Explaining the Ambiguity of Object Detection and 6D Pose From Visual Data Supplementary Material

Abstract

This document supplements our main paper entitled Explaining the Ambiguity of Object Detection and 6D Pose From Visual Data by providing 1. details on the used datasets, 2. quantitative results for the quality of ambiguity prediction, 3. further quantitative and qualitative evaluations for the tasks of both unambiguous (single-hypothesis) pose estimation, and ambiguity characterization, 4. examples of confidence estimation, 5. Implementation details and pseudocode of our inference and finally 6. representative images and details of the synthetic training dataset we used to train our networks.

(a) 'ape' (b) 'bvise' (c) 'cam' (d) 'can' (e) 'driller' (f) 'duck' (g) 'holep' (h) 'iron' (i) 'phone'

(a) 'eggb' (b) 'glue' (c) 'obj_04' (d) 'obj_05' (e) 'obj_09' (f) 'obj_10'

Figure 1: Top: 3D Models of the *unambiguous* Dataset from 'LineMOD' [2]. Bottom: 3D Models of the *ambiguous* Datasets from 'LineMOD [2] (first two) and T-Less [3] (last four).

1. Datasets

In Fig 1 we would like to demonstrate all the objects we employed for our experiments. Thereby, the upper row illustrates all objects of the *unambiguous* dataset, taken from 'LineMOD' [2]. These objects do not exhibit any views which might induce ambiguities. On the contrary, the lower row depicts all objects of the *ambiguous* dataset. While the first two objects also belong to the 'LineMOD' dataset, the last four accompany the T-LESS dataset [3]. All these objects can induce ambiguities for certain viewpoints. For instance 'obj 04' is a symmetric screw, however, possessing distinct textures on its head. Due to this only the views from the bottom (which do not show the texture) are ambiguous. In contrast, for each viewpoint in 'obj 09' and 'obj 10', there exists always one identical viewpoint on the other side. Thus, these objects are never ambiguity-free.

2. Robust Ambiguity Detection and Estimation

	ape	bvise	cam	can	cat	driller	duck	holep	iron	phone
Ambiguity Detection Accuracy [%]	99.9	99.9	99.9	99.8	99.7	99.7	99.6	99.1	100	100

	'eggb'	'glue'	'obj 04'	'obj 05'	'obj 09'	'obj 10'
Ambiguity Detection Accuracy [%]	50.4	86.6	90.3	94.3	100	71.2
Mean Symmetry Axis Deviation [°]	8.23	28.0	21.3	22.1	38.9	21.3
Meanshift Bin Size	$\frac{\pi}{4}$	$\frac{\pi}{4}$	$\frac{\pi}{2}$	$\frac{\pi}{5}$	$\frac{\pi}{8}$	$\frac{\pi}{10}$

Table 1: Top: Individual ambiguity detection accuracies for the *unambiguous* dataset. Bottom: Individual ambiguity detection accuracies and mean axis deviations for the *ambiguous* dataset.

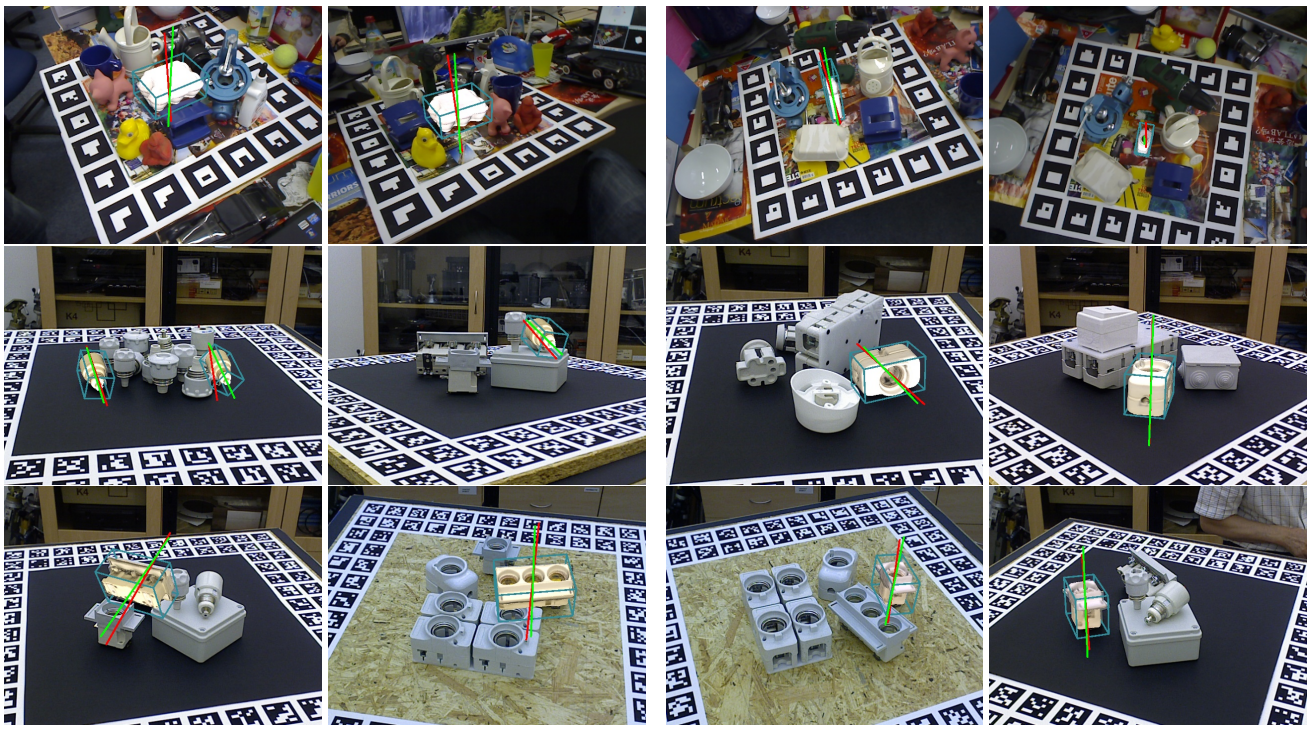


Figure 2: Qualitative samples for ambiguity detection and ambiguity axis estimation. The green line illustrates the computed axis and the red axis depicts the ground truth axis.

Tab.1 shows our detailed ambiguity detection results for the unambiguous (top) and ambiguous (bottom) objects, respectively. In addition, we also report our individual results for the ambiguity axis estimation. We compute the mean deviation from the labeled ground truth. As a threshold for σ_1 we empirically find 0.8 to offer good accuracy. Fig. 2 demonstrates more qualitative results for ambiguity detection and the computation of the corresponding ambiguity axis.

3. 2D Object Detection and 6D Pose Estimation

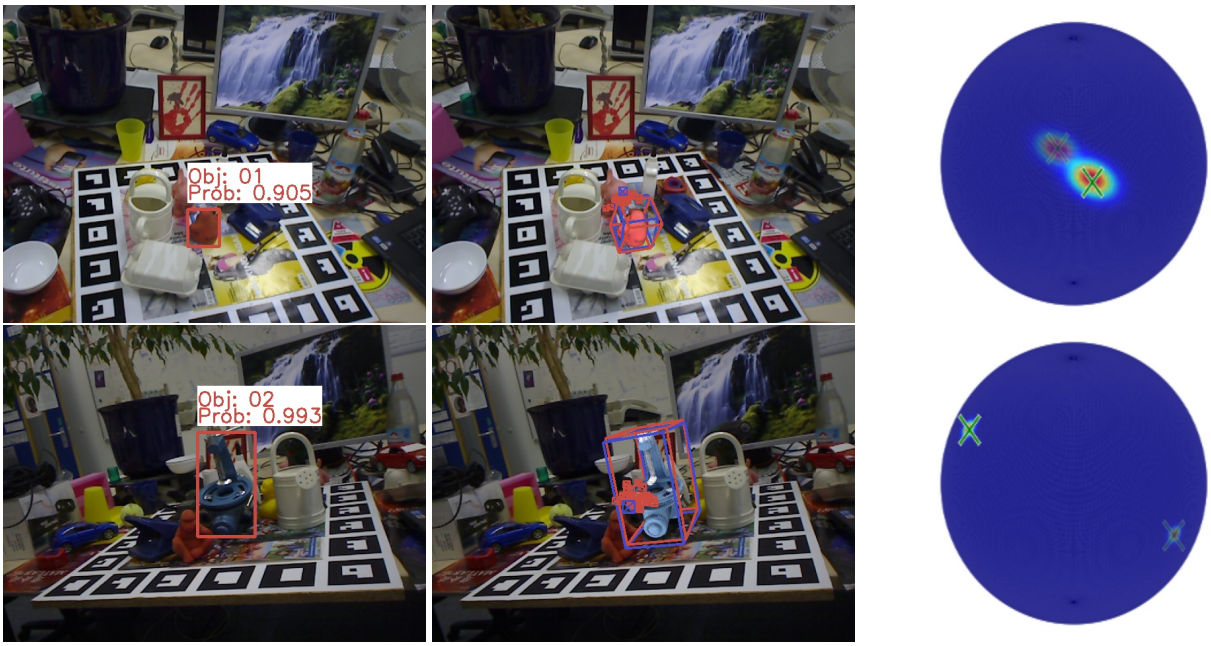
In this section, we present our detailed results for 6D pose estimation and 2D detection. As in the paper, for the *unambiguous* dataset we present our numbers with $M = 5$ and for the *ambiguous* dataset we set $M = 30$.

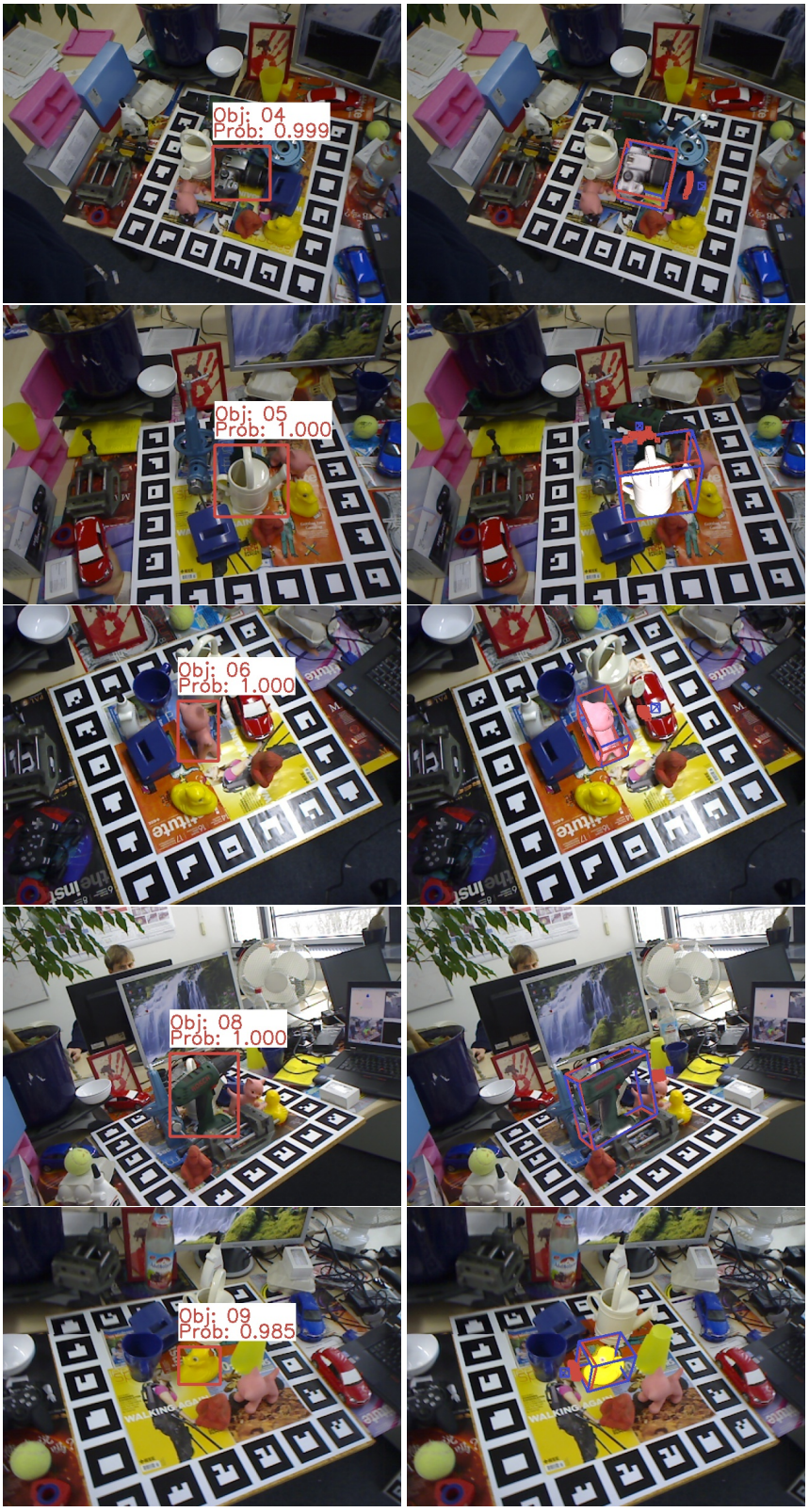
3.1. Unambiguous Object Detection and Pose Estimation

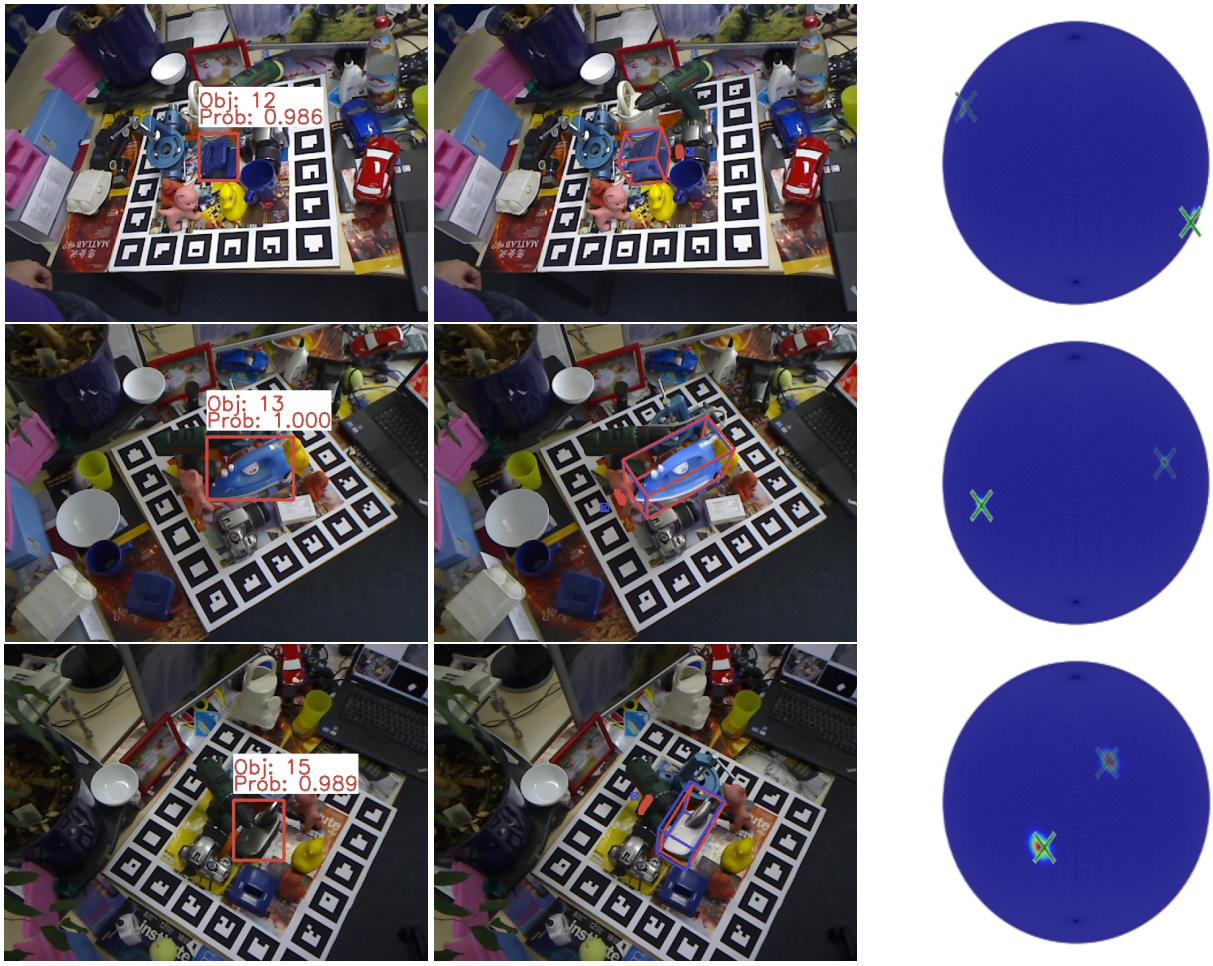
We present an ablation study for different numbers of hypotheses $\mathbf{M}$ in Tab. 2. We obtained our best results employing $\mathbf{M} = 5$ hypotheses. Below we show one qualitative sample for each object. In addition, on the right we also visualize the corresponding Bingham Distributions for visual validation. Lastly, we depict some qualitative results on the 'LineMOD Occlusion' dataset.

	Rot. [°]	Trans. [mm]	VSS [%]	ADD [%]	F1
$\mathbf{M} = 1$	17.9	45.6	76.8	31.2	91.6
$\mathbf{M} = 2$	18.9	44.3	76.3	32.8	92.1
$\mathbf{M} = 5$	**17.4**	**39.5**	**78.2**	**35.3**	**93.4**
$\mathbf{M} = 10$	19.2	45.6	77.2	31.3	90.6
$\mathbf{M} = 20$	18.7	44.6	77.4	33.8	92.7
$\mathbf{M} = 30$	19.2	44.9	77.3	32.7	91.0
$\mathbf{M} = 40$	22.5	42.6	77.4	35.7	91.0

Table 2: Ablation study on the impact of different number of hypotheses.




Obj: 04
Prob: 0.999
Obj: 05
Prob: 1.000
Obj: 06
Prob: 1.000
Obj: 08
Prob: 1.000
Obj: 09
Prob: 0.985



(a) 2D Detections (b) 6D Pose and Associated Hypotheses (c) Bingham Distributions

Figure 3: Qualitative results for the unambiguous objects.

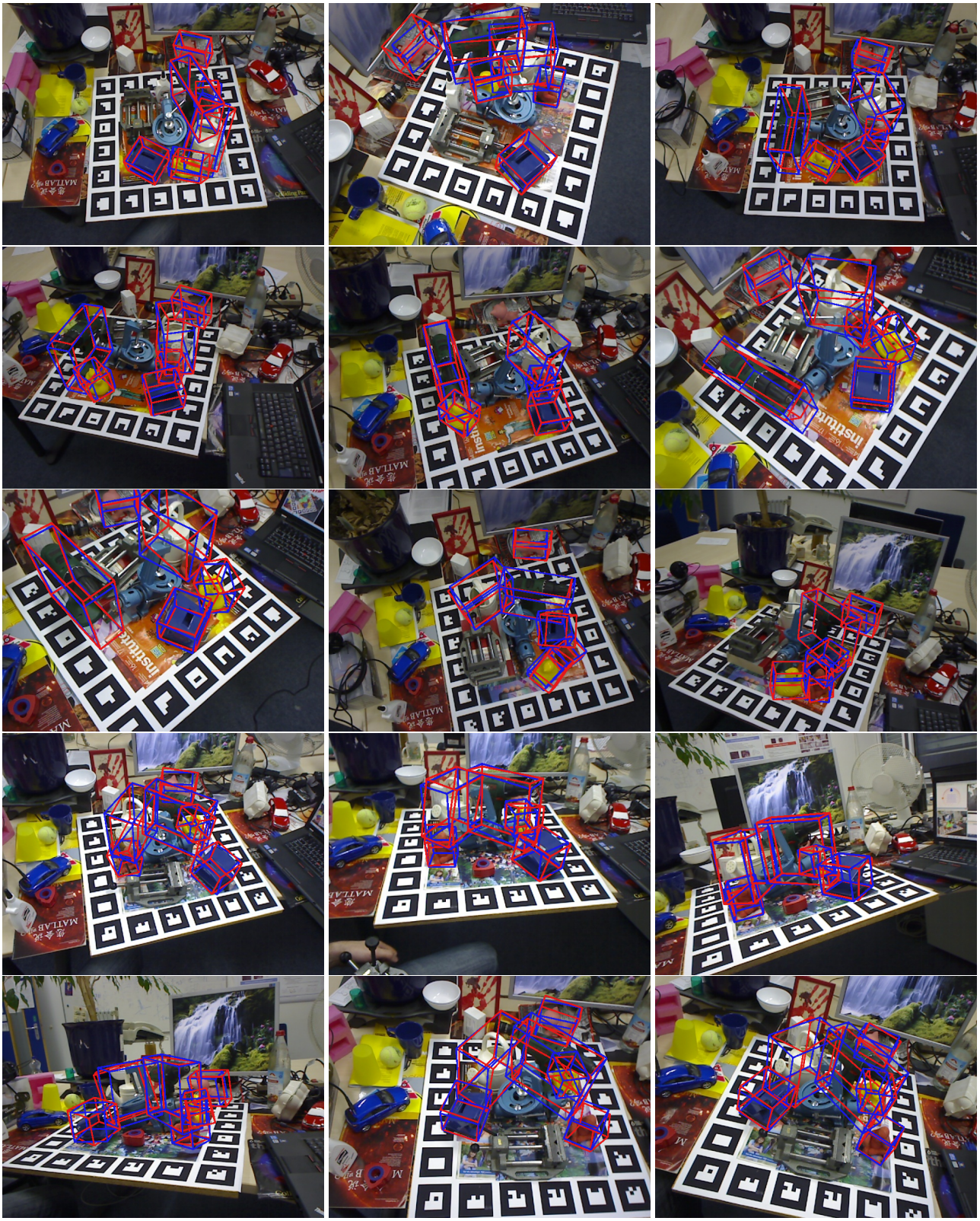


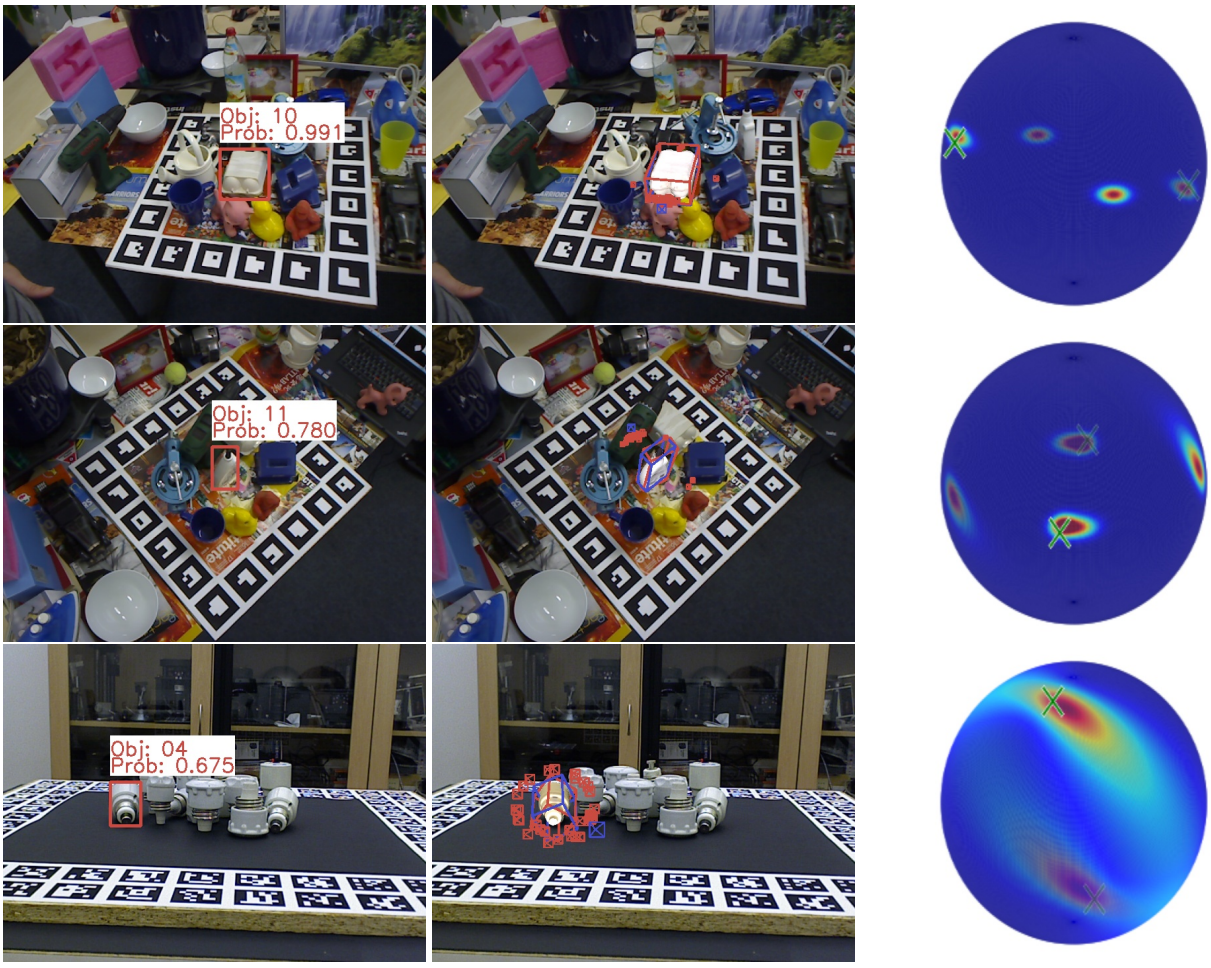
Figure 4: Qualitative results for ‘LineMOD Occlusion’

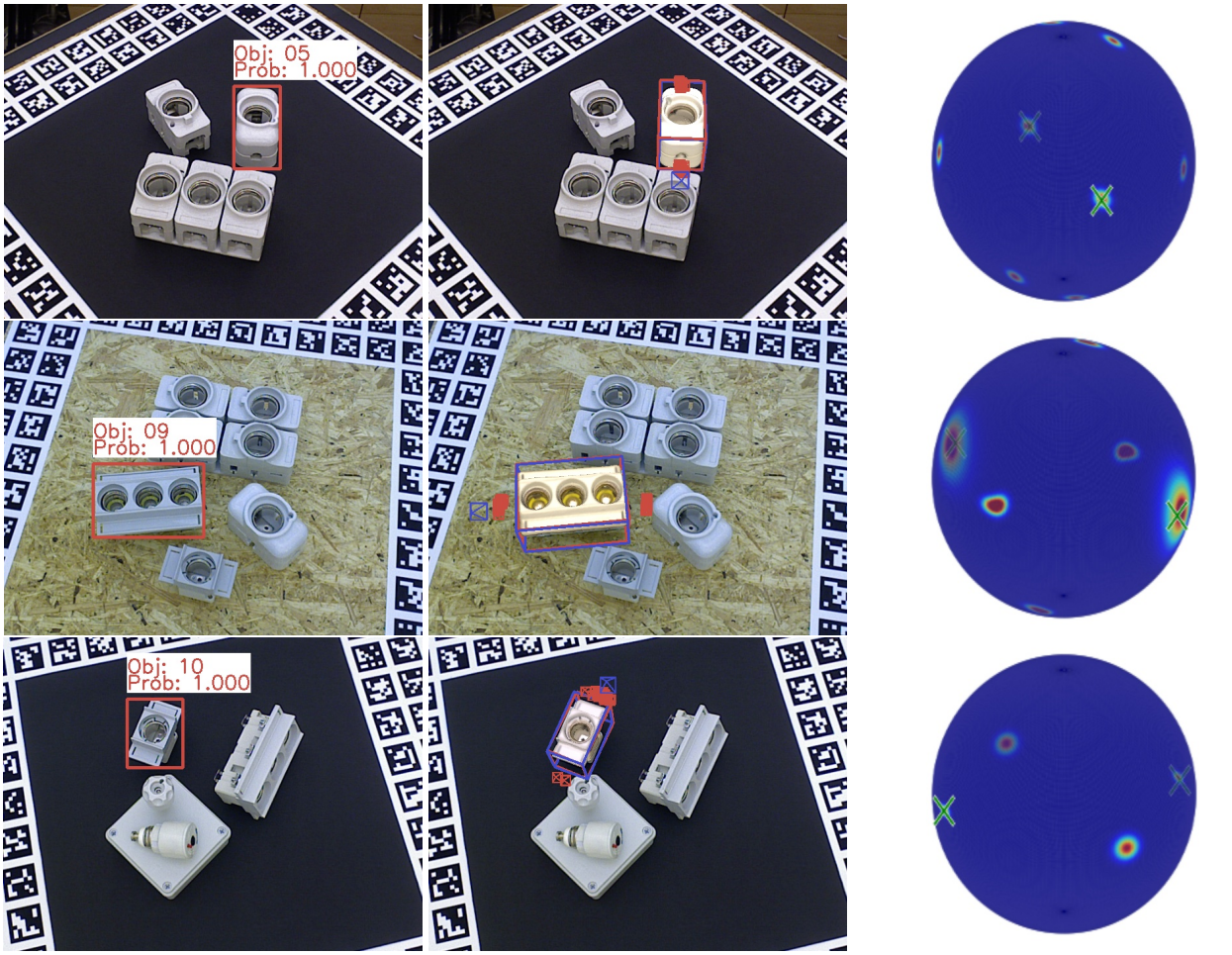
3.2. Ambiguous Object Detection and Pose Estimation

Object	'eggb'	'glue'	obj_04		obj_05			obj_09		obj_10	
Scene	–	–	5	9	2	3	9	5	11	5	11
ADI [%]	54.0	54.1	15.1	20.5	87.6	62.1	84.4	66.5	74.2	39.8	38.2
VSS [%]	83.5	74.3	68.4	76.0	89.2	85.9	87.6	84.4	84.4	82.4	83.2

Table 3: Detailed evaluation scores the *ambiguous* dataset.

Since comparing against the ground truth is not suitable in a multiple hypothesis scenario, only metrics that do not rely on this value are apt for this case. We thus chose the Visual Surface Similarity [5, 8] and Average Distance of Indistinguishable points [4] as metrics for pose. We always take the detection with the highest confidence. We present our individual scores for the *ambiguous* dataset in Tab 3. Additionally, below we show one qualitative sample for each object.





(a) Input Image (b) 2D Detections (c) 6D Pose and Associated Hypotheses

Figure 5: Qualitative results for the ambiguous objects.

4. Employing Multiple Hypothesis as Measurement For Reliability

We would like to present more qualitative samples that the hypotheses can be employed as measurement for confidence. To this end, for each object of the *unambiguous* dataset we show the poses possessing the lowest and the highest standard deviation in the hypotheses.

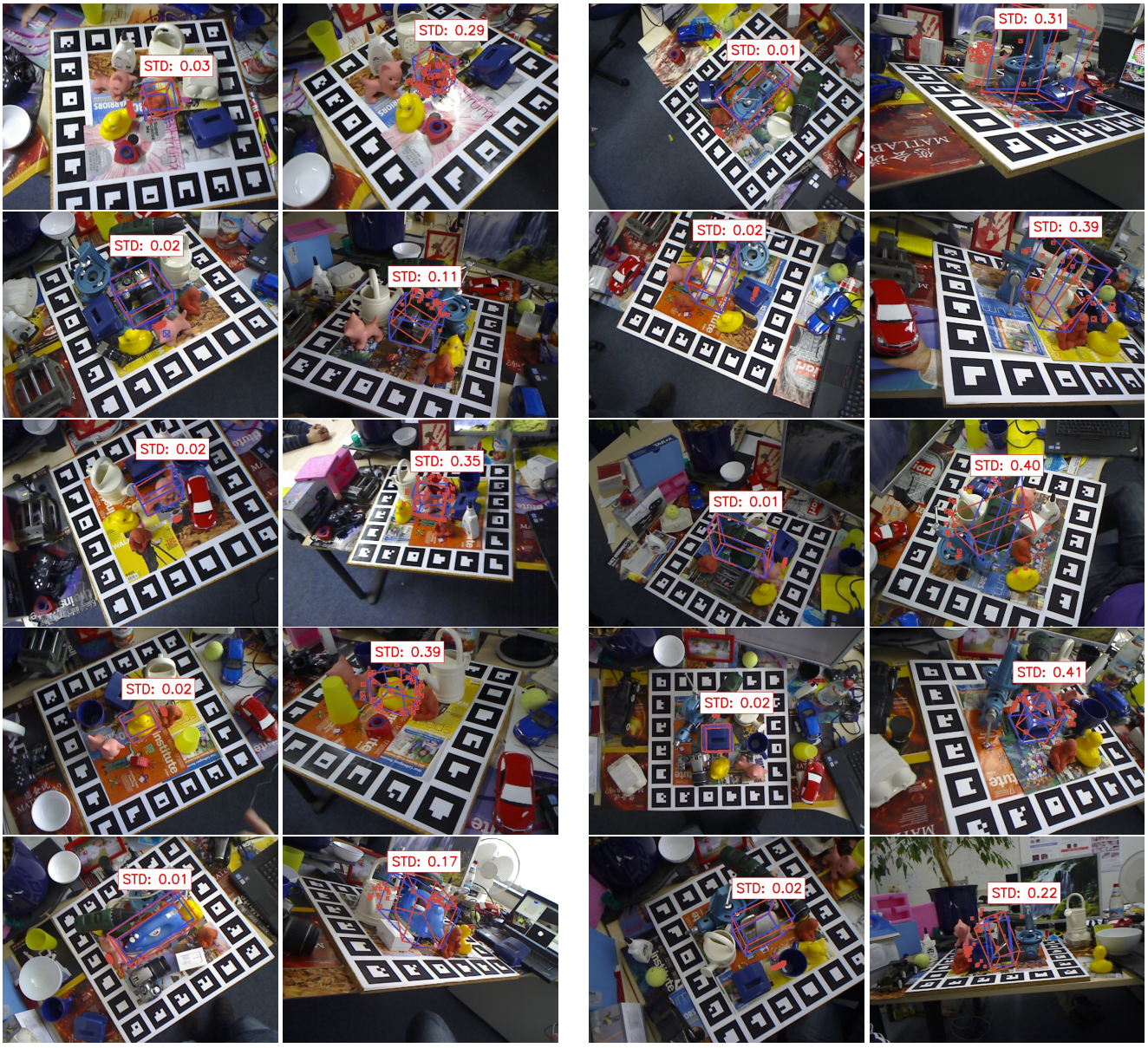


Figure 6: Qualitative examples referring to each object of the *unambiguous* dataset. We show the pose with the lowest (left) and the highest (right) standard deviation in the hypotheses. Thereby, the blue bounding box depicts the ground truth pose, the red bounding box the predicted pose and the red frustums illustrate the hypotheses.

5. Implementation Details

We implemented our method in TensorFlow [1] v1.5 and conducted all experiments on an Intel i7-5820K@3.3GHz CPU with an NVIDIA GTX 1080 GPU. We train with a batch size of 10 and use Adam [6] with a learning rate of 10^{-4}.

We decay the relaxation weight ϵ from 0.05 to 0.01 during training (Eq. 7). Further, we empirically set $\alpha = 1.0$ and linearly increment β from 3 to 10 (Eq. 8). Finally, we set $\lambda = 3$, which balances rotation and translation in the final loss (Eq. 10).

To avoid hypotheses to *die* due to bad initialization, besides sharing loss through the relaxation weight ϵ, we also employ *Hypotheses Dropout*: during training we deactivate a hypothesis with a probability of $p = 50\%$ for the current image.

The mean shift and PCA implementations were taken from scikit-learn. We use `verify_6D_poses` in rendering/utils.py from [5]'s git repository to find the best cluster after mean shift.

To estimate and plot the Bingham distributions, we referred to this https://github.com/SebastianRiedel/bingham matlab implementation. Given a set of 4D quaternions, we compute the maximum likelihood Bingham distribution employing `bingham_fit`. We then render the sphere conducting an equatorial projection to 3D (`plot_bingham_3d`). Similarly, we also project the groundtruth and single hypothesis quaternions to 3D and superimpose them on the rendered sphere.

The pseudo-code below depicts the 6D pose inference procedure after the input image has been processed by the network.

Algorithm 1: Pose Inference

detections $\leftarrow$ { };
forall *Anchors a* **do**
  # Check If Confidence of Anchor Box is Larger Than Threshold
  if *confidence(a)* $\geq$ *threshold* **then**
    # Extract Object ID and Bounding Box Center
    o $\leftarrow$ object(a);
    x, y $\leftarrow$ center(a);
    # Extract Rotation and Depth hypotheses
    $\{q_j\}_{j=1}^{m} \leftarrow$ rotations(a);
    $\{d_j\}_{j=1}^{m} \leftarrow$ depths(a);
    # Principal Component Analysis on Rotation Hypotheses
    $\{e_j\}_{j=1}^{u}$, $\{v_j\}_{j=1}^{u} \leftarrow$ PCA($\{q_j\}_{j=1}^{m}$);
    if $e_1 \geq 0.8$*)* **then**
      # Pose is Ambiguous
      $\{c_j\}_{j=1}^{n} \leftarrow$ meanshift($\{q_j\}_{j=1}^{m}$);
      R, Z $\leftarrow$ contours($\{c_j\}_{j=1}^{n}$, $\{d_j\}_{j=1}^{m}$);
    else
      # Pose is Unambiguous
      R $\leftarrow$ weiszfeld($\{q_j\}_{j=1}^{m}$);
      Z $\leftarrow$ median($\{d_j\}_{j=1}^{m}$);
    end
    # Compute 3D Translation
    t $\leftarrow$ backproject(camera, x, y, Z);
    detections.append({o, R, t});
  end
end
return detections;

6. Synthetic Training Samples

Figure 7: Example from the utilized training datasets. Left: 'T-LESS' - Right: 'LineMOD'

We generate **synthetic samples** by rendering objects with random poses onto images from the MS COCO dataset [7]. Using OpenGL commands we generate a random pose from a valid range: 360 on the azimuth and altitude along a view sphere, and 180 for inplane rotation. We also vary the radius of the viewing sphere to enable multi-scale detection. In order to increase the variance of the dataset, we add random perturbations such as illumination and contrast changes, among others. This is a similar approach to [5, 9]. However, in contrast to them, for each assigned anchor box, we save exactly one 4D quaternion as the ground truth for the rotation, even if ambiguous.

References

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, X. Zheng, and G. Brain. TensorFlow: A System for Large-Scale Machine Learning TensorFlow: A system for large-scale machine learning. In *OSDI*, 2016. 10

[2] S. Hinterstoisser, S. Holzer, C. Cagniart, S. Ilic, K. Konolige, N. Navab, and V. Lepetit. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In *ICCV*, 2011. 1

[3] T. Hodan, P. Haluza, Š. Obdrzalek, J. Matas, M. Lourakis, and X. Zabulis. T-LESS: An RGB-D dataset for 6D pose estimation of texture-less objects. In *WACV*, 2017. 1

[4] T. Hodan, J. Matas, and S. Obdrzalek. On Evaluation of 6D Object Pose Estimation. In *ECCV Workshop*, 2016. 7

[5] W. Kehl, F. Manhardt, S. Ilic, F. Tombari, and N. Navab. SSD-6D: Making RGB-Based 3D Detection and 6D Pose Estimation Great Again. In *ICCV*, 2017. 7, 10, 11

[6] D. P. Kingma and J. L. Ba. Adam: a Method for Stochastic Optimization. In *ICLR*, 2015. 10

[7] 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 *ECCV*, 2014. 11

[8] F. Manhardt, W. Kehl, N. Navab, and F. Tombari. Deep model-based 6d pose refinement in rgb. In *ECCV*, 2018. 7

[9] M. Sundermeyer, Z.-C. Marton, M. Durner, M. Brucker, and R. Triebel. Implicit 3d orientation learning for 6d object detection from rgb images. In *ECCV*, 2018. 11