truth bounding box and class are used instead of the object detector predictions.

We implement PoET using PyTorch [7] and train it on a single NVIDIA GeForce RTX 3090 for 50 epochs using AdamW [8] with a learning rate of $2e-5$ and a batch size of 16. Our best performing network has five encoder and decoder layers, $d_h = 256$, 16 attention heads and a positional embedding dimension of $L = 32$. The network is simultaneously trained for 3D translation and 3D rotation estimation and the weighting parameters are set to $\lambda_t = 2$ and $\lambda_{rot} = 1$, such that both losses are in the same value range after scaling. During evaluation, our whole pipeline consumes around 5.3GB of VRAM and runs with 71 FPS.

YCB-V [2] contains 21 objects [9] and consists of 92 video sequences totalling 133,827 frames (`real`). Each video sequence contains multiple objects and the level of object occlusion and scene clutter varies between the sequences. Throughout the whole dataset at most 9 objects are contained in one scene and thus we fix n_q to 10. Following the proposed and commonly used data split [2, 10, 11, 12, 13, 14], we train our network on 80 of those 92 video sequences and reserve the remaining 12 sequences for testing. Out of these 12 sequences, the challenging keyframes are used for evaluation. The original authors of the dataset provide 80,000 synthetic images generated from 2D projections of the 3D object models (`synt`). We include these synthetic images during training for better network generalization. The background of the synthetic images is blank and thus, to improve network generalization even further, we randomly choose an image from the MS COCO dataset [4] as a background. Even though the BOP challenge also provides photorealistic images (`pbr`) for YCB-V, we do not utilize this data at all.

LM-O [1] is a single video sequence of the LM dataset [15] containing 1214 frames and the annotated ground-truth pose for 8 objects. In this specific scene, there is significant occlusion between the objects. Following the BOP challenge [10], we only use the 50,000 publicly available synthetic images, which were generated using physically-based rendering (`pbr`), for training PoET. Given that per image only a single instance of each object is present, the number of object queries n_q is set to 10. For evaluation we use the ADD(-S) metric as described by Hinterstoisser et al. [15].

We also perform random RGB augmentation during training by not only modifying the image color, sharpness, brightness and contrast, but also blurring the image and converting it to grayscale. However, images are not scaled, flipped or cropped.

3 Main Ablation Study on YCB-V

In this ablation study, we want to investigate the influence of different components on the performance of our PoET framework. By assuming a perfect object detector that provides ground-truth bounding boxes to PoET, we ensure that the evaluation includes every object present in the image even for those which might be not detected by the object detector. During the ablation study, we focus on the AUC of ADD/ADD-S metric, the average translation error and rotation error. All results reported in this section are for the same hyperparameter configuration as described in the main paper and the same fixed seed. The final results are summarized in Table 1. Detailed results for each class along with an in-depth analysis are provided in Section 4.

Network Architecture. We utilize our best performing model from the main paper as our baseline model (`Baseline`). First of all, we compare it to its class-agnostic counterpart (`Agnostic`). We observe that having dedicated outputs in the rotation and translation head for each class improves the results across all four metrics. Especially the average translation error is improved. We also

Table 1: Ablation study results of PoET on YCB-V. We report the AUC of ADD/-S and the average translation and rotation error. Ablation of class mode (`Agnostic`), network size (`Small`), rotation representation (`(Silho)Quat`), data augmentation (`w/o Aug.`) and transformer modifications (`RP, Q, RP + Q`). The exact meaning of the tags are described in the text.

Metric	Baseline	Agnostic	Small	Jitter	Quat	SilhoQuat	w/o Aug.	RP	Q	RP + Q
AUC of ADD-S	**92.8**	88.9	91.8	90.7	91.8	88.8	85.6	87.6	82.2	42.0
AUC of ADD	**80.8**	73.2	78.1	73.5	77.3	71.6	63.2	66.8	59.5	12.2
Avg. T. Error [cm]	**1.20**	1.95	1.48	1.58	1.25	2.11	2.36	1.92	2.99	9.06
Avg. Rot. Error [°]	**23.65**	24.92	25.64	30.35	28.63	27.54	37.95	37.31	35.39	74.26

investigate the influence of a smaller transformer network (`Small`) by training a PoET that only has 3 encoder and decoder layers; everything else is kept the same. While the smaller network results in an expected loss of performance across all metrics, it is still very competitive and leads to faster processing times (88 FPS). Therefore, by accepting minimal drops in performance a more efficient PoET can be trained in terms of memory consumption and computational time, which is beneficial for systems with hardware limitations.

Rotation representation. One of the design choices made was to use a 6D rotation representation. However, quaternions are still a popular rotation representation due to their ability to express a rotation with just 4 values and also they do not suffer from gimbal lock such as the Euler angle representation. Therefore, we want to show that PoET can be trained using quaternions (`Quat`). The output dimension of the rotation head is adjusted to $4 \cdot n_{cls}$, the output is L_2 normalized to ensure unit vector requirement for quaternions and the loss function is replaced by

$$L_{rot} = -\log(< q, \tilde{q} >^2 + \epsilon) \,, \tag{1}$$

where $< \cdot, \cdot >$ represents the regular vector dot product and ϵ is a small number for numerical stability. The 6D representation achieves only slightly better results than the quaternion representation. The main reason is that `Quat` has on average a bigger rotation error. Important to note, the choice of loss function influences the performance of PoET in case of the quaternions. We train PoET also using the same quaternion loss function as Billings and Johnson-Roberson [14] (`SilhoQuat`). We can observe that while the average rotation error is slightly better, the average translation error is almost twice as big. The main reason for a decreased performance with respect to the translation estimation might be due to the different loss landscapes of the two different quaternion rotation loss functions. A more detailed analysis is provided in Section 9.

Data Augmentation. In previous work [16, 12] it was mentioned that data augmentation in the RGB space is beneficial for the pose estimation task. We investigated the influence of our data augmentations as described in Section 2 on PoET's performance (`w/o Aug`). The results clearly indicate that including simple RGB data augmentation has a significant impact on PoET's ability to learn the task of 6D pose estimation.

Transformer Modifications. Finally, we investigated the influence of replacing learnable reference points and query embeddings by directly passing bounding box information to the transformer decoder as described in our main paper's methods section. We either make the reference points (`RP`), the query embeddings (`Q`) or both (`RP + Q`) learnable. In the latter case, the transformer architecture does not perform at all in comparison to the baseline network given that they were trained for the same number of epochs. In general, directly feeding bounding box information as reference points and object query embeddings does not only yield an improvement in terms of performance, but it also saves computational complexity as it means less trainable components.

Backbone. Besides the ablation study regarding PoET's network architecture, we also want to investigate how the quality of the predictions of the object detector influences the performance. We measure quality of bounding boxes in terms of IoU with respect to their corresponding ground-truth. In order to model the decreasing object detector quality, we add jitter to the ground-truth bounding boxes before evaluating on them by sampling from a truncated normal distribution. Given a ground-truth bounding box, we sample a new center such that it is contained within the original bounding box. Furthermore, we sample new widths and heights from another truncated normal distribution

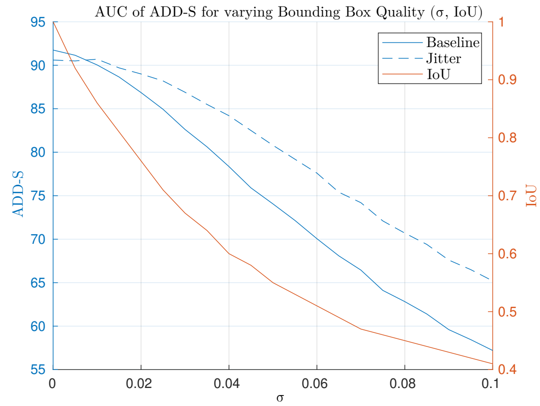


Figure 1: Comparison between `Baseline` and PoET trained with jitter bounding boxes for different bounding box qualities. With varying σ the IoU of the ground-truth boxes decreases and so the performance of PoET.

limiting them to be between 0.7 and 1.3 of the original size. The degree of noise is determined by the standard deviation σ. For both truncated normal distributions we use the same σ and calculate the AUC of ADD-S and the average IoU across all bounding boxes by varying σ between 0 and 0.1. While $\sigma = 0$ corresponds to no additional noise, a $\sigma = 0.1$ results in 99% of the sampled values being within a range of 150 pixels with respect to their original value. The results of this analysis can be found in Fig. 1. With decreasing object detector quality, the performance of PoET decreases. While the peak performance of `Baseline` cannot be matched by `Jitter`, we can see that the latter is not as influenced by the decreasing IoU of the object detector. Even though PoET's performance depends on the quality of the object detector, we would still achieve state-of-the-art AUC of ADD-S when using non-perfect object detectors. For reference, FCOS [17], Faster R-CNN [18] and Scaled-YOLOv4 [3] evaluated on the keyframes achieve an average IoU across all detections of 0.86, 0.86 and 0.92 respectively. The main reasons for the discrepancy between the expected AUC of ADD-S score of PoET and the true one achieved by PoET working on Scaled-YOLOv4 is due to the fact that an object detector does not always detect all the objects present. However, the achieved score is close to the expected score. Moreover, we also compare our baseline to a PoET trained with jittered bounding boxes and $\sigma = 0.02$ (`Jitter`), but evaluated with ground-truth bounding boxes to make the comparison of the ablation networks independent of the quality of the object detector.

4 In-depth Analysis of Ablation Study

This section serves as an extension to our ablation study on the YCB-V dataset [2]. We present detailed results for each class for the AUC of ADD-S, AUC of ADD, average translation and average rotation error in Table 2, Table 3, Table 4 and Table 5 respectively.

The difference in performance in terms of the AUC of ADD-S metric for our different ablation networks mostly stems from their performance with respect to the translation estimation. When comparing the achieved ADD-S scores, reported in Table 2 of the ablation networks to their average translation and rotation errors, as reported in Table 4 and Table 5 respectively, one can see that the better the translation estimation is, the better the ADD-S score is, while the rotation seems to have a slightly smaller influence, e.g. for `Agnostic` and `Small`. On the other hand, a better estimation of the object rotation improves the performance on the AUC of ADD metric. Nevertheless, for the AUC of ADD metric a good translation estimation is required as can be seen by comparing `Quat` to `SilhoQuat`. We refer the reader to Section 9 for an in-depth comparison between `Quat` and `SilhoQuat`.

The class-agnostic version of PoET (`Agnostic`) mostly achieves two to three points less than the `Baseline` model for the AUC of ADD-S metric. However, the performance drops especially for the wood block and the foam brick, two objects that show symmetries in the RGB as well as silhouette space. While `Agnostic` can better estimate the rotation of these two objects, the translation estimation suffers heavily due to not learning class-specific outputs.

As already mentioned in Section 3, `Small` achieves slightly lower scores across all metrics than our `Baseline`. To no surprise, this is due to the smaller network architecture offering less parameters to better capture objects' characteristics. The performance drop is especially noticeable for the banana. There are many objects in the dataset that share similar shapes and thus can be easier learned. However, the banana has a unique shape and the smaller network seems to be not able to capture important characteristics of the banana.

Applying simple RGB data augmentation during the training improves the performance of PoET drastically. For the translation we can observe an improvement across all classes. While most classes also benefit from RGB augmentation in terms of rotation error (especially objects symmetric in RGB and silhouette space), we have observed the opposite effect for the master chef and the tomato soup can. These two objects are symmetric in silhouette space, but contain enough features in the RGB space to precisely determine the orientation of the object. Therefore, it is apparent that RGB augmentation benefits the network to focus more on geometric features of objects rather than RGB features.

In the main ablation study it was already discussed that replacing learnable reference points and object query embeddings by bounding box information yields better results for PoET. Given the results from Table 2, it is clear that using positionally-encoded bounding box information as the query embedding contains the most information. Namely, making this part of the transformer learnable (`Q`) performs significantly worse than our `Baseline`. A version of PoET trained with learnable reference points (`RP`) does also not achieve the same scores as the `Baseline` but it still performs better than `Q`. This shows that the positional bounding box encoding contains a lot of information, as `RP` can still predict sufficient reference points from the query embeddings and thus achieve higher scores. However, there are differences between `RP` and `Q` when comparing on the class level. While `RP` performs better for objects that have either (borderline) rotational symmetries in the silhouette space or multiple symmetry axes and planes in the RGB as well as silhouette space, having learnable query embeddings (`Q`) benefits PoET for objects that have either unique shapes (scissors, pitcher base and bananas) or that have features in RGB space that break the symmetries present in the silhouette space (gelatin and pudding box). Nonetheless, there are still outliers to this observation, e.g., the cracker box. Learning both reference points and query embeddings (`RP + Q`) significantly degrades performance across all classes. Further investigation is needed as to why this is the case.

These results illustrate that the design decisions for the baseline model indeed lead to a significant improvement in performance.

Table 2: Ablation study results of PoET on YCB-V. We report the AUC of ADD-S scores for each class and average score over all classes. Ablation of class mode, network size, rotation representation, data augmentation and transformer modifications.

Object	Baseline	Agnostic	Small	Jitter	Quat	SilhoQuat	w/o Aug	RP	Q	RP + Q
master chef can	**92.9**	87.6	92.6	91.4	92.4	88.6	82.4	91.0	71.5	58.1
cracker box	**90.4**	88.0	89.4	85.3	86.3	85.5	80.9	84.8	79.1	46.6
sugar box	**94.5**	91.2	92.4	93.6	93.3	89.4	92.7	88.0	89.5	49.5
tomato soup can	**94.0**	89.9	92.8	92.8	93.4	88.3	92.4	88.5	85.3	27.3
mustard bottle	**94.8**	87.4	90.0	89.1	91.6	82.7	88.3	88.2	81.7	59.1
tuna fish can	94.0	90.9	93.1	93.3	**95.3**	92.0	84.4	91.4	87.5	51.3
pudding box	93.8	93.6	93.0	92.0	**94.1**	89.2	88.0	84.4	89.9	37.2
gelatin box	92.7	90.6	92.3	**93.4**	91.0	92.9	84.6	86.9	91.7	40.0
potted meat can	**94.1**	90.8	92.9	91.8	93.9	89.7	91.6	90.2	83.9	59.3
banana	**94.3**	91.0	89.4	88.0	86.0	91.5	86.9	80.5	87.0	52.8
pitcher base	**94.3**	92.6	92.8	91.5	92.9	89.8	92.0	83.9	89.5	70.8
bleach cleanser	**92.6**	90.1	89.3	91.6	90.5	87.9	84.5	87.7	85.6	58.8
bowl*	92.1	87.6	**94.2**	91.1	93.2	92.4	93.5	89.9	80.5	17.8
mug	**94.1**	89.5	92.2	93.9	93.8	92.0	86.4	93.6	82.0	53.7
power drill	**94.3**	91.9	91.9	91.6	92.0	88.1	90.2	88.5	85.9	45.7
wood block*	92.0	81.4	**92.9**	86.4	90.5	85.2	75.6	91.2	63.4	31.5
scissors	92.5	90.8	91.8	88.3	**92.6**	90.5	83.8	68.7	80.6	25.9
large marker	81.6	84.5	84.7	83.1	85.3	82.0	60.8	**85.6**	77.3	18.5
large clamp*	95.7	93.0	95.6	94.7	**96.0**	92.6	88.7	93.6	80.5	20.3
extra large clamp*	**96.0**	88.5	94.8	94.8	94.5	88.6	91.4	91.3	83.6	45.3
foam brick*	89.7	75.2	88.9	87.2	90.3	85.3	79.3	**91.0**	69.3	11.6
All	**92.8**	88.9	91.8	90.7	91.8	88.8	85.6	87.6	82.2	42.0

Table 3: Ablation study results of PoET on YCB-V. We report the AUC of ADD scores for each class and average score over all classes. Ablation of class mode, network size, rotation representation, data augmentation and transformer modifications.

Object	Baseline	Agnostic	Small	Jitter	Quat	SilhoQuat	w/o Aug.	RP	Q	RP + Q
master chef can	40.6	28.3	34.9	37.9	37.6	**45.5**	40.3	30.6	21.5	16.8
cracker box	**79.6**	72.8	78.7	67.6	67.4	70.0	55.7	64.0	41.2	3.9
sugar box	**89.6**	83.4	85.0	87.1	86.7	79.3	85.3	74.0	78.4	12.0
tomato soup can	72.5	66.1	67.9	70.1	69.6	64.1	**73.5**	62.6	53.0	9.7
mustard bottle	**82.2**	69.4	76.4	59.1	71.8	50.6	53.7	64.0	63.9	12.2
tuna fish can	66.2	64.3	63.8	62.8	**68.8**	63.4	40.5	65.2	51.9	16.1
pudding box	88.4	88.2	86.8	84.9	**88.5**	79.1	77.4	69.3	80.5	13.8
gelatin box	87.0	83.4	85.4	**87.9**	82.7	86.8	68.4	73.9	84.4	11.3
potted meat can	**88.3**	81.1	85.4	83.5	87.3	79.9	81.0	79.4	66.6	29.1
banana	**86.3**	81.6	70.5	45.0	72.4	81.2	52.1	41.8	42.7	8.1
pitcher base	**87.0**	83.4	82.7	80.1	82.3	77.8	80.5	38.8	75.6	23.8
bleach cleanser	**82.8**	81.0	77.3	76.6	74.4	68.1	58.8	73.2	69.8	26.3
bowl*	**76.9**	62.4	82.3	66.8	68.7	59.4	60.9	71.9	45.0	1.3
mug	**86.9**	77.4	82.6	86.1	85.3	82.1	69.3	85.4	62.5	16.1
power drill	**88.0**	82.7	83.7	81.6	83.4	74.2	77.3	71.2	71.1	13.5
wood block*	83.8	64.2	**86.1**	73.1	78.6	70.3	50.6	81.4	35.7	2.9
scissors	**85.9**	82.3	83.7	75.8	85.5	81.3	63.0	33.5	66.2	15.4
large marker	73.3	76.1	76.3	73.0	**76.9**	72.8	48.4	76.5	66.8	7.7
large clamp*	90.0	83.7	89.4	87.3	**90.8**	82.5	74.6	85.3	64.7	2.1
extra large clamp*	**90.4**	71.7	86.7	87.7	86.4	69.5	72.2	80.5	64.3	14.4
foam brick*	72.0	53.7	75.4	70.0	78.6	66.6	44.9	**79.3**	44.1	0.3
All	**80.8**	73.2	78.1	73.5	77.3	71.6	63.3	66.8	59.5	12.2

Table 4: Ablation study results of PoET on YCB-V. We report the average translation error in cm for each class and average score over all classes. Ablation of class mode, network size, rotation representation, data augmentation and transformer modifications.

Object	Baseline	Agnostic	Small	Jitter	Quat	SilhoQuat	w/o Aug.	RP	Q	RP + Q
master chef can	**1.37**	2.43	1.38	1.68	1.49	2.29	3.47	1.75	5.47	7.60
cracker box	**1.48**	2.05	1.78	2.50	2.14	2.54	3.12	2.25	3.00	9.28
sugar box	**0.94**	1.57	1.42	1.16	1.04	2.03	1.33	2.22	1.99	9.38
tomato soup can	**1.09**	1.87	1.30	1.35	**1.09**	2.25	1.39	2.04	2.70	11.70
mustard bottle	**0.94**	2.28	1.77	1.93	1.02	3.46	1.96	1.94	3.23	6.64
tuna fish can	0.95	1.80	1.32	1.30	**0.79**	1.63	3.12	1.58	2.51	7.83
pudding box	1.01	0.99	1.19	1.45	**0.84**	2.00	2.02	2.49	1.62	8.79
gelatin box	1.20	1.59	1.42	**1.14**	1.44	1.28	2.67	2.14	1.40	8.90
potted meat can	1.13	1.82	1.40	1.59	**1.06**	1.98	1.54	1.83	3.07	7.20
banana	**1.06**	1.76	2.02	1.60	1.49	1.77	2.30	2.51	1.62	6.63
pitcher base	**0.95**	1.32	1.10	1.49	1.07	1.88	1.39	2.92	1.71	5.82
bleach cleanser	**1.09**	1.68	1.75	1.39	1.44	2.22	2.63	1.87	2.50	6.40
bowl*	1.51	2.37	**1.05**	1.38	1.17	1.46	1.14	1.71	2.85	14.54
mug	**1.28**	2.24	1.69	1.33	1.31	1.77	2.92	1.40	3.73	8.47
power drill	**0.98**	1.57	1.52	1.48	1.02	2.05	1.54	1.54	2.60	7.53
wood block*	1.41	3.54	**1.29**	2.52	1.53	2.81	4.57	1.38	6.69	13.10
scissors	1.38	1.73	1.53	2.16	**1.14**	1.93	2.16	3.67	3.23	12.31
large marker	2.68	2.30	2.15	2.57	2.12	2.63	4.64	**2.11**	2.94	11.16
large clamp*	0.98	1.59	1.01	1.22	**0.75**	1.71	2.42	1.37	3.34	12.40
extra large clamp*	**0.91**	2.12	1.24	1.19	1.13	2.20	1.83	1.73	3.15	8.34
foam brick*	1.90	4.64	2.31	2.58	1.79	2.80	3.42	**1.76**	5.29	14.12
All	**1.20**	1.95	1.48	1.58	1.25	2.11	2.36	1.92	2.99	9.06

Table 5: Ablation study results of PoET on YCB-V. We report the average rotation error in degrees for each class and average score over all classes. Ablation of class mode, network size, rotation representation, data augmentation and transformer modifications.

Object	Baseline	Agnostic	Small	Jitter	Quat	SilhoQuat	w/o Aug.	RP	Q	RP + Q
master chef can	89.25	107.22	106.61	92.34	92.68	72.89	**58.75**	103.36	70.19	82.52
cracker box	9.68	13.76	**8.47**	14.82	20.46	11.40	27.55	25.62	52.10	94.01
sugar box	**3.95**	5.12	4.29	5.15	7.57	3.97	5.77	11.32	8.80	32.04
tomato soup can	50.97	54.62	59.57	56.96	56.63	51.62	**43.58**	59.95	76.68	81.93
mustard bottle	23.71	31.69	21.20	67.50	45.24	76.25	96.55	51.75	**20.42**	125.40
tuna fish can	60.30	**48.55**	61.78	64.54	52.37	57.98	84.49	53.85	73.29	75.59
pudding box	6.36	7.02	7.43	**4.92**	10.96	8.08	11.35	27.65	15.61	51.83
gelatin box	6.69	7.92	4.84	6.09	13.85	**3.80**	18.01	26.92	10.27	90.58
potted meat can	**5.06**	6.80	6.20	6.27	9.30	6.67	14.00	11.89	14.77	41.17
banana	7.90	5.12	26.63	81.88	26.71	**5.62**	60.20	78.62	82.19	119.98
pitcher base	**7.51**	8.98	11.71	12.04	11.60	10.08	12.46	71.09	14.26	67.58
bleach cleanser	16.32	**8.12**	15.61	26.35	27.92	33.37	50.96	20.61	17.86	62.75
bowl*	16.06	29.26	**12.23**	34.62	26.86	48.01	45.97	21.18	62.19	82.84
mug	3.86	7.29	5.88	5.58	8.46	**3.76**	17.51	6.39	10.32	51.55
power drill	5.92	5.90	**4.90**	10.47	12.75	12.20	15.90	29.50	11.40	81.50
wood block*	5.88	4.85	**3.94**	6.90	15.77	6.45	13.76	9.06	20.51	63.41
scissors	**3.19**	4.20	7.43	8.53	12.73	5.72	68.54	146.30	10.33	113.19
large marker	24.95	**19.85**	27.00	27.82	23.43	24.64	49.56	26.12	28.68	90.32
large clamp*	**2.61**	4.00	3.24	3.33	7.26	2.67	8.42	6.36	11.10	85.03
extra large clamp*	**2.38**	16.15	3.71	2.58	6.89	26.88	28.00	8.60	13.67	70.12
foam brick*	37.20	**11.39**	12.89	27.38	19.32	25.14	102.18	18.79	28.20	82.56
All	**23.65**	24.92	25.64	30.35	28.63	27.54	37.95	37.31	35.39	74.26

5 Backbone Comparison on YCB-V

The results of PoET reported in the main paper for the YCB-V dataset [2] were achieved with a Scaled-YOLOv4 [3] as the object detector. Moreover, the ablation study discusses the influence of the object detector quality on the performance of PoET. However, only the quality influences the performance of PoET and not the actual object detector architecture. In Table 6 we compare PoET trained on top of a Scaled-YOLOv4 (`YOLO`) to PoET trained on top of a pre-trained Mask R-CNN (`R-CNN`). We evaluate `YOLO` and `R-CNN` with ground-truth bounding boxes as well as the object detectors' actual predictions. In either case, the actual feature maps of the respective object detector is utilized. Moreover, we provide results of other state-of-the-art RGB-based methods for better comparison.

While the AUC of ADD-S score varies slightly for different classes, PoET achieves the same average score independent of the actual object detector architecture given ground-truth bounding boxes. Using the actual predictions confirms our ablation study's findings that the quality of the object detector influences PoET's performance. As reported in the main paper, our `YOLO` object detector performs better on YCB-V than a Mask R-CNN [5]. This is reflected in the small difference of the average AUC of ADD-S between `YOLO` and `R-CNN`. These results are a clear indication that PoET can be used in combination with different pretrained object detectors to achieve state-of-the-art results. Additionally, we provide the AUC of ADD scores in Table 7 and compare them to PoseCNN [2], the only other RGB-based method reporting results for this metric.

Taken together, these results illustrate that PoET can be used with different object detector backbones achieving competitive performance even for older object detector architectures.

Table 6: **AUC of ADD-S**. Comparison of PoET's performance trained on top of different object detectors for the YCB-V dataset [2]. We report results for PoET trained on top of a Scaled-YOLOv4 [3] (`YOLO`) and Mask R-CNN [5] (`R-CNN`) evaluated with ground-truth (gt) and predicted bounding boxes. In either case, the actual feature maps of the respective object detector is utilized. Additionally, we compare the results to other RGB-only state-of-the-art methods. The 3D model row indicates how 3D model information is incorporated into the network. `2D` indicates that only 2D image information is used.

Object	PoseCNN[2]	SilhoNet[14]	SilhoNet$_{gt}$	MCN[13]	MCN$_{gt}$	YOLO$_{gt}$	YOLO	R-CNN$_{gt}$	R-CNN
3D Model	Loss	Input + Sym	Input + Sym	2D	2D	2D	2D	2D	2D
master chef can	84.0	84.0	83.6	87.8	91.2	**92.9**	88.4	92.8	85.2
cracker box	76.9	73.5	88.4	64.3	78.5	90.4	80.5	**92.0**	87.0
sugar box	84.3	86.6	88.8	82.4	85.1	94.5	92.4	**95.1**	92.0
tomato soup can	80.9	88.7	89.4	87.9	93.3	94.0	91.4	**95.6**	91.7
mustard bottle	90.2	89.8	91.0	92.5	91.9	**94.8**	91.7	92.5	91.0
tuna fish can	87.9	89.5	89.9	84.7	95.2	94.0	90.4	**95.3**	92.3
pudding box	79.0	60.1	89.1	51.0	84.9	**93.8**	89.0	88.6	81.1
gelatin box	87.1	92.7	**94.6**	86.4	92.1	92.7	91.7	93.7	89.3
potted meat can	78.5	78.8	84.8	83.1	90.8	**94.1**	91.2	93.8	86.9
banana	85.9	80.7	88.7	79.1	70.0	**94.3**	89.5	84.7	80.3
pitcher base	76.8	91.7	91.8	84.8	91.1	94.3	91.7	**94.9**	93.1
bleach cleanser	71.9	73.6	72.0	76.0	86.8	92.6	85.4	**93.8**	88.0
bowl*	69.7	79.6	72.5	76.1	85.0	92.1	90.5	**92.9**	88.6
mug	78.0	86.8	92.1	91.4	91.9	94.1	91.4	**94.2**	88.5
power drill	72.8	56.5	82.9	76.0	87.2	94.3	88.8	**94.9**	91.6
wood block*	65.8	66.2	79.2	54.0	87.2	**92.0**	75.7	91.6	74.4
scissors	56.2	49.1	78.3	71.6	80.2	**92.5**	75.2	89.8	63.7
large marker	71.4	75.0	83.1	60.1	66.4	81.6	81.2	**90.8**	86.5
large clamp*	49.9	69.2	84.5	66.8	86.5	95.7	88.6	**96.2**	87.9
extra large clamp*	47.0	72.3	88.4	61.1	79.5	**96.0**	83.5	94.5	88.8
foam brick*	87.8	77.9	88.4	60.9	79.2	**89.7**	81.3	88.5	79.7
All	75.9	79.6	85.8	75.1	86.9	**92.8**	87.1	92.7	86.1

Table 7: **AUC of ADD**. Comparison of PoET's performance trained on top of different object detectors for the YCB-V dataset [2]. We report results for PoET trained on top of a Scaled-YOLOv4 [3] (`YOLO`) and Mask R-CNN [5] (`R-CNN`) evaluated with ground-truth and predicted bounding boxes.

Object	PoseCNN [2]	YOLO_{gt}	YOLO	R-CNN_{gt}	R-CNN
master chef can	**50.9**	40.6	42.2	38.6	37.0
cracker box	51.7	79.6	57.9	**83.2**	70.8
sugar box	68.6	89.6	85.0	**91.0**	84.5
tomato soup can	66.0	72.5	69.5	**72.9**	68.4
mustard bottle	79.9	**82.2**	76.9	72.3	67.1
tuna fish can	**70.4**	66.2	59.7	70.0	65.4
pudding box	62.9	**88.4**	79.3	80.1	68.2
gelatin box	75.2	87.0	84.9	**88.2**	79.6
potted meat can	59.6	**88.3**	81.8	87.4	76.1
banana	72.3	**86.3**	74.1	57.8	53.7
pitcher base	52.2	87.0	82.1	**88.7**	84.7
bleach cleanser	50.5	82.8	68.6	**88.4**	76.0
bowl	6.5	**76.9**	68.7	69.1	59.7
mug	57.7	86.9	81.1	**87.1**	74.8
power drill	55.1	88.0	75.5	**89.6**	82.1
wood block	31.8	**83.8**	49.3	82.6	52.2
scissors	35.8	**85.9**	62.5	80.6	47.7
large marker	58.0	73.3	72.6	**83.8**	77.7
large clamp	25.0	90.0	75.9	**91.4**	75.2
extra large clamp	15.8	**90.4**	65.1	86.5	75.0
foam brick	40.4	72.0	60.0	**75.1**	61.9
All	53.7	**80.8**	70.1	79.3	68.5

6 Qualitative Results

6.1 6D Pose Estimation

Figure 2: Qualitative results for selected frames of the YCB-V dataset. Given the relative 6D object poses predicted by PoET we project the 3D model into the image.

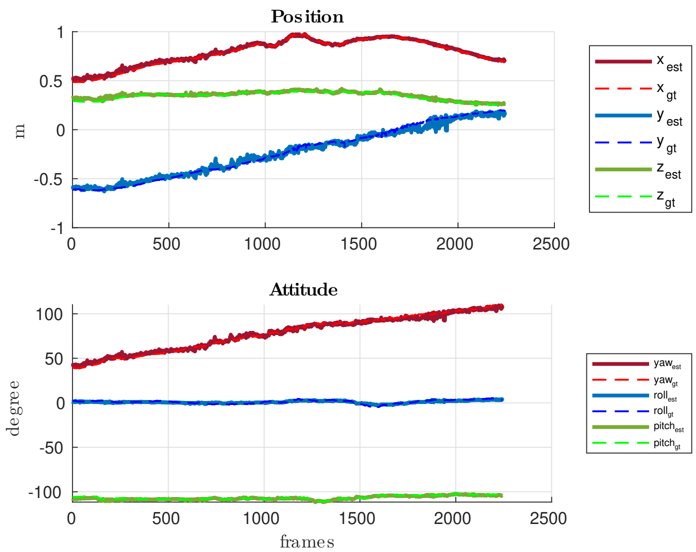
In Fig. 2 we present PoET's qualitative results for the 6D pose estimation task on the YCB-V dataset. The results range from almost perfect estimation in the top left image to failure cases in the bottom right image. The qualitative results show that PoET is able to handle occlusion due to its feature maps containing global information.

6.2 Localization Trajectories

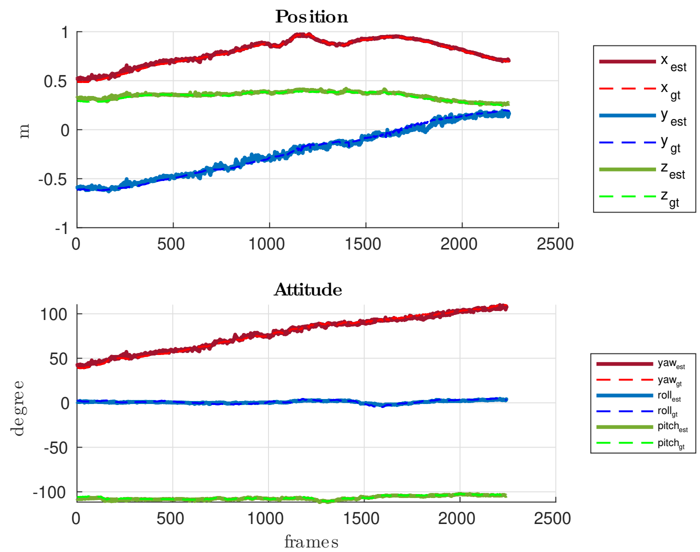
In this subsection we provide further examples for the localization task discussed in the main paper. We introduced three different approaches to the problem of localizing a camera given landmark world positions and using PoET's relative object pose estimates. Furthermore, we investigated the best possible estimate by choosing the hypothesis that is closest to the current ground-truth camera pose. In Fig. 3 we compare the estimated position and attitude for the two outlier rejection approaches to the best trajectory approach across one trajectory. In comparison to the best possible estimate, simple outlier rejection performs only slightly worse for this specific trajectory. We visualize the 3D trajectories in Fig. 4.

Additionally, we provide PoET's performance for the localization task for a different trajectory in Fig. 5. There we compare between PoET being provided either ground-truth or backbone bounding box information. In both cases, it is possible to estimate good trajectories based on PoET's relative object poses. However, PoET with backbone information performs slightly worse and is more prone to wrong estimates resulting in a noisier estimated trajectory.

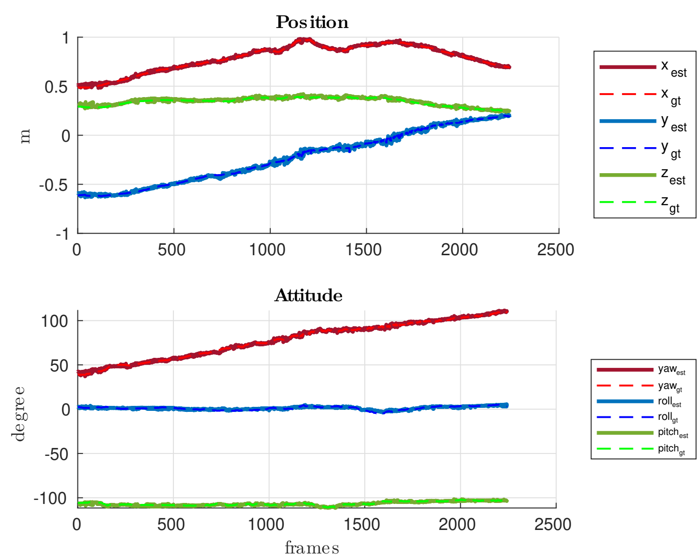
Summarizing, these results support our claim that PoET can be used as a pose sensor for the localization task in robotics applications.



(a) `out`

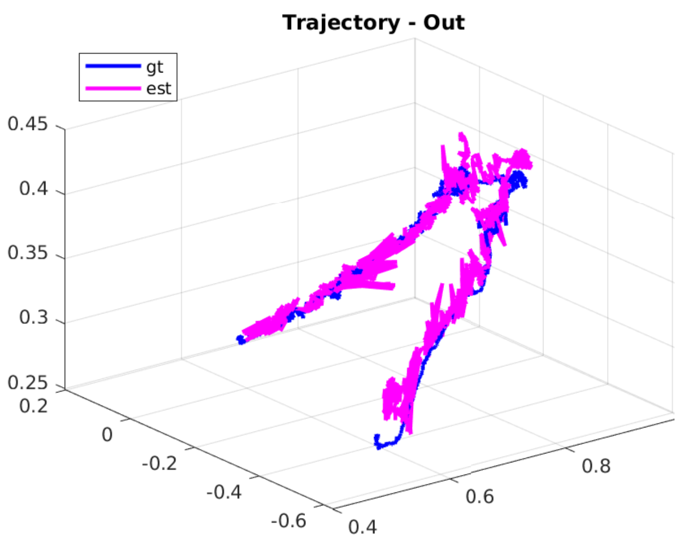


(b) `prev`

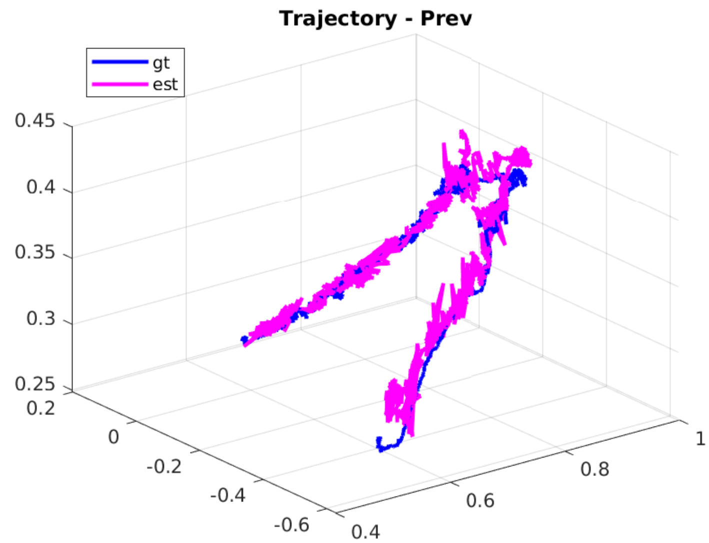


(c) `best`

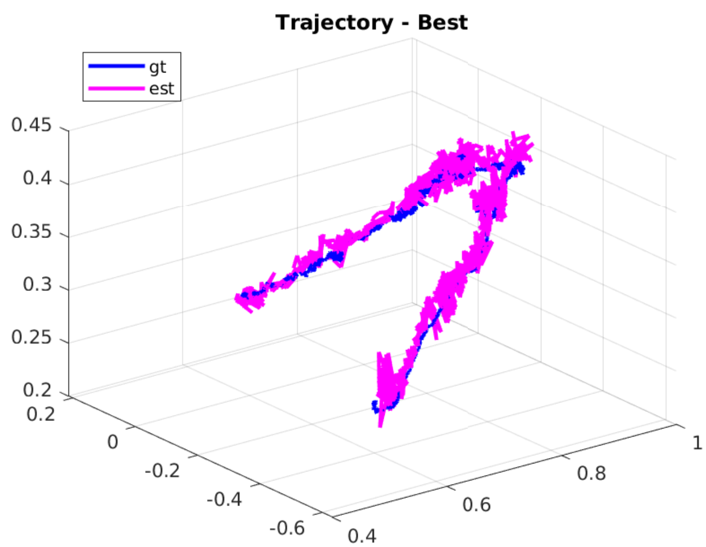
Figure 3: Estimated trajectories (`est`) for the same trajectory as in the main paper.



(a) `out`

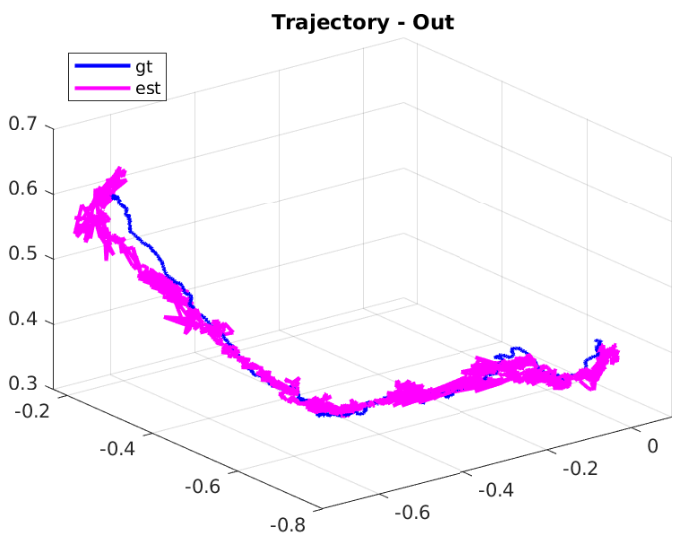


(b) `prev`

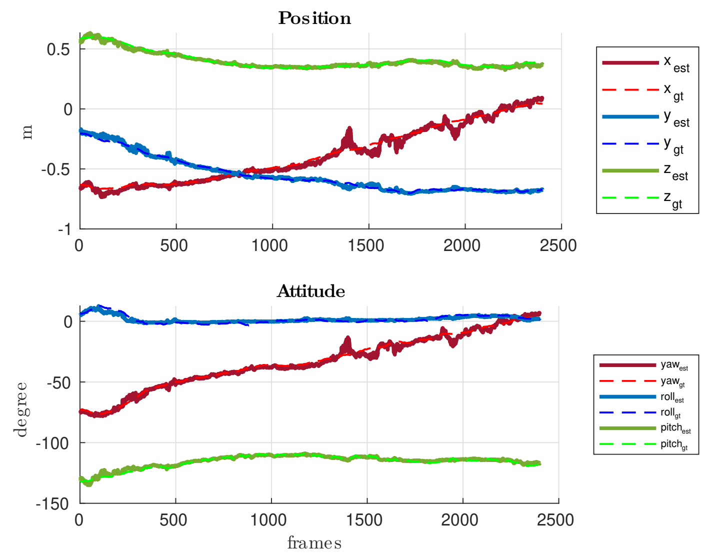


(c) `best`

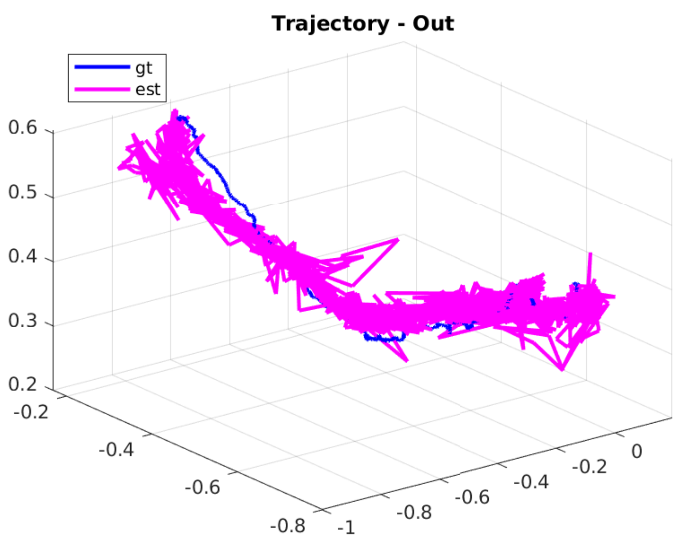
Figure 4: Visualization of the estimated trajectory for the same YCB-V scene shown in the main paper. We compare the estimated trajectory (`est` to the ground-truth trajectory (`gt`).



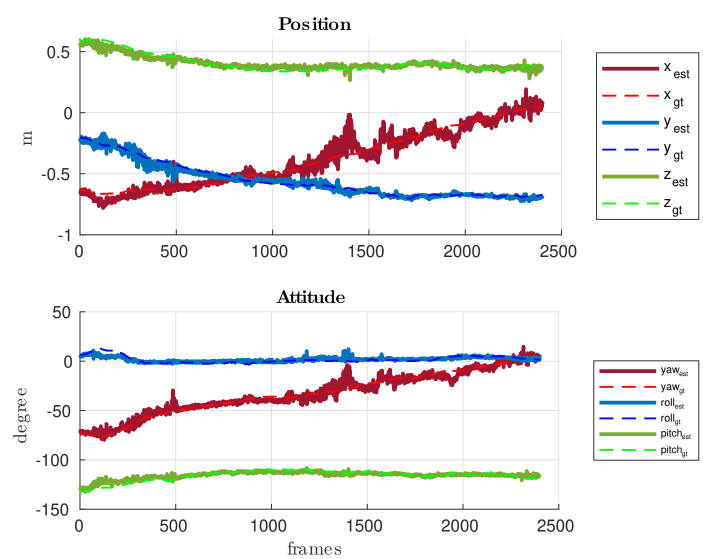
(a) `gt + out`



(b) `gt + out`



(c) `bb + out`



(d) `bb + out`

Figure 5: Comparison of estimated trajectories between PoET with ground-truth (`gt`) or predicted (`bb`) bounding box information. This is a different trajectory than the one used for discussion in the main paper.

7 Results on LM-O

In Table 8 we report the results of PoET for the ADD(-S) metric as described by Hinterstoisser et al. [15]. In contrast to other state-of-the-art approaches, we only train on pbr data as dictated by the BOP challenge [10]. Moreover, we only train a single PoET model, instead of training an individual network for each object class. Similarly as for YCB-V, methods that utilize either PnP or iterative refinement and thus 3D object models achieve state-of-the-art results. PoET outperforms other purely RGB-based approaches by more than 10 points.

Furthermore, we conduct a comparison between PoET trained on top of a pre-trained Mask R-CNN [5] for the LM-O dataset [1, 15] by evaluating it with ground-truth and predicted bounding boxes. Once again, evaluating with ground-truth bounding boxes results in better scores. Nevertheless, PoET still achieves state-of-the-art results for RGB-only methods when evaluated on actual predictions.

Table 8: Comparison with state-of-the-art on LM-O. We report the ADD(-S). PE indicates the number of pose estimators trained. N means that one PE was trained for each class. (*) denotes symmetric objects. `real`, `synt` and `pbr`, respectively, refer to real data, synthetically generated data by projecting the 3D models onto a black image background and photorealistic simulated images.

				PnP			IR
Method	PoseCNN [2]	PoET	PoET$_{gt}$	Pix2Pose [19]	PVNet [20]	GDR-Net [16]	DeepIM [11]
PE	1	1	1	N	N	N	1
Data	real + syn	pbr	pbr	real	real + syn	real + pbr	real + syn
Ape	9.6	10.2	**12.7**	22.0	15.8	46.8	59.2
Can	45.2	31.8	**51.0**	44.7	63.3	90.8	63.5
Cat	0.9	9.0	**10.9**	22.7	16.7	40.5	26.2
Driller	41.4	33.9	**53.3**	44.7	65.7	82.6	55.6
Duck	19.6	15.4	**22.6**	15.0	25.2	46.9	52.4
Eggbox*	22.0	44.7	**50.4**	25.2	50.2	54.2	63.0
Glue*	38.5	58.7	**63.9**	32.4	49.6	75.8	71.7
Holep.	22.1	24.7	**29.8**	49.5	36.1	60.1	52.5
MEAN	24.9	28.5	**36.8**	32.0	40.8	62.2	55.5

8 BOP Results

In the main paper we presented the results on YCB-V [2] following the most commonly used evaluation metrics [2, 11, 12, 13, 14, 16]. In recent years the evaluation protocol of the BOP challenge [10, 21] has become more popular and thus, we present the results of PoET for YCB-V on the BOP metrics. We refer the reader to [10] for a detailed explanation of the evaluation metrics. We report the average recall (AR) score by calculating the mean of the three main metrics: AR = (AR$_{MSPD}$ + AR$_{MSSD}$ + AR$_{VSD}$) / 3. The final results are summarized in Table 9. To the best of our knowledge, we are the only approach to report their results on the BOP challenge that relies solely on 2D image information. In contrast to the AUC of ADD-S metric, the metrics employed by the BOP challenge are stricter with respect to the objects final estimated rotation. Therefore, it is no surprise that state-of-the-art results are achieved by methods that either use 3D object model keypoints to predict the final pose (PnP) or that perform iterative refinement utilizing the 3D model given an initial estimate (IR). Nevertheless, PoET achieves competitive results and even outperforms the two PnP-based methods Pix2Pose [19] and CDPNv2 [22].

Table 9: Comparison of state-of-the-art methods on YCB-V for BOP metrics [10]. The results of other approaches are either obtained from the corresponding work or from the official leaderboard: `https://bop.felk.cvut.cz/leaderboards/`. PnP and IR stand for methods that either use PnP or iterative refinement respectively. 2D indicates that only RGB-image information was utilized.

Method	3D Model	AR$_{MSPD}$	AR$_{MSSD}$	AR$_{VSD}$	AR
PoET	2D	54.2	57.2	49.4	53.6
PoET$_{gt}$	2D	66.4	71.9	66.5	68.3
Pix2Pose [19]	PnP	57.1	42.9	37.2	45.7
EPOS [23]	PnP	78.3	67.7	62.6	69.6
CDPNv2 [22]	PnP	63.1	57.0	39.6	53.2
GDR-Net [16]	PnP	84.2	75.6	66.8	75.5
CosyPose [12]	IR	85.0	84.2	77.2	82.1

9 Quaternion Loss

In this section we compare different quaternion loss functions and their loss landscapes to each other. In the main paper we achieve state-of-the-art results on the YCB-V dataset [2] using a 6D rotation representation. PoET achieves similar results by using a quaternion representation and the loss function

$$L_{rot} = -\log(< q, \tilde{q} >^2 + \epsilon) \ , \tag{2}$$

where $< \cdot, \cdot >$ represents the regular vector dot product and ϵ is a small number for numerical stability. However, as shown in the ablation study, using the same quaternion loss function as SilhoNet [14] results in a worse performance. This loss function is given by

$$L_{rot} = \log(\epsilon + 1 - | < q, \tilde{q} > |) \ , \tag{3}$$

where $| \cdot |$ denotes the absolute value. Comparing the loss function landscape from Eq. (2) with the one from Eq. (3) in Fig. 6, it is observable that for small errors our loss converges to 0, while the loss from [14] goes to $-\infty$. The former will result in a more stable training towards the end due to smaller gradients. On the other hand, for large errors our loss function is close to ∞ and thus yielding stronger gradients, while the other loss function is close to 0. Having those smaller gradients towards the end of the training benefits the multi-task loss as the translation loss is not overshadowed by the rotation loss. Thus, the network can focus on further improving its performance with respect to the translation estimation as can be seen in Table 4. On the contrary, Table 5 shows that the large gradients of Eq. (3) for small errors influence PoET to achieve better average rotational error. In the end, being able to better estimate the object translation benefits the pose estimation task in terms of the AUC of ADD-S metric.

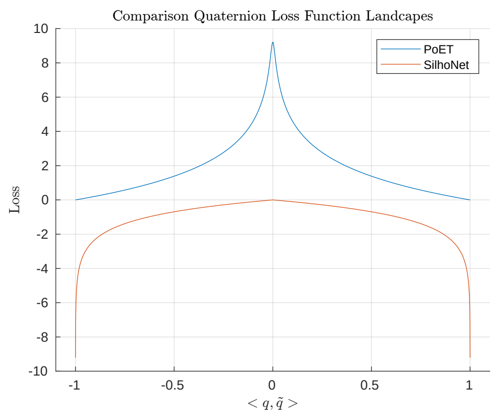


Figure 6: Comparison of the loss landscapes between our proposed quaternion loss function in Eq. (2) and the one used by [14] as shown in Eq. (3). For small errors, that is $< q, \tilde{q} >$ being close to -1 or 1, our loss function results in smaller gradients and thus allows our network to better learn. For both loss functions ϵ is set to $1\mathrm{e}-4$.

References

[1] A. Krull, E. Brachmann, F. Michel, M. Y. Yang, S. Gumhold, and C. Rother. Learning analysis-by-synthesis for 6d pose estimation in rgb-d images. In *Proceedings of the IEEE international conference on computer vision*, pages 954–962, 2015.

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

[3] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao. Scaled-yolov4: Scaling cross stage partial network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13029–13038, 2021.

[4] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014.

[5] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2961–2969, 2017.

[6] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations (ICLR)*, 2021.

[7] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[8] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. *International Conference on Learning Representations (ICLR)*, 2019.

[9] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar. The ycb object and model set: Towards common benchmarks for manipulation research. In *2015 International Conference on Advanced Robotics (ICAR)*, pages 510–517. IEEE, 2015.

[10] T. Hodaň, M. Sundermeyer, B. Drost, Y. Labbé, E. Brachmann, F. Michel, C. Rother, and J. Matas. Bop challenge 2020 on 6d object localization. In *European Conference on Computer Vision*, pages 577–594. Springer, 2020.

[11] Y. Li, G. Wang, X. Ji, Y. Xiang, and D. Fox. Deepim: Deep iterative matching for 6d pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 683–698, 2018.

[12] Y. Labbé, J. Carpentier, M. Aubry, and J. Sivic. Cosypose: Consistent multi-view multi-object 6d pose estimation. In *European Conference on Computer Vision*, pages 574–591. Springer, 2020.

[13] C. Li, J. Bai, and G. D. Hager. A unified framework for multi-view multi-class object pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 254–269, 2018.

[14] G. Billings and M. Johnson-Roberson. Silhonet: An rgb method for 6d object pose estimation. *IEEE Robotics and Automation Letters*, 4(4):3727–3734, 2019.

[15] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *Asian Conference on Computer Vision*, pages 548–562. Springer, 2012.

[16] G. Wang, F. Manhardt, F. Tombari, and X. Ji. Gdr-net: Geometry-guided direct regression network for monocular 6d object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16611–16621, 2021.

[17] Z. Tian, C. Shen, H. Chen, and T. He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9627–9636, 2019.

[18] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 28, 2015.

[19] K. Park, T. Patten, and M. Vincze. Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7668–7677, 2019.

[20] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao. Pvnet: Pixel-wise voting network for 6dof pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4561–4570, 2019.

[21] T. Hodan, F. Michel, E. Brachmann, W. Kehl, A. GlentBuch, D. Kraft, B. Drost, J. Vidal, S. Ihrke, X. Zabulis, et al. Bop: Benchmark for 6d object pose estimation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 19–34, 2018.

[22] Z. Li, G. Wang, and X. Ji. Cdpn: Coordinates-based disentangled pose network for real-time rgb-based 6-dof object pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7678–7687, 2019.

[23] T. Hodan, D. Barath, and J. Matas. Epos: Estimating 6d pose of objects with symmetries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11703–11712, 2020.