Supplementary Material for "DualPoseNet: Category-level 6D Object Pose and Size Estimation using Dual Pose Network with Refined Learning of Pose Consistency"

Jiehong Lin, Zewei Wei, Zhihao Li, Songcen Xu, Kui Jia*, Yuanqing Li

A. More Implementation Details

A.1. Implementation Details of Instance-level 6D Pose Estimation

In Sec. 5.2, we apply our proposed DualPoseNet to instance-level 6D pose estimation and achieve remarkable results on the benchmark YCB-Video [1] and LineMOD [3] datasets. Here we elaborate on the adaptive modifications of DualPoseNet to the instance-level task.

Firstly, we modify the framework in Fig. 1 and the objectives in Sec. 4.5. For Ψ_{exp}, we remove the estimation of size and revise the training objective in Eq. (5) as follows:

$$\mathcal{L}_{\Phi,\Psi_{exp}} = \frac{1}{M}\sum_{i=1}^{M} \left\| \boldsymbol{R}^T(\boldsymbol{R}^*\boldsymbol{y}_i + \boldsymbol{t}^* - \boldsymbol{t}) - \boldsymbol{y}_i \right\|_2, \quad (9)$$

where $\mathcal{Y} = \{\boldsymbol{y}_i\}_{i=1}^M$ denotes the sampled point set of object CAD model with a total of M points. For Ψ_{im}, the transformed point set $\{\boldsymbol{R}^*\boldsymbol{y}_i + \boldsymbol{t}^*\}_{i=1}^M$ is used as input to replace the observed $\mathcal{P}$, and the objective in Eq. (6) is modified as follows:

$$\mathcal{L}_{\Phi,\Psi_{im}} = \frac{1}{M}\sum_{i=1}^{M} \|\boldsymbol{q}_i - \boldsymbol{y}_i\|_2 . \quad (10)$$

The above objectives are well-defined for asymmetric objects, but not suitable for symmetric ones. Following [9], we alternatively use Chamfer distance to define the objectives of symmetric objects as follows:

$$\mathcal{L}_{\Phi,\Psi_{exp}} = \frac{1}{M}\sum_{i=1}^{M} \min_{0<j\le M} \left\| \boldsymbol{R}^T(\boldsymbol{R}^*\boldsymbol{y}_i + \boldsymbol{t}^* - \boldsymbol{t}) - \boldsymbol{y}_j \right\|_2,$$
$$\mathcal{L}_{\Phi,\Psi_{im}} = \frac{1}{M}\sum_{i=1}^{M} \min_{0<j\le M} \|\boldsymbol{q}_i - \boldsymbol{y}_j\|_2 . \quad (11)$$

Combining $\mathcal{L}_{\Phi,\Psi_{exp}}$ and $\mathcal{L}_{\Phi,\Psi_{im}}$ results in the following optimization problem:

$$\min_{\Phi,\Psi_{exp},\Psi_{im}} \mathcal{L}_{\Phi,\Psi_{exp}} + \lambda\mathcal{L}_{\Phi,\Psi_{im}}, \quad (12)$$

where the penalty parameter λ is set as 1.

Secondly, we modify the objective of the refined learning in Sec.4.6, since CAD model of each object is available. Specifically, we make a pose refinement by simultaneously enforcing the pose consistencies of both pose decoders against the CAD model, and thus revise the objective of the refined learning in Eq. (8) as follows:

$$\min_{\Phi} \mathcal{L}_{\Phi}^{Refine} =$$
$$\frac{1}{N}\sum_{i=1}^{N} \min_{0<j\le M} \left\| \boldsymbol{R}^\top(\boldsymbol{p}_i - \boldsymbol{t}) - \boldsymbol{y}_j \right\|_2 + \min_{0<j\le M} \|\boldsymbol{q}_i - \boldsymbol{y}_j\|_2, \quad (13)$$

where the input point set of Ψ_{im} is the observed $\mathcal{P}$, and the predicted $\mathcal{Q}$ is the canonical version of $\mathcal{P}$.

Thirdly, we augment our DualPoseNet with a 2nd-stage module Ω for iterative refinement of residual pose, which follows [9, 13]. Given the observed $\mathcal{P}$ and the initially predicted $(\boldsymbol{R}_0, \boldsymbol{t}_0)$, Ω is proposed to correct the pose estimation error by learning the residual pose:

$$(\Delta\boldsymbol{R}, \Delta\boldsymbol{t}) = \Omega(\mathcal{P}, \boldsymbol{R}_0, \boldsymbol{t}_0), \quad (14)$$

and the refined pose is given as the concatenation of the residual and initial poses:

$$(\boldsymbol{R}, \boldsymbol{t}) = [\Delta\boldsymbol{R}|\Delta\boldsymbol{t}] \cdot [\boldsymbol{R}_0|\boldsymbol{t}_0]. \quad (15)$$

We construct Ω based on a small PointNet [5], as shown in Fig. 6, where the network specifics are given. During testing, Ω is applied in an iterative manner to refine the pose, *e.g.*, with K iterations, the resulting pose is obtained as:

$$(\boldsymbol{R}, \boldsymbol{t}) = [\Delta\boldsymbol{R}_K|\Delta\boldsymbol{t}_K] \cdots [\Delta\boldsymbol{R}_1|\Delta\boldsymbol{t}_1] \cdot [\boldsymbol{R}_0|\boldsymbol{t}_0], \quad (16)$$

where $(\Delta\boldsymbol{R}_k, \Delta\boldsymbol{t}_k)$ is the predicted residual pose of iteration k, and $(\boldsymbol{R}_0, \boldsymbol{t}_0)$ is given by the refined learning of pose consistency of the previous stage. In our experiments, K is set as 2, following [9, 13]. We note that different from [9, 13], we do not consider RGB embeddings in Ω for simplicity.

*Corresponding author

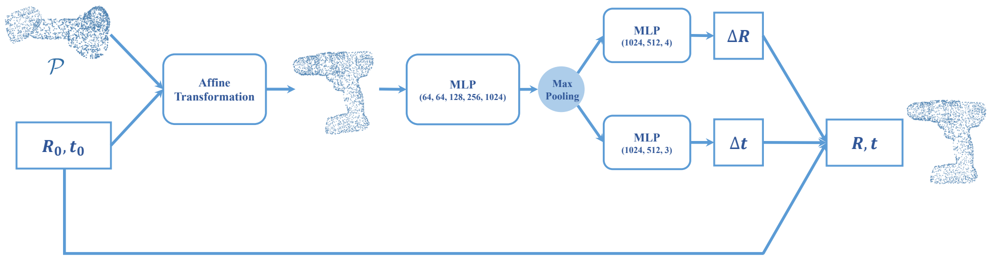


Figure 6. An illustration of the augmented 2nd-stage module Ω for iterative refinement of residual pose.

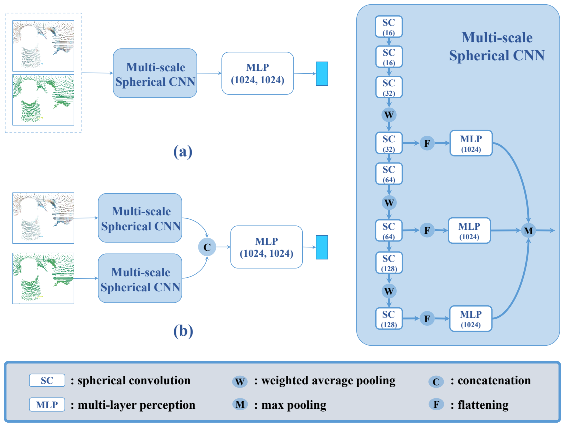

Multi-scale Spherical CNN
MLP (1024, 1024)
(a)
Multi-scale Spherical CNN
Multi-scale Spherical CNN
C
MLP (1024, 1024)
(b)
Multi-scale Spherical CNN
SC (16)
SC (16)
SC (32)
W
SC (32)
F
MLP (1024)
SC (64)
W
SC (64)
F
MLP (1024)
SC (128)
W
SC (128)
F
MLP (1024)
M
SC : spherical convolution
W : weighted average pooling
C : concatenation
MLP : multi-layer perception
M : max pooling
F : flattening

Figure 7. Architectures of the pose decoders (a) **SCNN-EarlyFusion** and (b) **SCNN-LateFusion**.

A.2. Architectures of SCNN-EarlyFusion and SCNN-LateFusion

In Sec. 5.1.1, to evaluate the efficacy of our proposed spherical fusion based encoder Φ, we build alternative encoders, **SCNN-EarlyFusion** (*c.f.* Fig. 7(a)) and **SCNN-LateFusion** (*c.f.* Fig. 7(b)), based on a multi-scale spherical CNN. For the sake of fairness, the used multi-scale spherical CNN is constructed, similar to Φ, by stacking 8 spherical convolution layers with aggregation of multi-scale spherical features. Fig. 7 gives the illustration, where layer specifics are also given.

B. More Quantitative Results

B.1. Quantitative Results of the Implicit Pose Decoder Ψ_{im}

As illustrated in Sec 4.1, there exist (at least) three ways to obtain the pose predictions from DualPoseNet, and we have shown the results obtained by the first and third ways in Table 1. Here we supplement the results of the second way on REAL275 dataset [10] in Table 5. Following [10, 7], we predict the 6D pose (rotation $\boldsymbol{R}$ and translation $\boldsymbol{t}$) and the 1-DoF scaling by solving Umeyama algorithm [8] with the observed $\mathcal{P}$ and the predicted $\mathcal{Q}$ in its canonical pose, and the multiplication of the 1-DoF scaling and the 3D extension of $\mathcal{Q}$ gives the estimated size of $\mathcal{P}$. Due to the incompleteness of $\mathcal{Q}$ (as well as $\mathcal{P}$), the size estimated by Ψ_{im} in this way is less precise than that directly regressed by Ψ_{exp}; thus performances on metrics related to IoU in Table 5, which are greatly influenced by the precision of size predictions, are obviously worse than those in Table 1. However, in terms of 6D pose precision, *e.g.*, on the metric of $m^{\circ}ncm$, the results obtained by both decoders are of similar quality.

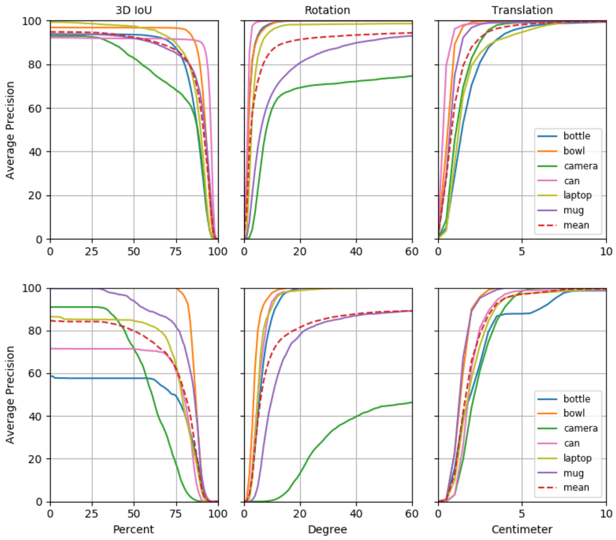


Figure 8. Plottings of per-category average precision versus different thresholds on 3D IoU, rotation error, and translation error, tested on CAMERA25 (top row) and REAL275 (bottom row) datasets [10].

B.2. Per-Category Performance on CAMERA25 and REAL275

Fig. 8 shows the average precision versus different thresholds for all 6 categories on both CAMERA25 and REAL275 datasets; it also provides independent evaluations on 3D IoU, rotation error, and translation error.

B.3. Per-Instance Performance on YCB-Video and LineMOD

We compare per-instance quantitative results of different methods in Table 6 and Table 7, separately.

	mAP											
	IoU_{75} 5°, 5%	IoU_{75} 10°, 5%	IoU_{75} 5°, 10%	IoU_{50} 5°, 20%	IoU_{50} 10°, 10%	IoU_{50} 10°, 20%	IoU_{50}	IoU_{75}	5° 2cm	5° 5cm	10° 2cm	10° 5cm
W/o refining	4.7	7.2	13.7	29.0	44.0	54.0	79.8	35.8	28.5	34.7	49.2	65.8
With refining	5.1	7.4	14.0	30.0	44.7	54.8	77.8	35.1	29.3	35.9	50.1	66.7

Table 5. Quantitative evaluation of the Implicit Decoder Ψ_{im} on REAL275. Evaluations are based on both our proposed metrics (left) and the metrics (right) proposed in [10].

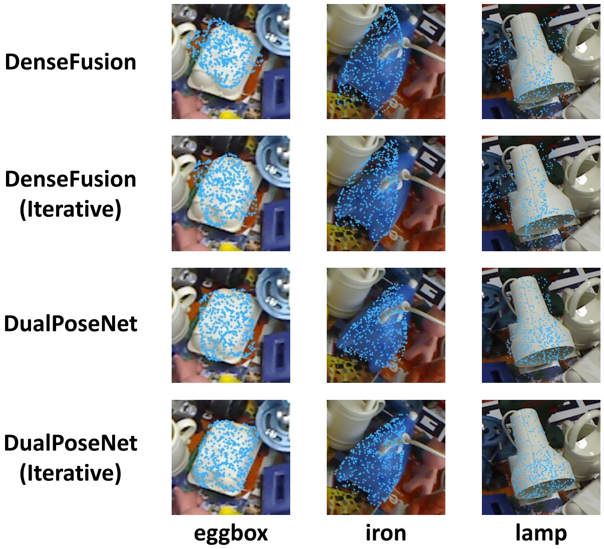


Figure 9. Qualitative results of DenseFusion [9] and DualPoseNet, with or without iterative pose refinement, on the LineMOD dataset [3]. The sampled points (in blue) of object models are transformed by the predicted pose and projected back to 2D images.

C. More Qualitative Results

For category-level 6D pose and size estimation, we visualize more qualitative results of different methods on CAMERA25 and REAL275 [10] in Fig. 11 (a) and (b), separately. For instance-level 6D pose estimation, we show qualitative comparisons between classical DenseFusion [9] and our DualPoseNet on YCB-Video (*c.f.* Fig. 10) and LineMOD (*c.f.* Fig. 9) datasets.

References

[1] Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron 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. 1, 4

[2] Yisheng He, Wei Sun, Haibin Huang, Jianran Liu, Haoqiang Fan, and Jian Sun. Pvn3d: A deep point-wise 3d keypoints voting network for 6dof pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11632–11641, 2020. 4

[3] 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. 1, 3, 4

[4] Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab. Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. In *Proceedings of the IEEE international conference on computer vision*, pages 1521–1529, 2017. 4

[5] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 1

[6] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. Implicit 3d orientation learning for 6d object detection from rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 699–715, 2018. 4

[7] Meng Tian, Marcelo H Ang Jr, and Gim Hee Lee. Shape prior deformation for categorical 6d object pose and size estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, August 2020. 2

[8] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (4):376–380, 1991. 2

[9] Chen Wang, Danfei Xu, Yuke Zhu, Roberto Martín-Martín, Cewu Lu, Li Fei-Fei, and Silvio Savarese. Densefusion: 6d object pose estimation by iterative dense fusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3343–3352, 2019. 1, 3, 4

[10] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2642–2651, 2019. 2, 3, 5

[11] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. 2018. 4

[12] Danfei Xu, Dragomir Anguelov, and Ashesh Jain. Pointfusion: Deep sensor fusion for 3d bounding box estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 244–253, 2018. 4

[13] Zelin Xu, Ke Chen, and Kui Jia. W-posenet: Dense correspondence regularized pixel pair pose regression. *arXiv preprint arXiv:1912.11888*, 2019. 1, 4

	PointFusion [12]	PoseCNN + ICP[11]	Densefusion [9]	Densefusion (Iterative)[9]	$\mathcal{W}$-PoseNet [13]	$\mathcal{W}$-PoseNet (Iterative)[13]	PVN3D [2]	PVN3D + ICP [2]	DualPoseNet	DualPoseNet (Iterative)
002_master_chef_can	90.9	95.8	95.2	96.4	96.0	96.0	96.0	95.2	96.6	**97.9**
003_cracker_box	80.5	92.7	92.5	95.5	93.0	95.5	96.1	94.4	92.8	**99.2**
004_sugar_box	90.4	98.2	95.1	97.5	96.7	97.8	97.4	97.9	98.1	**99.6**
005_tomato_soup_can	91.9	94.5	93.7	94.6	94.3	94.5	**96.2**	95.9	95.0	95.5
006_mustard_bottle	88.5	98.6	95.9	97.2	97.3	98.1	97.5	98.3	98.0	**99.5**
007_tuna_fish_can	93.8	97.1	94.9	96.6	96.5	97.3	96.0	96.7	95.6	**98.1**
008_pudding_box	87.5	97.9	94.7	96.5	95.1	96.6	97.1	98.2	97.1	**99.6**
009_gelatin_box	95.0	98.8	95.8	98.1	96.9	98.5	97.7	98.8	98.7	**99.6**
010_potted_meat_can	86.4	92.7	90.1	91.3	90.8	91.6	93.3	93.8	92.0	**96.7**
011_banana	84.7	97.1	91.5	96.6	95.8	97.2	96.6	98.2	95.7	**98.4**
019_pitcher_base	85.5	97.8	94.6	97.1	96.3	98.3	97.4	97.6	97.0	**99.7**
021_bleach_cleanser	81.0	96.9	94.3	95.8	95.2	96.3	96.0	97.2	97.0	**99.5**
024_bowl	75.7	81.0	86.6	88.2	94.3	**96.2**	90.2	92.8	90.1	90.2
025_mug	94.2	95.0	95.5	97.1	96.5	97.1	97.6	97.7	97.3	**98.6**
035_power_drill	71.5	98.2	92.4	96.0	95.8	97.4	96.7	97.1	97.0	**99.6**
036_wood_block	68.1	87.6	85.5	89.7	91.5	91.7	90.4	91.1	96.1	**96.6**
037_scissors	76.7	91.7	96.4	95.2	88.0	89.7	**96.7**	95.0	83.3	95.1
040_large_marker	87.9	97.2	94.7	97.5	97.1	97.5	96.7	98.1	97.7	**99.2**
051_large_clamp	65.9	75.2	71.6	72.9	75.7	76.1	93.6	**95.6**	78.5	86.8
052_extra_large_clamp	60.4	64.4	69.0	69.8	73.3	74.6	88.4	**90.5**	72.0	81.6
061_foam_brick	91.8	97.2	92.4	92.5	95.8	96.9	96.8	**98.2**	94.3	96.9
MEAN	83.9	93.0	91.2	93.1	93.0	94.0	95.4	96.1	93.3	**96.5**

Table 6. Quantitative evaluation of instance-level 6D pose (ADD-S) on YCB-Video dataset [1]. Objects with bold name are symmetric.

	Implicit +ICP[6]	SSD6D +ICP[4]	PointFusion [12]	Densefusion [9]	Densefusion (Iterative)[9]	$\mathcal{W}$-PoseNet [13]	$\mathcal{W}$-PoseNet (Iterative)[13]	DualPoseNet	DualPoseNet (Iterative)
ape	20.6	65	70.4	79.5	92.3	91.7	94.9	89.6	**96.6**
bench vise	64.3	80	80.7	84.2	93.2	98.8	**98.9**	94.2	97.1
camera	63.2	78	60.8	76.5	94.4	98.4	**99.1**	90.7	97.8
can	76.1	86	61.1	86.6	93.1	96.5	**97.8**	92.8	97.0
cat	72.0	70	79.1	88.8	96.5	97.7	98.8	98.1	**99.6**
driller	41.6	73	47.3	77.7	87.0	96.3	97.1	96.3	**98.8**
duck	32.4	66	63.0	76.3	92.3	95.0	**97.7**	87.7	97.1
eggbox	98.6	100	99.9	99.9	99.8	99.8	99.8	**100.0**	**100.0**
glue	96.4	100	99.3	99.4	**100.0**	99.9	**100.0**	97.4	98.5
hole punch	49.9	49	71.8	79.0	92.1	94.4	96.5	94.4	**98.8**
iron	63.1	78	83.2	92.1	97.0	97.7	97.9	97.2	**98.9**
lamp	91.7	73	62.3	92.3	95.3	**99.6**	99.3	98.0	98.7
phone	71.0	79	78.8	88.0	92.8	96.8	97.6	94.5	**98.5**
MEAN	64.7	79	73.7	86.2	94.3	97.2	98.1	94.6	**98.2**

Table 7. Quantitative evaluation of instance-level 6D pose (ADD-S) on LineMOD dataset [3]. Objects with bold name are symmetric.

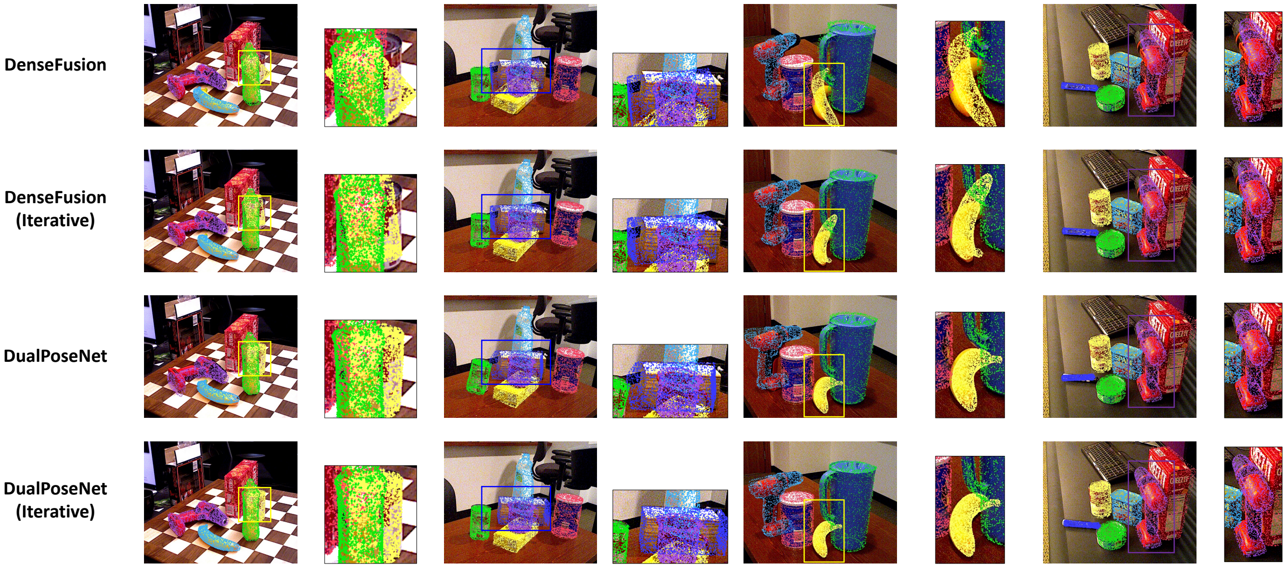
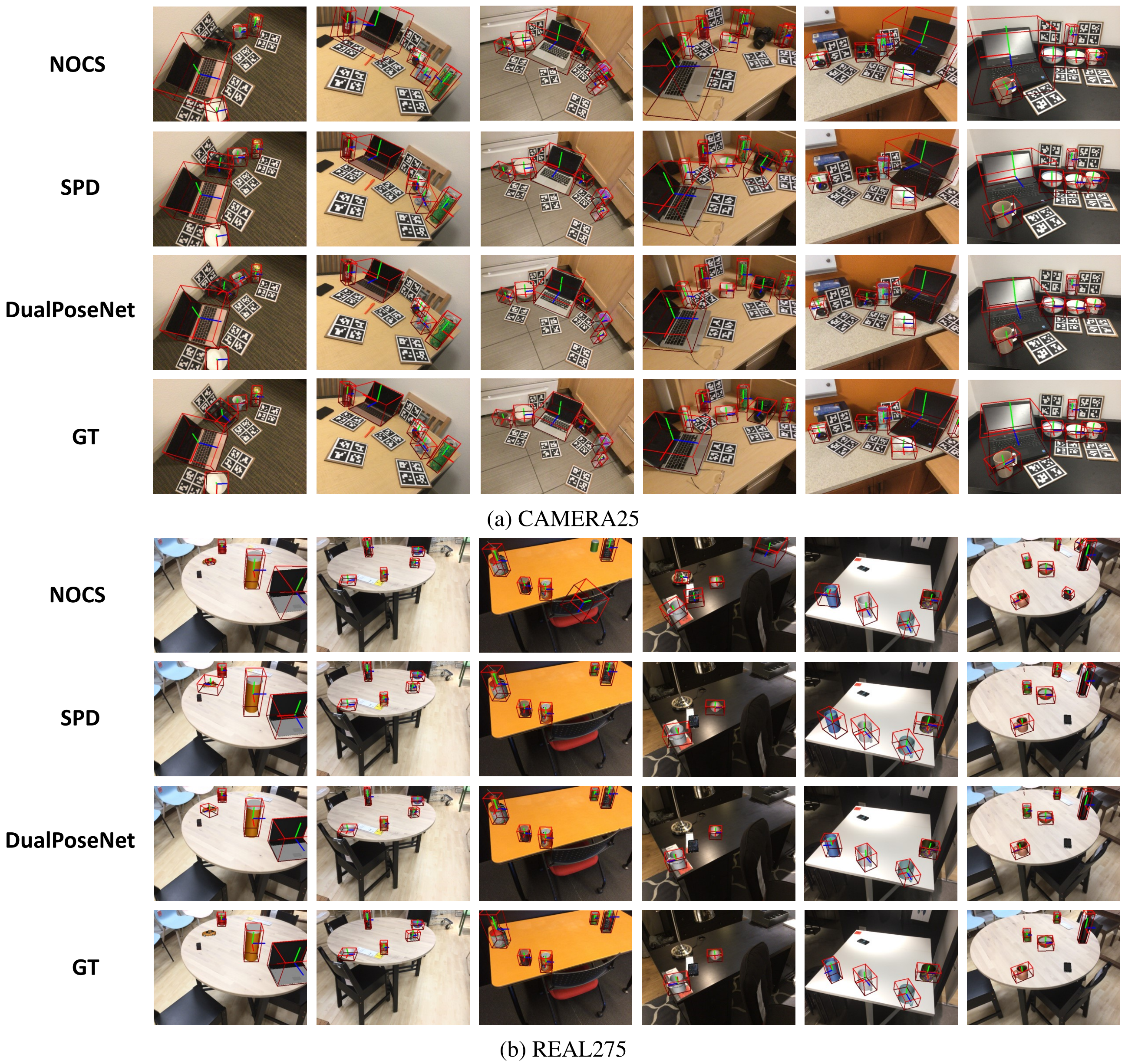


Figure 10. Qualitative results of DenseFusion [9] and DualPoseNet, with or without iterative pose refinement, on the YCB-Video dataset [1]. The sampled points of object models are transformed by the predicted pose and projected back to 2D images. Different model points in the same scene are in different colors.



(a) CAMERA25

(b) REAL275

Figure 11. Qualitative results of different methods on CAMERA25 and REAL275 datasets [10] for category-level 6D pose and size estimation.