DAPO: Self-Supervised Domain Adaptation for 6DoF Pose Estimation

Juseong Jin*
Seoul National University
jinju4948@snu.ac.kr

Eunju Jeong*
ETRI
eunju@etri.re.kr

Joonmyun Cho
ETRI
jmcho@etri.re.kr

Joon Hee Park
ETRI
juni@etri.re.kr

Young-Gon Kim
Seoul National University Hospital
younggon2.kim@gmail.com

Abstract

The main challenge of pose estimation for six degrees of freedom (6DoF) is the lack of labeled data in real environment. In order to overcome this problem, many studies recently have trained deep learning models with synthetic data. However, a domain gap between real and synthetic environments exists, prompting various approaches to address this issue. In this work, we propose domain adaptation for self-supervised 6DoF pose estimation (DAPO), which leverages the components and introduces an effective method to reduce domain discrepancy. First, we adopt a multi-level domain adaptation module, on image level and instance level, to learn domain-invariant features. Second, we used entropy-based alignment to minimize the entropy of representation embedding. Finally, we evaluate our approach on LineMOD and Occlusion-LineMOD datasets. Experiments show that our proposed method achieves higher performance compared to the prior methods and demonstrate effectiveness in domain shift scenarios on 6DoF pose estimation.

1 Introduction

Six-degree-of-freedom (6DoF) pose estimation has been receiving increasing attention over the last few years because of the increase in the number of potential applications of robot manipulation systems [1]. Many studies are being actively conducted on applying deep learning to 6DoF pose estimation [2, 3, 4]. However, the direct application of deep learning approaches to 6DoF pose estimation is difficult due to the lack of labeled data from real environments. Moreover, 6DoF pose annotation is expensive and requires a lot of labor. Existing works address this issue by generating synthetic data virtually. The benefits of using synthetic data are that it simplifies the generation of accurate ground truth and costs less. In contrast to these advantages, a domain discrepancy exists between real data (target) and synthetic data (source). Some work has created source data similar to the target, reducing the gap in a domain to improve performance [5] but less scalable to other types of data. Some work uses the depth map to align the two domain samples [6]. Unsupervised domain adaptation (UDA) has been widely studied [7, 8, 9] to overcome the discrepancy. To overcome the problem of cross-domain 6DoF pose estimation in such a constrained environment, we propose a novel approach for 6DoF pose estimation using self-supervised technique. Specifically, we examine the scenario of unsupervised domain adaptation, where full supervision exists in the source domain but no supervision is available in the target domain. Our contributions are as follows:

*These authors contributed equally

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

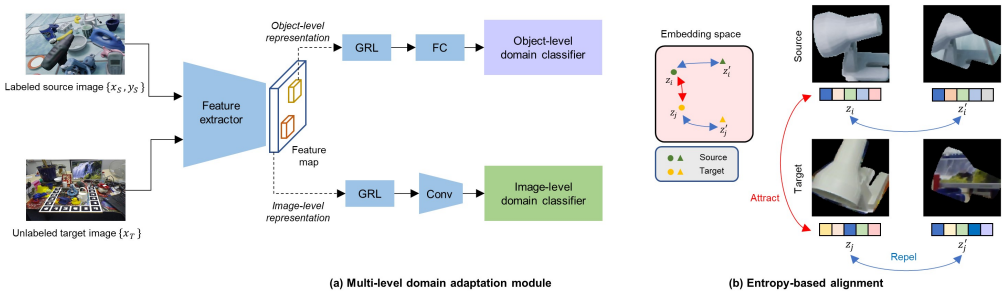

Figure 1: The framework of our proposed algorithm

- We design an unsupervised domain adaptation pipeline for 6DoF pose estimation for RGB image, which consists of a multi-level domain adaptation module and entropy-based alignment.
- We propose an entropy-based alignment which train to capture visual representative features better.
- Our approach achieves higher performance on the LineMOD dataset and LineMOD-Occlusion dataset.

2 Related Works

Six DoF pose estimation is the task of determining the 6DoF pose of an object in 3D space based on images. Recent advancements in deep learning have shown remarkable performance in 6DoF pose estimation from RGB images, primarily due to the availability of large-scale training data[10, 11]. For instance, PVNet [12] utilizes a regression-based approach to predict pixel-wise unit vectors pointing to key points. These vectors are then used to vote for key-point locations using the RANSAC method. EfficientPose [13] is another approach that detects 2D targets to solve the PnP problem for estimating the 6D pose based on EfficientNet [14]. In our work, we adopt EfficientPose as the base estimator due to its impressive performance and efficiency.

UDA aims to learn domain-invariant features from labeled source data, which can then be applied to unlabeled target data. Recently, several research has emerged in UDA [15, 16, 17]. In particular, adversarial training (AT) has been widely used in UDA in recent years. AT utilizes a domain discriminator to predict the domain of the input image and it tries to extract domain-invariant features by fooling the domain discriminator [18, 19]. Recent self-supervised learning (SSL) with unlabeled data has emerged[20]. In particular, contrastive learning[21, 22] is commonly used to enhance representation learning by training models to maximize similarity between positive pairs and minimize similarity between negative pairs in the embedding space.

3 Methods

Labeled data for source domain is represented as $S = (x_S, y_S)$, where x_S is source images and y_S is the label for the corresponding 6DoF pose. The unlabeled data for the target domain is represented as $T = x_T$, where x_T, denotes a target image without annotation.

3.1 Multi-level Domain Adaptation

The $\mathcal{H}$-divergence is utilized to measure between two set of samples with different distributions. That is, the $\mathcal{H}$-divergence relies on the capacity of the hypothesis class $\mathcal{H}$-divergence to distinguish between examples generated by D_S^x from examples generated by D_T^x. Let us denote $\mathcal{H}$ is the domain classifier, the $\mathcal{H}$-divergence defines the distance between two domains as follows:

$$d_H(S,T) = 2(1 - \min_{h \in H}(err_S(h(x)) + err_T(h(x)))) \quad (1)$$

where err_S and err_T are the prediction errors of $h(x)$ on source and target domain samples. Let us denote by f the network that produces x. To we need to enforce the networks f to output feature vectors that minimize the domain distance $d_H(S,T)$.

Image-level adaptation The image-level representation refers to the feature map of the feature extractor. Let us denote by i-th image, D_i (D = 0 for source and D =1 for target domains, respectively) and prediction of the domain classifier as p_i . The domain classifier is trained by minimizing the cross-entropy loss as follows:

$$L_{img} = -\sum_i [D_i \log p_i + (1 - D_i) \log(1 - p_i)] \tag{2}$$

Object-level adaptation The instance-level representation refers to region of interest based-feature vectors. Similar to the image-level adaptation, we train domain classifier to learn domain-invariant feature. Denoting the output of the instance-level domain classifier for the j-th region proposal in the i-th image as $p_{i,j}$.

$$L_{obj} = -\sum_{i,j} [D_i \log p_{i,j} + (1 - D_i) \log(1 - p_{i,j})] \tag{3}$$

The loss computed using Eq 2, 3 is minimized so that the features extracted from the two domains are indistinguishable to the discriminator. For the implementation we use the gradient reverse layer.

3.2 Entropy-based Alignment

We propose entropy-based alignment to minimize the distance of representation embedding between the source and target domain. To this end, we construct positive pairs to aid in the convergence of representations, as well as negative pairs to amplify their divergence.

We utilize an RGB image that reflects the pose by masking created by projecting transformation. The masking is constructed by dot producting the transformation consisting of a rotation R and a translation t and the point of the object P. Pose reflecting image is embedded into a representation vector z by head h consisting of convolution layers and multi-layer perceptrons. Let us denote representation embedding from the source sample as z_i, from the target of the same class as z_j. To create negative pairs, gaussian noise is used at transformation value. Let us denote $z_i^{'}$ and $z_j^{'}$ negative pairs with noise added for z_i and z_j, respectively. To calculate the representation embedding similarity between each pair, we conduct a dot product between the embeddings. Then, the entropy for positive pair is calculated as

$$p_{po} = \frac{\exp(z_i, z_j/\tau)}{\exp(z_i, z_j/\tau) + \sum_k^N \exp(z_i, z_{i,k}^{'})/\tau) + \sum_k^N \exp(z_j, z_{j,k}^{'}/\tau)} \tag{4}$$

In the same way, the entropy for negative pair is calculated as

$$p_{ne} = \frac{1}{\sum_k^N \left(\exp(z_i, z_{i,k}^{'}/\tau) + \exp(z_j, z_{j,k}^{'}/\tau)\right)} \tag{5}$$

where temperature parameter τ controls the centration degree and k is the index of the negative pairs. By combining equations (4-5), we note the our entropy-based alignment loss L_{EA}

$$L_{EA} = -\log\left(\frac{p_{po}}{p_{po} + p_{ne}}\right) \tag{6}$$

Note that in general contrastive learning, positive pairs are typically defined as pairs originating from the same image, while negative pairs are formed by combining images from different sources. However, to minimize the domain discrepancy, we designate pairs from different domains as positive pairs.

4 Experiments

4.1 Experimental Settings

Dataset Our approach was verified by conducting experiments using two widely recognized datasets for 6DoF pose estimation in RGB images. The first dataset is called LineMOD, which is the

Table 1: Quantitative results for LineMOD

	Ape	*Bvis*	*Cam*	*Can*	*Cat*	*Dril*	*Duck*	*Eggb*	*Glue*	*Holp*	*Iron*	*Lamp*	*Phone*	*Avg*
baseline	23.5	80.7	37.1	80.2	46.5	83.5	19.3	73.1	64.0	9.7	88.7	60.5	57.5	55.7
pseudo [26]	23.6	65.7	37.8	81.1	43.2	85.2	18.6	38.4	66.0	8.7	89.9	66.8	62.6	52.8
DANN [27]	22.4	83.0	42.1	83.6	47.8	85.5	21.3	79.4	66.0	13.5	90.1	68.8	62.7	58.8
DPOD [10]	35.1	59.4	15.5	48.8	28.1	59.3	25.6	51.2	34.6	17.7	84.7	45.0	20.9	40.5
Self6D [28]	38.9	75.2	36.9	65.6	57.9	67.0	19.6	**99.0**	**94.1**	15.5	77.9	68.2	50.1	58.9
Cons. [29]	**37.6**	78.6	**65.5**	65.6	52.5	48.8	**35.1**	89.2	64.5	**41.5**	80.9	70.7	60.5	60.6
Ours	25.5	**84.9**	45.2	**86.1**	**52.6**	**88.4**	26.5	92.9	88.0	18.0	**92.1**	**74.5**	**68.6**	**64.9**

Table 2: Quantitative results for Occlusion-LineMOD

	Ape	*Can*	*Cat*	*Dril*	*Duck*	*Eggb*	*Glue*	*Holp*	*Avg*
baseline	6.4	36.7	5.5	38.9	13.8	39.1	32.4	24.1	22.8
pseudo	6.2	36.0	5.2	39.1	13.4	39.2	32.7	22.8	22.3
DANN	6.5	37.1	5.4	40.4	14.0	39.4	34.5	24.8	24.7
CDPN [11]	**20.0**	15.1	**16.4**	5.0	22.2	36.1	27.9	24.0	20.8
Cons.	12.0	27.5	12.0	20.5	**23.0**	25.1	27.0	**35.0**	22.8
Ours	6.6	**39.0**	5.6	**40.9**	14.5	**40.0**	**36.8**	25.0	**26.0**

benchmark dataset used for the 6DoF pose estimation. The second dataset is called Occlusion-LineMOD, which is a subset of the LineMOD dataset and consists of a single scene in which 8 heavily occluded objects are annotated in a video sequence. We followed the same protocol followed by previous works to split the dataset [23].

Synthetic dataset We employed the synthetic data, published in [24], which was generated by OpenGL-based render. Lightweight and backgrounds were randomly selected and objects were scattered based on physical engine. Its scale and field of view was also randomly generated.

4.2 Evaluation metric

We evaluate our approach with the commonly used ADD(-S) metric[25]. This metric calculates the average point distances between the 3D model point set M, transformed with the ground truth rotation R and translation t and the model point set transformed with the estimated rotation $\hat{R}$ and translation $\hat{t}$. For asymmetric objects the ADD metric is defined as follows:

$$ADD = \frac{1}{m} \sum_{x \in M} \|(Rx + t) - (\hat{R}x + \hat{t})\|_2 \tag{7}$$

Symmetric objects are evaluated using the ADD-S metric which is given by following equation

$$ADD - S = \frac{1}{m} \sum_{x \in M} min\|(Rx + t) - (\hat{R}x + \hat{t})\|_2 \tag{8}$$

An estimated 6D pose is considered correct if the average point distance is smaller than 10% of the object's diameter.

4.3 Quantitative Results

We evaluate the performance on the LineMOD dataset and Occlusion-LineMOD dataset, in terms of ADD(-S) metric. We show quantitative results in Table 1, 2. Compared to the performance of the popular domain adaptation method with the baseline and the 6DoF pose estimation studies with the synthetic data, we achieve the highest performance. We achieved 9.2% and 3.2% performance improvement over the baseline. Although the baseline performed well, the performance improvement is greater with the proposed method than with the other DA methods applied to the baseline and 6DoF pose estimation methods. The results shows the effectiveness of DAPO in 6DoF pose estimation.

5 Conclusions

We propose an approach for self-supervised domain adaptation for 6DoF pose estimation. Multi-level domain adaptation module allows to learn domain-invariant features. In addition, entropy-

based alignment is designed to transfer knowledge from the source domain and reduce the distance representation embedding by minimizing the entropy of two domains. As a result, we focused on visual features through self-training without adding other modalities. Experiments on two benchmark datasets show the effectiveness of our approach. Although performance is not high for some objects, we believe that our work makes a significant contribution, which understands visual features well and for potential applications in robot manipulation systems.

6 Acknowledgments

This work was supported by Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government. [23ZR1100, A Study of Hyper-Connected Thinking Internet Technology by autonomous connecting, controlling and evolving ways]. Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No.2022-0-00187, Development of an edge brain framework to make manufacturing equipment and robots intelligent)
This research was supported by a grant of Patient-Centered Clinical Research Coordinating Center (PACEN) funded by the Ministry of Health & Welfare, Republic of Korea (grant number: HC19C0164)

References

[1] Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *Proceedings of the IEEE international conference on computer vision*, pages 2794–2802, 2015.

[2] Kiru Park, Timothy Patten, and Markus 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.

[3] Yan Xu, Kwan-Yee Lin, Guofeng Zhang, Xiaogang Wang, and Hongsheng Li. Rnnpose: Recurrent 6-dof object pose refinement with robust correspondence field estimation and pose optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14880–14890, 2022.

[4] Shun Iwase, Xingyu Liu, Rawal Khirodkar, Rio Yokota, and Kris M Kitani. Repose: Fast 6d object pose refinement via deep texture rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3303–3312, 2021.

[5] Zhigang Li, Yinlin Hu, Mathieu Salzmann, and Xiangyang Ji. Sd-pose: Semantic decomposition for cross-domain 6d object pose estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2020–2028, 2021.

[6] Gu Wang, Fabian Manhardt, Xingyu Liu, Xiangyang Ji, and Federico Tombari. Occlusion-aware self-supervised monocular 6d object pose estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[7] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015.

[8] Riccardo Volpi, Pietro Morerio, Silvio Savarese, and Vittorio Murino. Adversarial feature augmentation for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5495–5504, 2018.

[9] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive faster r-cnn for object detection in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3339–3348, 2018.

[10] Sergey Zakharov, Ivan Shugurov, and Slobodan Ilic. Dpod: 6d pose object detector and refiner. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1941–1950, 2019.

[11] Zhigang Li, Gu Wang, and Xiangyang 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.

[12] Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun 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.

[13] Yannick Bukschat and Marcus Vetter. Efficientpose: An efficient, accurate and scalable end-to-end 6d multi object pose estimation approach. *arXiv preprint arXiv:2011.04307*, 2020.

[14] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[15] Fabio Maria Carlucci, Lorenzo Porzi, Barbara Caputo, Elisa Ricci, and Samuel Rota Bulo. Autodial: Automatic domain alignment layers. In *Proceedings of the IEEE international conference on computer vision*, pages 5067–5075, 2017.

[16] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2517–2526, 2019.

[17] A Tuan Nguyen, Toan Tran, Yarin Gal, Philip HS Torr, and Atılım Güneş Baydin. Kl guided domain adaptation. *arXiv preprint arXiv:2106.07780*, 2021.

[18] Chen Li and Gim Hee Lee. From synthetic to real: Unsupervised domain adaptation for animal pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1482–1491, 2021.

[19] Farzaneh Rezaeianaran, Rakshith Shetty, Rahaf Aljundi, Daniel Olmeda Reino, Shanshan Zhang, and Bernt Schiele. Seeking similarities over differences: Similarity-based domain alignment for adaptive object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9204–9213, 2021.

[20] Linus Ericsson, Henry Gouk, Chen Change Loy, and Timothy M Hospedales. Self-supervised representation learning: Introduction, advances, and challenges. *IEEE Signal Processing Magazine*, 39(3):42–62, 2022.

[21] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[22] Mamatha Thota and Georgios Leontidis. Contrastive domain adaptation. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 2209–2218, 2021.

[23] Stefan Hinterstoisser, Stefan Holzer, Cedric Cagniart, Slobodan Ilic, Kurt Konolige, Nassir Navab, and Vincent Lepetit. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In *2011 international conference on computer vision*, pages 858–865. IEEE, 2011.

[24] Jonathan Tremblay, Thang To, Balakumar Sundaralingam, Yu Xiang, Dieter Fox, and Stan Birchfield. Deep object pose estimation for semantic robotic grasping of household objects. *arXiv preprint arXiv:1809.10790*, 2018.

[25] Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary Bradski, Kurt Konolige, and Nassir 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.

[26] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896. Atlanta, 2013.

[27] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

[28] Gu Wang, Fabian Manhardt, Jianzhun Shao, Xiangyang Ji, Nassir Navab, and Federico Tombari. Self6d: Self-supervised monocular 6d object pose estimation. In *European Conference on Computer Vision*, pages 108–125. Springer, 2020.

[29] Juil Sock, Guillermo Garcia-Hernando, Anil Armagan, and Tae-Kyun Kim. Introducing pose consistency and warp-alignment for self-supervised 6d object pose estimation in color images. In *2020 International Conference on 3D Vision (3DV)*, pages 291–300. IEEE, 2020.

[30] Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(11):4037–4058, 2020.

[31] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[32] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15750–15758, 2021.

[33] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[34] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[35] Tony Lindeberg. Scale invariant feature transform. 2012.

[36] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. Ieee, 2011.

[37] Chen Song, Jiaru Song, and Qixing Huang. Hybridpose: 6d object pose estimation under hybrid representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 431–440, 2020.

Appendix

A Supplement of Related Work

Self-supervised Representation Learning Self-supervised learning methods learn representations from unlabeled datasets where annotations are scarce. In vision application, the pretext task is typically to maximaize similarity between two augmented views of the same image [30]. This can be done in a contrastive learning using the InfoNCE loss[31], as in SimCLR[32] and MoCO[33].

6DoF pose estimation using RGB images Conventional methods for 6DoF pose estimation typically involve extracting features from RGB images and performing feature matching to achieve consensus [34]. These methods commonly employ hand-crafted features like SIFT [35] and ORB [36] due to their robustness against scale and rotation. The 6DoF pose is then estimated by solving a Perspective-n-Point (PnP) problem using these features [37]. Recent advancements in deep learning have shown remarkable performance in 6DoF pose estimation from RGB images. CDPN [11] disentangles the pose by separately predicting rotation and translation, resulting in highly accurate and robust pose estimation. DPOD [10] estimates dense multi-class 2D-3D correspondences between an input image and available 3D models.

B Implementation Details

We adopt the unsupervised domain adaptation protocol in our experiments. The training data consists of two parts: the source training data for which images and their annotations and the target training data for which unlabeled images. The domain classifier and projection head has a convolutional architecture, which consists of four convolutional layers and two linear layers which Relu as the activation function. We set the image to 512 pixels, following the EfficientPose implementation. For the transformation value, we use ground truth for the source domain and prediction for the target domain. Models using adversarial training and alignment are trained with learning rate 1e-6, clipnorm 1e-5. Each batch is composed of 16 images, half from each domain. With the mentioned setting, NVIDIA RTX A6000 GPU is used. Both methods are implemented with Tensorflow.

C Pose Reflecting RGB Image Extraction

Algorithm 1 Pose reflecting rgb image extraction algorithm

procedure EXTRACT2DIMAGE($P, R, t,$ img)
  Input:
  P: Point set of the object model
  R: Rotation matrix
  t: Translation vector
  img: RGB image
  Output:
  I_{2d}: 2D image reflecting the pose
  $Tv \leftarrow P \cdot R + t$ ▷ Calculate Transformation Value
  $I_{2d} \leftarrow Tv * \text{img}$ ▷ Project Transformation Value onto RGB Image
  return I_{2d} ▷ Return the 2D Image
end procedure

D Qualitative Results

We visualize the 6DoF pose by overlaying the image with the corresponding transformed 3D bounding box. For LineMOD dataset, *Green* refers the ground truth pose and *Blue* refers prediction box. (a), (d) Target images x_T from LineMOD; (b), (e) Baseline results; (c), (f) Predictive results from our method, respectively. For Occusion-LineMOD, *Green* refers the ground truth pose and *other colors* refers prediction box. (g) Target images x_T from Occlusion-LineMOD; (h) Baseline results; (i) Predictive results from our method, respectively. In LineMOD dataset, as shown in Figure 3 (b),

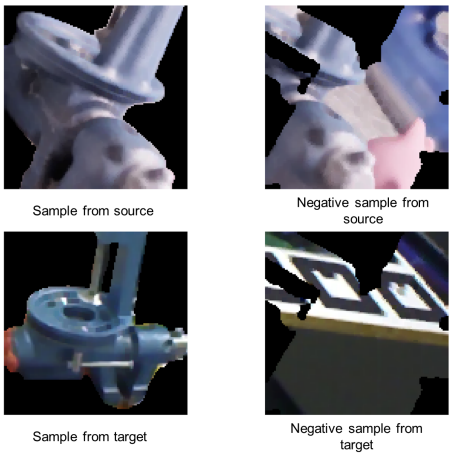

Figure 2: Example of pose reflecting image. Regions that do not reflect pose are masked in black. We add Gaussian noise to the transformation value to get a negative pair sample

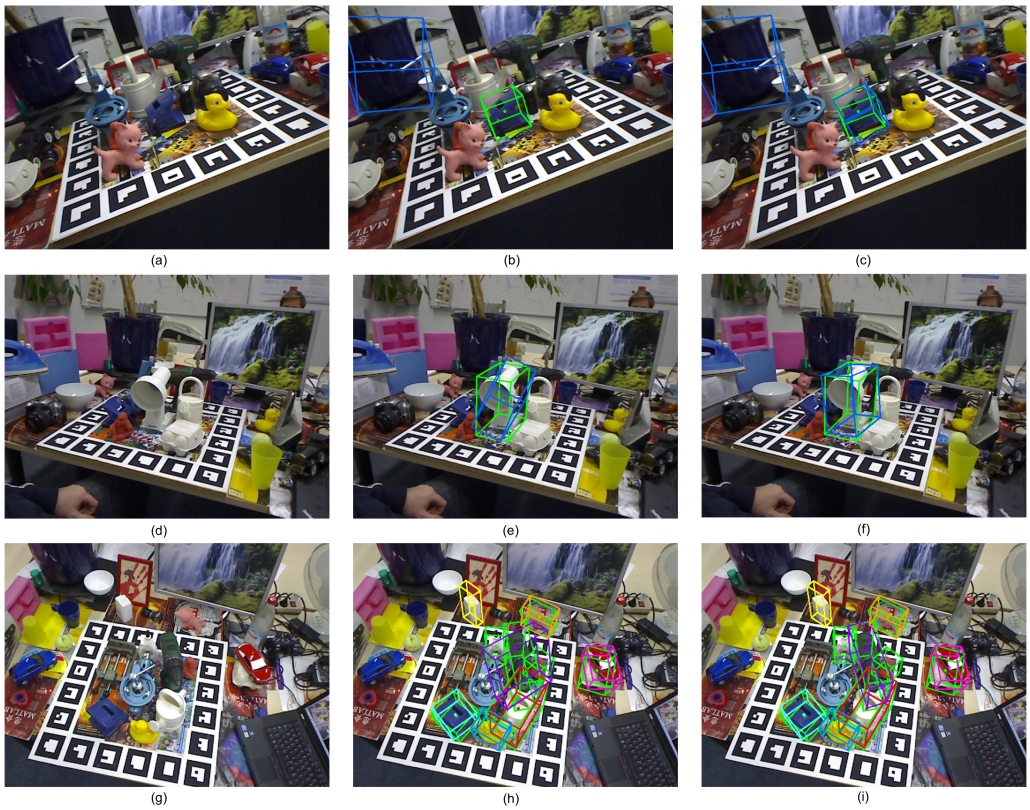

Figure 3: Qualitative results on LineMOD and Occlusion-LineMOD

the baseline model incorrectly predict a monitor with a color similar to the ground truth, Holep. In constrast, as shown in Figure 3 (c), the pose of the bounding box predicted with our model predict Holep., accurately. As shown in Figure 3 (e), the baseline model rotated the predicted bounding box incorrectly. In contrast, the pose of the bounding box predicted with our model was accurate, especially the rotations in three axes, as shown in Figure 3 (f). The pose estimation result of the Drill object (colored in purple) was incorrect, especially for rotation in three axes, as shown in Fig 3 (h). In contrast, the box was aligned more accurately by a model. Our method demonstrates the highest performance in complex situations where objects similar to real-world application steps are occluded.