Supplementary Material of CheckerPose: Progressive Dense Keypoint Localization for Object Pose Estimation with Graph Neural Network

Ruyi Lian Haibin Ling
Department of Computer Science, Stony Brook University, Stony Brook, NY 11794-2424, USA
{rulian,hling}@cs.stonybrook.edu

1. Hyper-parameters in the Pose Solver

We use both RANSAC/PnP [15] and Progressive-X [1] when evaluating the results on the LM dataset [6], and we use Progressive-X for LM-O [2] and YCB-V [24] datasets. For both pose solvers, we set the threshold of reprojection error as 2 pixels. We run 150 iterations when using RANSAC/PnP and run 400 iterations when using Progressive-X.

2. Additional Ablation Experiments on LINEMOD Dataset

Theoretically, increasing the number of keypoint N leads to more candidate 3D-2D correspondences and enhances the robustness of pose estimation. In our current implementation, we adopt $k = 20$ in EdgeConv following [23], and $N = 512$ based on our available computation resources. We also conduct ablation studies of N and k on the LM dataset in Table 1, showing that larger N and k help improve the performance.

N	k	ADD(-S)			2°2cm	5°5cm
		0.02d	0.05d	0.1d		
128	10	29.4	81.3	96.4	75.6	98.8
	15	29.2	81.0	96.1	74.8	98.6
	20	29.8	82.0	96.5	77.6	98.7
256	10	**36.0**	84.2	96.8	79.1	98.8
	15	32.0	83.4	96.8	78.1	98.8
	20	33.6	82.9	96.4	75.8	98.8
512	10	30.4	82.0	96.6	76.3	98.7
	15	29.9	82.8	96.3	76.4	98.5
	20	35.7	**84.5**	**97.1**	**79.7**	**98.9**

Table 1: **Ablation Study of N and k on the LM Dataset.**

3. Filtering Operation on LM-O and YCB-V

As discussed in the main paper, we empirically find that for a textureless object O with severe self-occlusions, filtering out the correspondences outside the visible segmentation masks M_{vis} can improve the pose estimation results. We quantify the self-occlusions of O using $r_{\text{so}}(O)$. As a common practice, the visibility of point $P \in O$ from each viewpoint can be determined by checking the intersections between the camera rays and the object mesh. However, this may produce undesired results for our task. For example, the mesh of the bowl in the YCB-V dataset can be treated as a half sphere with very small thickness. When sampling the dense keypoints from the surface, we get keypoints from both outer side and inner side. For the keypoint on the inner side of the bowl, it is considered as easily self-occluded when we use ray intersections to determine the visibility. However, since the bowl is textureless and the thickness of the mesh can be ignored, the keypoint is equivalent to the nearest surface point on the outer side, and should not be considered as easily self-occluded. Considering this issue and the slow computation speed, we instead use Hidden Point Removal (HPR) operator [13] to estimate the proportion $V(P)$ of the viewpoints for which P is visible. For a keypoint with high $V(P)$, it may be consistently misclassified as invisible by the HPR operator, so we ignore the points with $V(P) < 0.2$ estimated by the HPR operator.

We report the value of $r_{\text{so}}(O)$ for each object O of the LM-O dataset in Table 2. Since these objects do not have strong textures, we apply the filtering operation during the inference for the objects with $r_{\text{so}}(O) \geq 0.5$.

For the objects that requires filtering operation, we report the ADD(-S) metric without filtering in Table 3. We also report the results of using different segmentation masks to filter the correspondences in Table 3. Without the filtering operation, the ADD(-S) values decreases for all the objects. Since all the 2D projections should be located within the full segmentation mask M_{full}, using M_{full} to filter the correspondences aims to discard the wrong predictions out-

Object	r_{so}	filtering
ape	0.356	✗
can	0.650	✓
cat	0.584	✓
driller	0.657	✓
duck	0.483	✗
eggbox	0.529	✓
glue	0.362	✗
holep.	0.354	✗

Table 2: **Quantitative measure r_{so} of the self-occlusions of the objects on LM-O [2].** Since the objects do not have strong textures, for the objects with $r_{\text{so}} \geq 0.5$, we apply the filtering operation during inference, *i.e.*, discarding the correspondences outside the visible segmentation masks.

Object	w/o Filter	w/ Filter (M_{full})	w/ Filter (M_{vis})
can	95.2	95.1	**95.7**
cat	62.0	61.3	**62.3**
driller	92.6	92.6	**93.7**
eggbox	68.8	69.6	**70.0**

Table 3: **ADD(-S) metrics on LM-O [2] w.r.t. the filtering operation.** "w/o Filter" denotes using all predicted correspondences to compute the pose. "w/ Filter (M_{full})" denotes discarding the correspondences outside the full segmentation mask M_{full}, while "w/ Filter (M_{vis})" denotes discarding the correspondences outside the full segmentation mask M_{vis}.

side the object area. However, it does not improve the final estimations consistently, which indicates that we still need to discard more unstable correspondences within the object area.

We report the values of r_{so} for the textureless objects in the YCB-V dataset in Table 4. According to Table 4, only one textureless object, *i.e.*, 061_foam_brick, requires filtering operation due to severe self-occlusions.

We further report the ADD(-S) metric w.r.t. the filtering operation for 061_foam_brick in Table 5. The ADD(-S) of 061_foam_brick remains the same without filtering operation or using M_{full} rather than M_{vis} in the filtering operation. This observation suggests that the localization of the easily self-occluded regions may become stable after 380,000 training steps. We further investigate the results of 061_foam_brick after different training steps in Table 6. After 200,000 steps, the ADD(-S) without filtering is inferior to the result of discarding correspondences outside M_{vis}. This observation implies that the localization of the easily self-occluded regions are unstable with fewer training steps.

Besides textureless objects with severe self-occlusions,

Object	r_{so}	filtering
011_banana	0.240	✗
019_pitcher_base	0.221	✗
024_bowl	0.498	✗
025_mug	0.108	✗
036_wood_block	0.438	✗
037_scissors	0.365	✗
051_large_clamp	0.163	✗
052_extra_large_clamp	0.138	✗
061_foam_brick	0.542	✓

Table 4: **Quantitative measure r_{so} of the self-occlusions of the textureless objects on YCB-V [24].** For the object with $r_{\text{so}} \geq 0.5$, we apply the filtering operation during inference, *i.e.*, discarding the correspondences outside the visible segmentation masks.

Object	w/o Filter	w/ Filter (M_{full})	w/ Filter (M_{vis})
008_pudding_box	66.4	71.0	**86.5**
061_foam_brick	**87.2**	**87.2**	**87.2**

Table 5: **ADD(-S) metrics on YCB-V [24] w.r.t. the filtering operation.** "w/o Filter" denotes using all predicted correspondences to compute the pose. "w/ Filter (M_{full})" denotes discarding the correspondences outside the full segmentation mask M_{full}, while "w/ Filter (M_{vis})" denotes discarding the correspondences outside the full segmentation mask M_{vis}.

Steps	w/o Filter	w/ Filter (M_{full})	w/ Filter (M_{vis})
200k	86.1	85.4	86.8
380k	87.2	87.2	87.2

Table 6: **ADD(-S) metrics of 061_foam_brick with different training steps.** "w/o Filter" denotes using all predicted correspondences to compute the pose. "w/ Filter (M_{full})" denotes discarding the correspondences outside the full segmentation mask M_{full}, while "w/ Filter (M_{vis})" denotes discarding the correspondences outside the full segmentation mask M_{vis}.

we also apply filtering operation on 008_pudding_box from the YCB-V dataset. As shown in Figure 1, 008_pudding_box is severely occluded by 009_gelatin_box. We regard 009_gelatin_box as a distraction object for the keypoint localization task of 008_pudding_box, since these objects share similar appearances, especially the texts (*i.e.*, "JELL-O"). Such severe occlusions by the same distraction object exist in all the test images of 008_pudding_box, and can be automatically detected by checking the object detection results. Thus we discard the correspondences out-

Figure 1: **Example of test images for 008_pudding_box from the YCB-V dataset.** We visualize the zoomed-in RoI based on the detection results. For all test images, 008_pudding_box (the brown box) is severely occluded by 009_gelatin_box (the red box).

side M_{vis} to remove the unstable localization results due to the occlusions by the distraction object. We also report the ADD(-S) metric without filtering and using M_{full} in filtering in Table 5. Using either M_{full} or M_{vis} to filter the correspondences improve the pose estimation results compared with using all predicted correspondences. This indicates that the filtering operation can remove extreme outliers that are far from 008_pudding_box to improve the pose estimation. Using M_{vis} in the filtering operations obtains better results than M_{full}, which demonstrates that the localization results of the keypoints occluded by the distraction object are not accurate enough for recovering the pose.

4. Evaluation of 2D-3D Correspondences

The evaluation results in the main paper focus on the final estimated poses. We additionally evaluate the quality of the established dense correspondences before RANSAC. Specifically, for each test sample, we reproject the 3D keypoints by the ground truth pose and compute the mean distance between the reprojection results and predicted 2D locations. For symmetric objects, we use the equivalent rotation closest to our final estimated pose. To obtain the inlier ratio of the estimated correspondences, we regard a keypoint as an inlier if its reprojection error is less than 5 pixels. We compute the average reprojection error and inlier ratio for each object and report the average values over the whole dataset in Table 7.

5. BOP Results on LM-O and YCB-V

We report the performance of our method on LM-O and YCB-Video using the evaluation metrics from BOP challenge [7] in Table 8 and Table 9, respectively. We mainly select baselines from officially published work. We also include the results of GDRNPP [17] for reference, which improves upon GDR-Net [22] with implementation skills including stronger domain randomization, more powerful detectors, *etc*., to compensate for the domain gap between training and test images. Without these implementation skills, our method still achieves comparable performance with the state-of-the-art methods, including the refinement based method [16].

Dataset	LM	LM-O	YCB-V
reprojection error (pixel)	3.4	14.4	10.9
inlier ratio (%)	88.4	67.8	39.6

Table 7: **Evaluation results of predicted dense correspondences.**

Method	AR_{MSPD}	AR_{MSSD}	AR_{VSD}	AR
SurfEmb [5]	85.1	64.0	49.7	66.3
Coupled [16]	83.1	63.3	50.1	65.5
Zebra [20]	88.0	72.1	55.2	71.8
NCF [11]	–	–	–	63.2
PFA [8]	83.7	66.1	52.3	67.4
CRT-6D [4]	83.7	64.0	50.4	66.0
GDRNPP [17]	88.7	70.1	54.9	71.3
Ours	87.3	72.3	53.7	71.1

Table 8: **Results on LM-O dataset under BOP setup [7].** The results of Coupled [16] and NCF [11] are obtained from the original paper, and the results of other methods are obtained from https://bop.felk.cvut.cz/leaderboards/. We highlight the best result in red color, and the second best result in blue color. "–" denotes unavailable results.

6. Detailed Results of YCB-V

We report the detailed evaluation metrics of each object on YCB-V dataset [24] in Table 10 and Table 11. Our method outperforms previous methods w.r.t. ADD(-S) and AUC of ADD(-S), and achieves comparable performance with state of the art w.r.t. AUC of ADD-S.

7. Qualitative Results

We provide additional qualitative results for LM-O [2] and YCB-V [24] in Figure 2 and Figure 3, respectively. We render the 3D CAD model based on the predictions of CheckerPose, and highlight the contour in green. We also highlight the ground truth contour in blue. For better visualization, we crop the images and we also show the original input image on the left for LM-O and YCB-V.

Method	AR_{MSPD}	AR_{MSSD}	AR_{VSD}	AR
SurfEmb [5]	77.3	62.0	54.8	64.7
Coupled [16]	85.2	83.5	78.3	82.4
Zebra [20]	86.4	83.0	75.1	81.5
NCF [11]	–	–	–	77.5
PFA [8]	84.9	81.4	75.8	80.7
SC6D [3]	80.4	79.6	69.5	76.5
CRT-6D [4]	77.4	77.6	70.6	75.2
GDRNPP [17]	86.9	84.6	76.0	82.5
Ours	85.3	84.4	70.7	80.1

Table 9: **Results on YCB-Video dataset under BOP setup [7].** The results of Coupled [16] and NCF [11] are obtained from the original paper, and the results of other methods are obtained from https://bop.felk.cvut.cz/leaderboards/. We highlight the best result in red color, and the second best result in blue color. "–" denotes unavailable results.

Furthermore, we provide more keypoint localization results of duck, bowl, and banana in Figure 4. For better visualization we only plot eight keypoints that are evenly distributed over the object surface. While our network directly outputs the 2D locations, the results of other dense methods [20, 22] are computed by projecting the keypoints using the estimated poses. Considering the symmetry of the bowl, we use the equivalent rotations closest to our prediction to project the keypoints of bowl.

8. Failure Cases and Future Work

We visualize typical failure cases in Figure 5. As shown in Figure 5 (a) and (b), the textureless object eggbox from LM-O dataset is severely occluded by a toy car, and a distraction object with similar color also partially appears in the input RoI. As a result, the estimated 2D projections are shifted towards the distraction object. We also present a failure case of objects with textures in Figure 5 (c) and (d). The object in interest is 002_master_chef_can from YCB-V dataset, which is geometrically symmetric. Though the texture is almost symmetric as well, the barcode only appears on one side of the object, which causes the asymmetry. For the given input RoI, the keypoints are localized in the opposite directions, w.r.t. the central axis.

To improve the localization results, one future direction is the selection of 3D keypoints. Since we adopt farthest point sampling algorithm to obtain evenly distributed keypoints, we ignore other factors to make the keypoints more representative. For example, the issue of 002_master_chef_can may be solved by sampling more keypoints in the barcode area. Besides, no positional encoding [18, 21] is leveraged in graph feature aggregation and image feature fusion operations. Such encoding can provide additional cues for textureless regions. In future, we will explore the positional encoding to enhance the keypoint localization process.

References

[1] Daniel Barath and Jiri Matas. Progressive-x: Efficient, anytime, multi-model fitting algorithm. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3780–3788, 2019. 1

[2] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6d object pose estimation using 3d object coordinates. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 536–551. Springer, 2014. 1, 2, 3

[3] Dingding Cai, Janne Heikkilä, and Esa Rahtu. SC6D: symmetry-agnostic and correspondence-free 6d object pose estimation. In *2022 International Conference on 3D Vision (3DV)*. IEEE, 2022. 4

[4] Pedro Castro and Tae-Kyun Kim. CRT-6D: fast 6d object pose estimation with cascaded refinement transformers. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 5746–5755, 2023. 3, 4

[5] Rasmus Laurvig Haugaard and Anders Glent Buch. Surfemb: Dense and continuous correspondence distributions for object pose estimation with learnt surface embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6749–6758, 2022. 3, 4

[6] 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 (ACCV)*, pages 548–562. Springer, 2012. 1

[7] Tomáš Hodaň, Martin Sundermeyer, Bertram Drost, Yann Labbé, Eric Brachmann, Frank Michel, Carsten Rother, and Jiří Matas. BOP challenge 2020 on 6D object localization. *European Conference on Computer Vision Workshops (ECCVW)*, 2020. 3, 4

[8] Yinlin Hu, Pascal Fua, and Mathieu Salzmann. Perspective flow aggregation for data-limited 6d object pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 89–106. Springer, 2022. 3, 4

[9] Yinlin Hu, Pascal Fua, Wei Wang, and Mathieu Salzmann. Single-stage 6d object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2930–2939, 2020. 5

[10] Yinlin Hu, Joachim Hugonot, Pascal Fua, and Mathieu Salzmann. Segmentation-driven 6d object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3385–3394, 2019. 5

[11] Lin Huang, Tomas Hodan, Lingni Ma, Linguang Zhang, Luan Tran, Christopher Twigg, Po-Chen Wu, Junsong Yuan,

Method	SegDriven[10]	S.Stage[9]	RePose [12]	GDR [22]	Zebra [20]	DProST [19]	**Ours**
002_master_chef_can	33.0	-	-	41.5	**62.6**	-	45.9
003_cracker_box	44.6	-	-	83.2	**98.5**	-	94.2
004_sugar_box	75.6	-	-	91.5	96.3	-	**98.3**
005_tomato_soup_can	40.8	-	-	65.9	80.5	-	**83.2**
006_mustard_bottle	70.6	-	-	90.2	**100.0**	-	99.2
007_tuna_fish_can	18.1	-	-	44.2	70.5	-	**88.9**
008_pudding_box	12.2	-	-	2.8	**99.5**	-	86.5
009_gelatin_box	59.4	-	-	61.7	**97.2**	-	86.0
010_potted_meat_can	33.3	-	-	64.9	**76.9**	-	70.0
011_banana	16.6	-	-	64.1	71.2	-	**96.0**
019_pitcher_base	90.0	-	-	99.0	**100.0**	-	**100.0**
021_bleach_cleanser	70.9	-	-	73.8	75.9	-	**89.8**
024_bowl*	30.5	-	-	37.7	18.5	-	**68.0**
025_mug	40.7	-	-	61.5	77.5	-	**89.0**
035_power_drill	63.5	-	-	78.5	**97.4**	-	95.9
036_wood_block*	27.7	-	-	59.5	**87.6**	-	58.7
037_scissors	17.1	-	-	3.9	**71.8**	-	62.4
040_large_marker	4.8	-	-	7.4	**23.3**	-	18.8
051_large_clamp*	25.6	-	-	69.8	87.6	-	**95.4**
052_extra_large_clamp*	8.8	-	-	90.0	**98.0**	-	95.6
061_foam_brick*	34.7	-	-	71.9	**99.3**	-	87.2
MEAN	39.0	53.9	62.1	60.1	80.5	65.1	**81.4**

Table 10: **Detailed results on YCB-V [24] w.r.t. ADD(-S).** (*) denotes symmetric objects and "-" denotes unavailable results.

Cem Keskin, and Robert Wang. Neural correspondence field for object pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 585–603. Springer, 2022. 3, 4

[12] 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 (ICCV)*, pages 3303–3312, 2021. 5

[13] Sagi Katz, Ayellet Tal, and Ronen Basri. Direct visibility of point sets. *ACM Transactions On Graphics (TOG)*, 26(3):24, 2007. 1

[14] Yann Labbé, Justin Carpentier, Mathieu Aubry, and Josef Sivic. CosyPose: consistent multi-view multi-object 6d pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 574–591. Springer, 2020. 6

[15] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision*, 81(2):155, 2009. 1

[16] Lahav Lipson, Zachary Teed, Ankit Goyal, and Jia Deng. Coupled iterative refinement for 6d multi-object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6728–6737, 2022. 3, 4

[17] Xingyu Liu, Ruida Zhang, Chenyangguang Zhang, Bowen Fu, Jiwen Tang, Xiquan Liang, Jingyi Tang, Xiaotian Cheng, Yukang Zhang, Gu Wang, and Xiangyang Ji. GDRNPP. https://github.com/shanice-l/gdrnpp_bop2022, 2022. 3, 4

[18] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 4

[19] Jaewoo Park and Nam Ik Cho. DProST: 6-dof object pose estimation using space carving and dynamic projective spatial transformer. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 5, 6

[20] Yongzhi Su, Mahdi Saleh, Torben Fetzer, Jason Rambach, Nassir Navab, Benjamin Busam, Didier Stricker, and Federico Tombari. ZebraPose: coarse to fine surface encoding for 6dof object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6738–6748, 2022. 3, 4, 5, 6, 9

[21] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 2020. 4

[22] Gu Wang, Fabian Manhardt, Federico Tombari, and Xiangyang Ji. GDR-Net: geometry-guided direct regression network for monocular 6d object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision*

Method	CosyPose [14]		GDR-Net[22]		ZebraPose[20]		DProST [19]	**Ours**	
Metric	AUC of ADD-S	AUC of ADD(-S)	AUC of ADD-S	AUC of ADD(-S)	AUC of ADD-S	AUC of ADD(-S)	AUC of ADD(-S)	AUC of ADD-S	AUC of ADD(-S)
002_master_chef_can	-	-	96.3	65.2	93.7	75.4	-	87.5	67.7
003_cracker_box	-	-	97.0	88.8	93.0	87.8	-	93.2	86.7
004_sugar_box	-	-	98.9	95.0	95.1	90.9	-	95.9	91.7
005_tomato_soup_can	-	-	96.5	91.9	94.4	90.1	-	94.0	89.9
006_mustard_bottle	-	-	100.0	92.8	96.0	92.6	-	95.7	90.9
007_tuna_fish_can	-	-	99.4	94.2	96.9	92.6	-	97.5	94.4
008_pudding_box	-	-	64.6	44.7	97.2	95.3	-	94.9	91.5
009_gelatin_box	-	-	97.1	92.5	96.8	94.8	-	96.1	93.4
010_potted_meat_can	-	-	86.0	80.2	91.7	83.6	-	86.4	80.4
011_banana	-	-	96.3	85.8	92.6	84.6	-	95.7	90.1
019_pitcher_base	-	-	99.9	98.5	96.4	93.4	-	95.8	91.9
021_bleach_cleanser	-	-	94.2	84.3	89.5	80.0	-	90.6	83.2
024_bowl*	-	-	85.7	85.7	37.1	37.1	-	82.5	82.5
025_mug	-	-	99.6	94.0	96.1	90.8	-	96.9	92.7
035_power_drill	-	-	97.5	90.1	95.0	89.7	-	94.7	88.8
036_wood_block*	-	-	82.5	82.5	84.5	84.5	-	68.3	68.3
037_scissors	-	-	63.8	49.5	92.5	84.5	-	91.7	81.6
040_large_marker	-	-	88.0	76.1	80.4	69.5	-	83.3	72.3
051_large_clamp*	-	-	89.3	89.3	85.6	85.6	-	90.0	90.0
052_extra_large_clamp*	-	-	93.5	93.5	92.5	92.5	-	91.6	91.6
061_foam_brick*	-	-	96.9	96.9	95.3	95.3	-	94.1	94.1
MEAN	89.8	84.5	91.6	84.3	90.1	85.3	77.4	91.3	86.4

Table 11: **Detailed results on YCB-V [24] w.r.t. AUC of ADD-S and ADD(-S).** As in [24], symmetric metric is used for all objects in ADD-S while only for symmetric objects in ADD(-S). (*) denotes symmetric objects.

and Pattern Recognition (CVPR), pages 16611–16621, 2021. 3, 4, 5, 6, 9

[23] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions On Graphics (TOG)*, 38(5):1–12, 2019. 1

[24] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. PoseCNN: a convolutional neural network for 6d object pose estimation in cluttered scenes. In *Robotics: Science and Systems*, 2018. 1, 2, 3, 5, 6

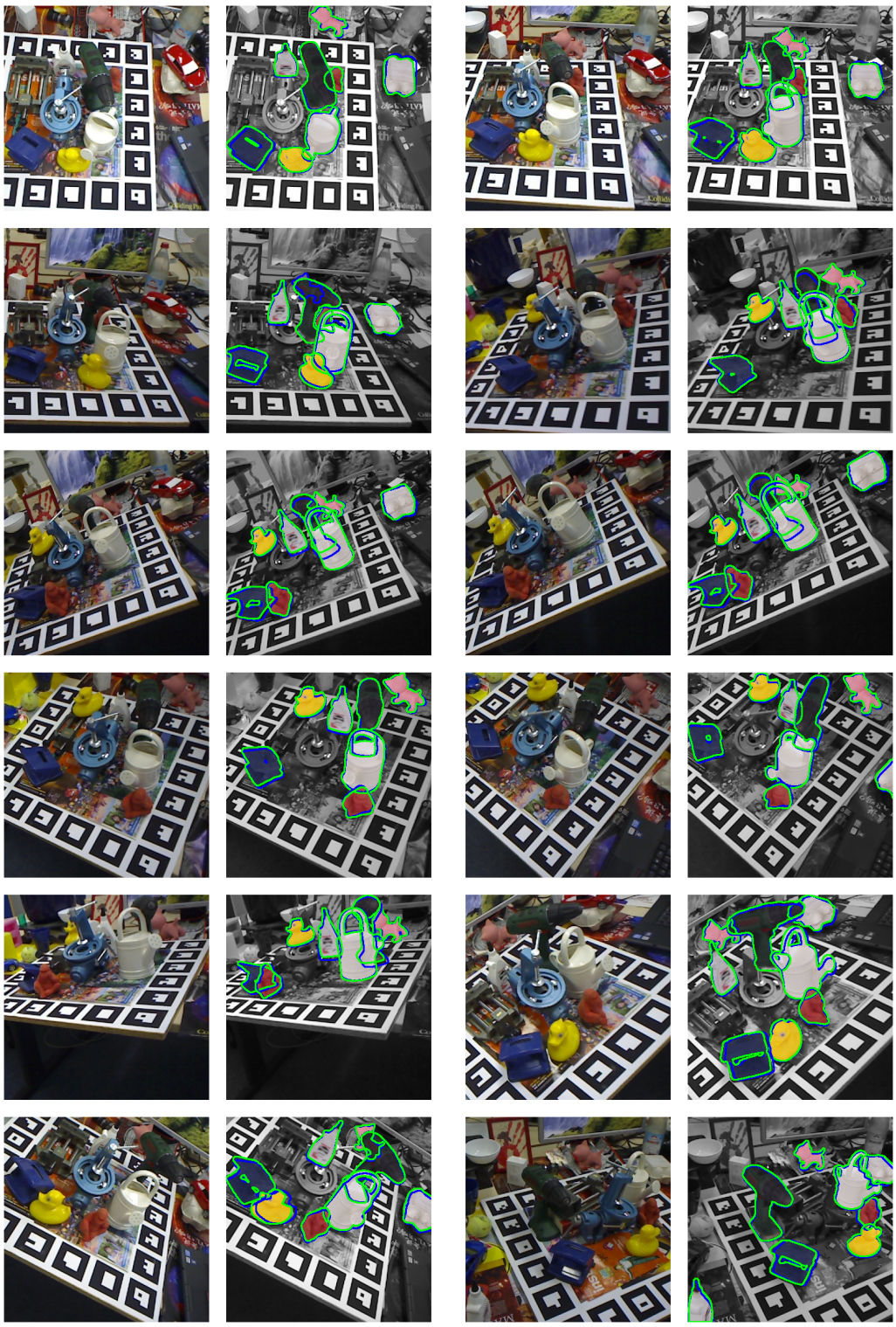


Figure 2: **Qualitative results on the LM-O dataset.** For each image on the left, we visualize the 6D pose by rendering the 3D CAD models and highlighting the contours on the right. Blue color denotes ground truth and green color denotes the prediction from CheckerPose.

Figure 3: **Qualitative results on the YCB-V dataset.** For each image on the left, we visualize the 6D pose by rendering the 3D CAD models and highlighting the contours on the right. Blue color denotes ground truth and green color denotes the prediction from CheckerPose.

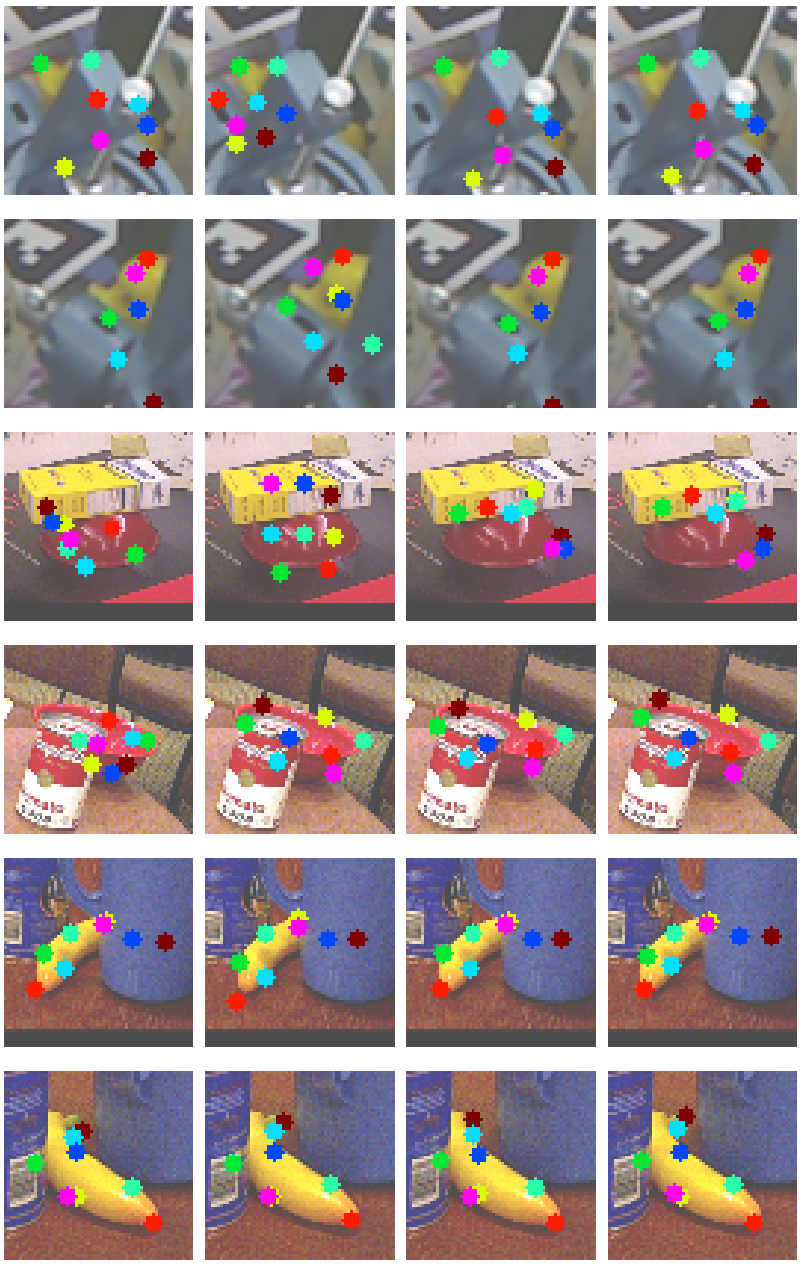
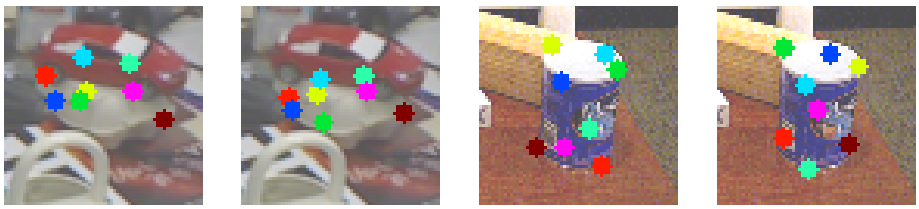


Figure 4: **Visualization of keypoint localization.** Each column visualizes the keypoint location results of ZebraPose [20], GDR-Net [22], our method, and ground truth. While our network directly outputs the 2D locations, the results of other dense methods [20, 22] are computed by projecting the keypoints using the estimated poses.



(a) Ground Truth (b) Prediction (c) Ground Truth (d) Prediction

Figure 5: **Failure cases.** We provide the localization results of eight keypoints that are inliers of the estimated poses.