Object Pose Estimation with Statistical Guarantees: Conformal Keypoint Detection and Geometric Uncertainty Propagation
Supplementary Material

Heng Yang and Marco Pavone
NVIDIA Research

A1. Proof of Proposition 2

Proof. Let $\{\alpha_i\}_{i=1}^n$ be the calibration scores obtained by applying r in (6) to the calibration set, and $\{\alpha_i^\rho\}_{i=1}^n$ be the scores obtained by applying r_ρ in (8). Observe that $\alpha_i^\rho = \rho(\alpha_i)$ because ρ being monotonically increasing implies $\max \circ \rho = \rho \circ \max$ ("$\circ$" describes function composition). As a result, it follows that $\alpha^\rho_{\pi(\lfloor (n+1)\epsilon \rfloor)} = \rho(\alpha_{\pi(\lfloor (n+1)\epsilon \rfloor)})$. Let F_ρ^ϵ be the ICP set due to r_ρ for a given ϵ, we have

$$\begin{aligned} F_\rho^\epsilon &= \{\boldsymbol{y} \in \mathcal{Y} \mid \max\{\rho(\phi(y_k, f(x)_k))\}_{k=1}^K \leq \alpha^\rho_{\pi(\lfloor (n+1)\epsilon \rfloor)}\} \\ &= \{\boldsymbol{y} \in \mathcal{Y} \mid \rho(\max\{\phi(y_k, f(x)_k)\}_{k=1}^K) \leq \rho(\alpha_{\pi(\lfloor (n+1)\epsilon \rfloor)})\} \\ &= \{\boldsymbol{y} \in \mathcal{Y} \mid \max\{\phi(y_k, f(x)_k)\}_{k=1}^K \leq \alpha_{\pi(\lfloor (n+1)\epsilon \rfloor)}\}, \end{aligned}$$

where the last set is precisely F^ϵ, the ICP set induced by r. □

A2. Proof of Proposition 3

Proof. Recall the ICP set in (10)

$$F^\epsilon(x) = \left\{\boldsymbol{y} \in \mathcal{Y} \mid (y_k - \mu_k)^\mathsf{T} \Lambda_k (y_k - \mu_k) \leq 1, \forall k\right\} \tag{A1}$$

that defines either a (ball) or an (ellipse). From the pinhole camera projection model, we know that the groundtruth keypoints $\boldsymbol{y} = (y_1, \ldots, y_K)$ satisfy

$$y_k = \Pi(R_{\text{gt}} Y_k + t_{\text{gt}}) = \frac{[P(R_{\text{gt}} Y_k + t_{\text{gt}})]_{1:2}}{[P(R_{\text{gt}} Y_k + t_{\text{gt}})]_3}, k = 1, \ldots, K \tag{A2}$$

where $P \in \mathbb{R}^{3\times 3}$ denotes the camera intrinsics, $Y_k \in \mathbb{R}^3$ is location of the k-th 3D keypoint in the object's coordinate frame, $[\boldsymbol{v}]_{1:2}$ (resp. $[\boldsymbol{v}]_3$) denotes the first two (resp. third) entries of a 3D vector $\boldsymbol{v}$. To simplify our notation, we develop (A2) as

$$PR_{\text{gt}} Y_k + Pt_{\text{gt}} = (Y_k^\mathsf{T} \otimes P)\text{vec}\,(R_{\text{gt}}) + Pt_{\text{gt}} = \underbrace{\begin{bmatrix} Y_k^\mathsf{T} \otimes P & P \end{bmatrix}}_{:= U_k \in \mathbb{R}^{3\times 12}} \begin{bmatrix} \text{vec}\,(R_{\text{gt}}) \\ t_{\text{gt}} \end{bmatrix} = \begin{bmatrix} u_{k,1}^\mathsf{T} \\ u_{k,2}^\mathsf{T} \\ u_{k,3}^\mathsf{T} \end{bmatrix} s_{\text{gt}} \tag{A3}$$

$$\implies y_k = \left(\begin{bmatrix} u_{k,1}^\mathsf{T} \\ u_{k,2}^\mathsf{T} \end{bmatrix} s_{\text{gt}}\right) \Big/ (u_{k,3}^\mathsf{T} s_{\text{gt}}), \tag{A4}$$

where $u_{k,j}^\mathsf{T} \in \mathbb{R}^{1\times 12}$ denotes the j-th row of matrix U_k. Notice that $u_{k,3}^\mathsf{T} s_{\text{gt}}$ is the depth of the k-th 3D keypoint in the camera coordinate frame (after rigid transformation $(R_{\text{gt}}, t_{\text{gt}})$).

In front of the camera. Since the camera observes the object, the groundtruth pose s_{gt} must transform the object to lie in front of the camera. Therefore, the keypoints must have positive depth values:

$$u_{k,3}^\mathsf{T} s_{\text{gt}} > 0, k = 1, \ldots, K. \tag{A5}$$

Within ICP sets. We now insert (A4) back to the constraint defined by the ICP set (A1), leading to

$$(y_k - \mu_k)^\mathsf{T} \Lambda_k (y_k - \mu_k) \leq 1 \iff \tag{A6}$$

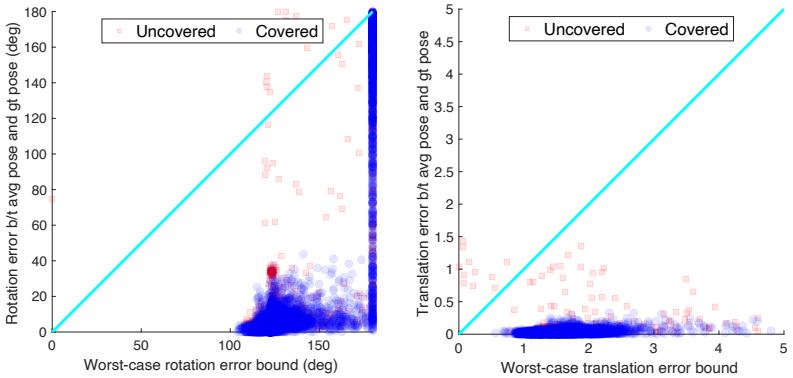
$$\frac{1}{(u_{k,3}^\mathsf{T} s_{\mathrm{gt}})^2} s_{\mathrm{gt}}^\mathsf{T} \begin{bmatrix} u_{k,1} - \mu_{k,1} u_{k,3} & u_{k,2} - \mu_{k,2} u_{k,3} \end{bmatrix} \Lambda_k \begin{bmatrix} u_{k,1}^\mathsf{T} - \mu_{k,1} u_{k,3}^\mathsf{T} \\ u_{k,2}^\mathsf{T} - \mu_{k,2} u_{k,3}^\mathsf{T} \end{bmatrix} s_{\mathrm{gt}} \leq 1 \iff \tag{A7}$$

$$s_{\mathrm{gt}}^\mathsf{T} \begin{bmatrix} u_{k,1} - \mu_{k,1} u_{k,3} & u_{k,2} - \mu_{k,2} u_{k,3} \end{bmatrix} \Lambda_k \begin{bmatrix} u_{k,1}^\mathsf{T} - \mu_{k,1} u_{k,3}^\mathsf{T} \\ u_{k,2}^\mathsf{T} - \mu_{k,2} u_{k,3}^\mathsf{T} \end{bmatrix} s_{\mathrm{gt}} \leq s_{\mathrm{gt}}^\mathsf{T} \left(u_{k,3} u_{k,3}^\mathsf{T} \right) s_{\mathrm{gt}} \iff \tag{A8}$$

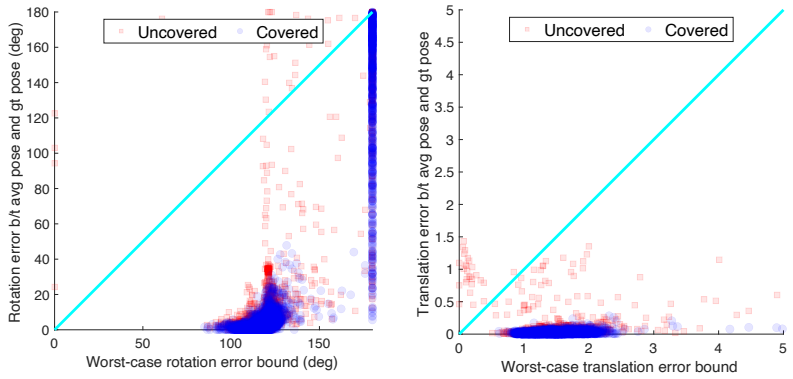
$$s_{\mathrm{gt}}^\mathsf{T} \underbrace{\left(\begin{bmatrix} u_{k,1} - \mu_{k,1} u_{k,3} & u_{k,2} - \mu_{k,2} u_{k,3} \end{bmatrix} \Lambda_k \begin{bmatrix} u_{k,1}^\mathsf{T} - \mu_{k,1} u_{k,3}^\mathsf{T} \\ u_{k,2}^\mathsf{T} - \mu_{k,2} u_{k,3}^\mathsf{T} \end{bmatrix} - u_{k,3} u_{k,3}^\mathsf{T} \right)}_{:= A_k \in \mathbb{S}^{12}} s_{\mathrm{gt}} \leq 0, \tag{A9}$$

which indicates that the groundtruth pose s_{gt} must satisfy K quadratic constraints, one for each keypoint. In summary, the groundtruth pose s_{gt} must lie in the (PURSE) with $b_k = u_{k,3}$ and A_k as in (A9). □

A3. Supplementary Experiments



(a) $\epsilon = 0.1$



(b) $\epsilon = 0.4$

Figure A1. Looser (albeit faster) worst-case error bounds computed from solving first-order relaxation of (12), compared to worst-case error bounds computed from solving second-order relaxation shown in Fig. 3 middle and right columns.

A3.1. Ablation: Relaxation Order

In the main document, we briefly described that we applied second-order semidefinite relaxations to compute the worst-case error bounds in (12) and reported that the average runtime is around 8 seconds on an ordinary workstation. Here we justify the choice of second-order relaxations by showing that first-order relaxations, although much faster (average runtime is about 0.1 seconds), lead to much looser upper bounds for the optimization (12).

To help the reader better understand the approach, we first give a very short introduction to semidefinite relaxations for polynomial optimization problems (POPs). We refer the reader to [6, Section 2] for a detailed introduction.

Polynomial optimization problems (POPs) are problems of the following general formulation

$$\min_{x\in\mathbb{R}^n} \quad p(x) \tag{A10}$$

$$\text{subject to} \quad h_i(x) = 0, i = 1,\dots,l_h, \tag{A11}$$

$$g_j(x) \geq 0, j = 1,\dots,l_g \tag{A12}$$

where $p, \{h_i\}_{i=1}^{l_h}, \{g_j\}_{j=1}^{l_g}$ are all polynomial functions in $x \in \mathbb{R}^n$. Notice that if we denote $s = [\text{vec}\,(R)^\mathsf{T}, t^\mathsf{T}]^\mathsf{T} \in \mathbb{R}^{12}$, it is clear that the cost function of (12) is a polynomial in s when fixing a particular λ (we can add a minus sign to the cost of (12) so that we convert "max" to "min"). The constraints for (12) is $(R,t) \in S^\epsilon$ where S^ϵ has the form in (PURSE). We claim that the (PURSE) can be described by a set of polynomial equalities and inequalities. This is because (i) the rotation constraint $R \in \text{SO}(3)$ can be described by 15 quadratic equality constraints [6]; (ii) the quadratic constraints in (PURSE) are already polynomial constraints; and (iii) the linear inequalities $b_k^\mathsf{T} s > 0$ can be equivalently written as $b_k^\mathsf{T} s \geq \epsilon$ for a small $\epsilon > 0$ (note that $b_k^\mathsf{T} s$ is the depth of the 3D keypoints, so it makes sense to enforce they are larger than, say $\epsilon = 0.001$). We conclude that computing the worst-case error bounds (12) is a POP.

Semidefinite relaxations are a powerful tool to approximate (or even exactly compute) the *global optimal* solutions for (generally nonconvex) POPs. In particular, Lasserre's hierarchy of moment and sums-of-squares relaxations [2] provides a systematic approach to design such semidefinite relaxations. In particular, Lasserre's hierarchy relaxes a POP into a hierarchy of convex semidefinite programs (SDPs) of increasing size. Each relaxation, at a so-called *relaxation order*, in this hierarchy can be solved in polynomial time and provides a valid lower bound for the POP (if the POP aims to maximize, as in (12), then a valid upper bound is provided). Moreover, under mild technical conditions, the lower (upper) bounds of these relaxations coincide with the global optimum of the original POP, in which case we say the relaxation is *exact*, or *tight*.

First-order vs. second-order relaxations. The minimum relaxation order for the POP (12) is 1, since all the polynomials in (12) have degree at most 2 (in general, the minimum relaxation order for a POP is $\lceil d/2 \rceil$, where d is the maximum degree of the polynomials defining a POP). In practice we choose a second-order relaxation instead of a first-order relaxation because first-order relaxations give loose upper bounds for problem (12). Fig. A1 plots the worst-case error bounds computed by solving the first-order relaxation of (12) under the same gt-ball setup. Compared to Fig. 3 middle and right columns, we clearly see that solving the first-order relaxation produces overly conservation upper bounds for (12). For example, when $\epsilon = 0.1$, solving the first-order relaxation never produces a rotation error bound that is below $100°$, while in Fig. 3 we see a cluster of blue circles near the bottom left corner indicating tight bounds.

One nice property of applying semidefinite relaxations is that we get a certificate of global optimality when the relaxation is indeed exact. Such certificates typically come in the form of a rank-one optimal SDP solution, or a relative suboptimality gap (*cf*. [6, eq. (24)]), which indicates exactness of the relaxation when the value is numerically zero (loosely speaking, a relative suboptimality gap of $\epsilon\%$ means that the global optimum of the SDP is at most ϵ percentage away from the global optimum of the POP). When we solve second-order relaxations of problem (12) under the gt-ball setup with $\lambda = 1$, we obtain a relative suboptimality gap that is below 10^{-3} (resp. 10^{-6}) for 99.02% (resp. 72.51%) of the 8784 test problems, indicating that the second-order relaxation is sufficient to obtain (approximately) globally optimal solutions for problem (12).

A3.2. Qualitative ICP Sets

Fig. 1(b) shows circular and elliptical examples of the ICP sets. Fig. A2 provides more examples of the ICP sets with $\epsilon = 0.1$ and $\epsilon = 0.4$. Notice how the ICP sets become smaller when ϵ increases.

A3.3. Worst-case Error Bounds under gt-ellipse, frcnn-ball, and frcnn-ellipse setups

Fig. 3 middle and right columns (from the main document) show the worst-case error bounds (computed from (12)) of the average pose under the gt-ball setup. Fig. A3 shows the worst-case error bounds under the gt-ellipse, frcnn-ball, and frcnn-ellipse setups, which are qualitatively similar to Fig. 3. Notice that the blue circles never cross the $y = x$ diagonal, indicating our bounds are always valid when the PURSE contains the groundtruth pose.

A3.4. A Closer Look at the Conservative Error Bounds

The reader may have noticed two unusual results in the experiments on LM-O. First, the success rate on eggbox is consistently lower than other categories in our methods and other baselines (*e.g.*, PVNet achieves 8.43% success rate on eggbox, while the second lowest success rate is 55.37%). Second, the worst-case error bounds can be overly conservative, *e.g.*, having $180°$ rotation error bounds. It turns out both unusual results can be explained by the same reason: a labelling discrepancy in the LM-O dataset about eggbox.

We noticed the low success rate on eggbox across all baseline methods and contacted the authors of [5], who encountered the same problem. One author told us "*I think this is a mistake or discrepancy in the 6DoF annotations of the dataset itself. As it [the eggbox] is considered a symmetric object, annotators for LMOD might not have consistently annotate it*". Though it is possible to revise the nonconformity score for symmetric objects, the manually chosen keypoints by [5] break the symmetry. Therefore we decided to leave this discrepancy as is because it does not affect our probabilistic guarantees.

This labeling discrepancy, however, does translate to *conservative* prediction sets for the eggbox, in order to contain the (wrong) groundtruth at the desired probability. Fig. A4 shows the eggbox prediction sets are *one order of magnitude* larger than the other categories, leading to worst-case rotation error bounds being mostly 180° (because the PURSE is large enough to cover the entire $\mathrm{SO}(3)$). This indeed shows the advantage of our framework: *the user will see the large uncertainty produced by our algorithm and be alerted*!

Finally, because the PURSE is too large, RANSAG essentially returns a random sample in $\mathrm{SO}(3)$, which has zero probability being close to the (wrong) groundtruth. Hence, a 0% success rate makes sense.

A3.5. Best Worst-case Error Bounds from Samples (Remark 4)

In Remark 4, we discussed that since solving (12) can provide worst-case error bounds for any pose estimator, the natural question is to ask if we can find better pose estimators (than the average pose computed from RANSAG) with tighter worst-case error bounds, which boils down to solving the minimax problem in (14). However, problem (14) is much more challenging to solve than (12), and to the best of our knowledge, there is no efficient way to obtain a globally optimal solution. We think a good future research direction may be to explore methods in [4] or [3] for solving (14).

In this section, we provide a very preliminary study to explore if (14) can indeed offer us tighter error bounds. Towards this goal, we randomly select $M = 5$ pose samples $\{(R_i, t_i)\}_{i=1}^M$ from the results of RANSAG (recall RANSAG not only returns an average pose, but also returns a set of poses), and compute

$$\underline{d}^2_{\epsilon,\lambda} = \min \left\{ d^2_{i,\epsilon,\lambda} = \max_{(R,t)\in S^\epsilon} \lambda \|R - R_i\|_{\mathrm{F}}^2 + (1-\lambda)\|t - t_i\|^2 \right\}_{i=1}^M, \tag{A13}$$

which first solves (12) (inner "max" in (A13)) for each (R_i, t_i) and then selects the minimum (tightest) error bounds. Note that we still apply a second-order SDP relaxation when computing the error bounds for each (R_i, t_i) since (12) is nonconvex.

Fig. A5 plots the cumulative distribution functions (CDF) of the error bounds under the gt-ball setup with $\epsilon = 0.1$. The blue curves plot the CDF of the error bounds computed for the average pose, while the red curves plot the CDF of the error bounds computed from solving (A13). We can see that solving (A13) does slightly improve the tightness of the translation bounds (while the rotation bounds are very close). Considering that we only select the minimum error bounds from $M = 5$ samples, we conjecture solving the minimax problem can give us much tighter error bounds, and we leave this as an exciting future research.

References

[1] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6d object pose estimation using 3d object coordinates. In *European Conf. on Computer Vision (ECCV)*, pages 536–551. Springer, 2014. 5

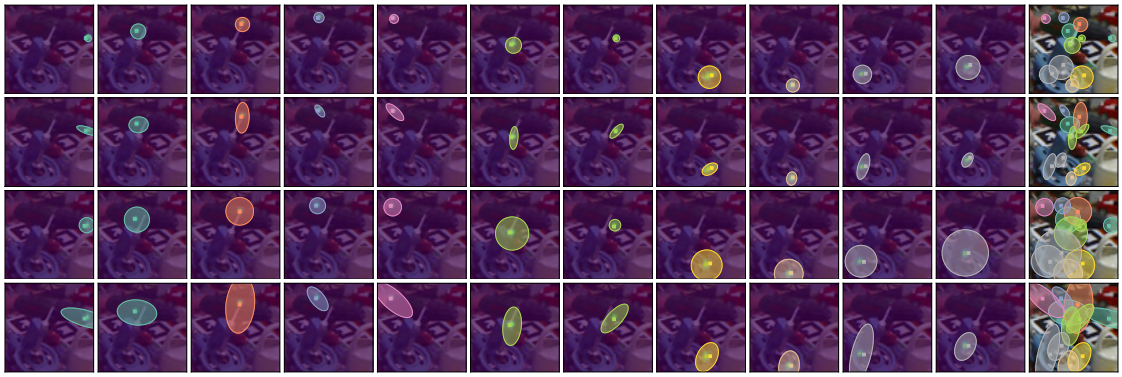
[2] Jean B Lasserre. Global optimization with polynomials and the problem of moments. *SIAM J. Optim.*, 11(3):796–817, 2001. 3

[3] Jiawang Nie, Li Wang, and Jane J Ye. Bilevel polynomial programs and semidefinite relaxation methods. *SIAM Journal on Optimization*, 27(3):1728–1757, 2017. 4

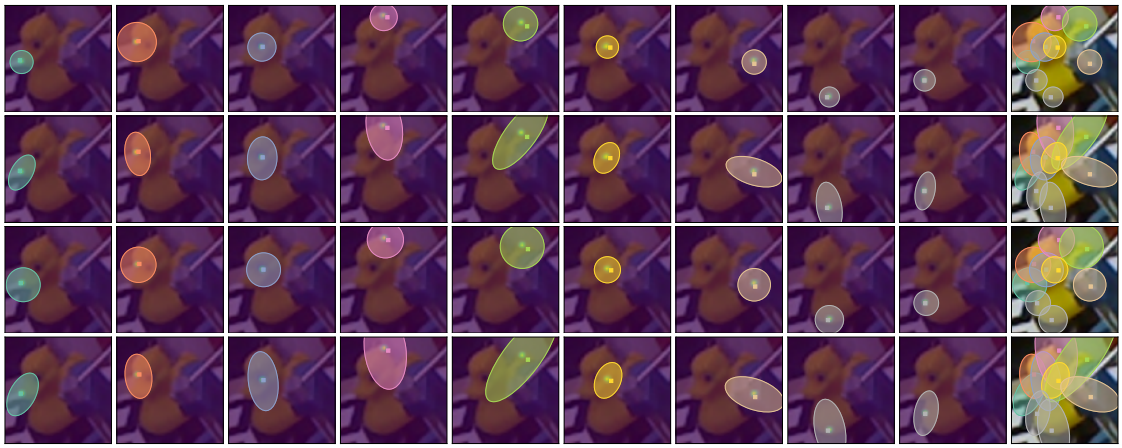
[4] Luis Pineda, Taosha Fan, Maurizio Monge, Shobha Venkataraman, Paloma Sodhi, Ricky TQ Chen, Joseph Ortiz, Daniel DeTone, Austin Wang, Stuart Anderson, Jing Dong, Brandon Amos, and Mustafa Mukadam. Theseus: A Library for Differentiable Nonlinear Optimization. In *Advances in Neural Information Processing Systems (NIPS)*, 2022. 4

[5] Karl Schmeckpeper, Philip R Osteen, Yufu Wang, Georgios Pavlakos, Kenneth Chaney, Wyatt Jordan, Xiaowei Zhou, Konstantinos G Derpanis, and Kostas Daniilidis. Semantic keypoint-based pose estimation from single RGB frames. *J. of Field Robotics*, 2022. 4

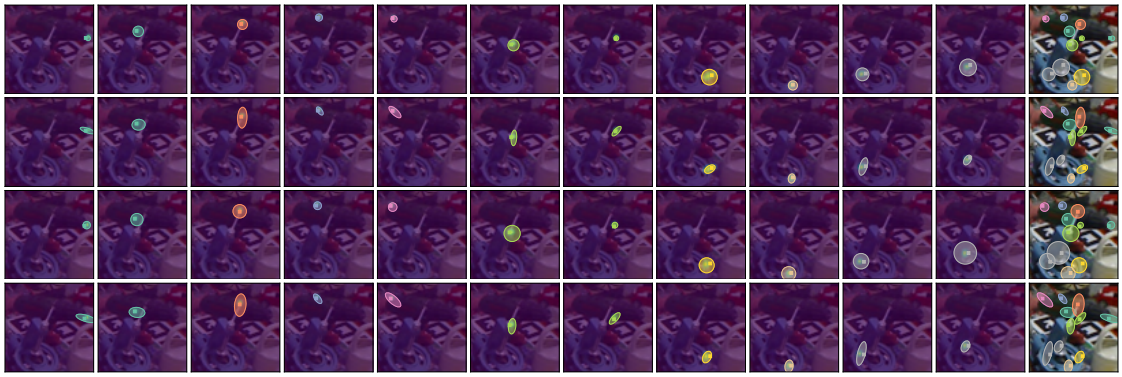
[6] Heng Yang and Luca Carlone. Certifiably optimal outlier-robust geometric perception: Semidefinite relaxations and scalable global optimization. *IEEE Trans. Pattern Anal. Machine Intell.*, 2022. 2, 3



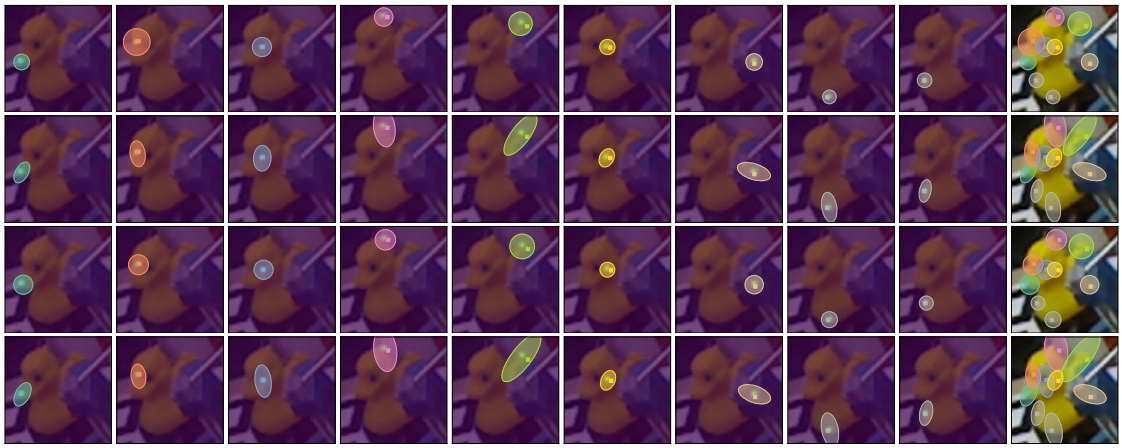
(a) $\epsilon = 0.1$, object: *driller*, top to bottom: gt-ball, gt-ellipse, frcnn-ball, frcnn-ellipse



(b) $\epsilon = 0.1$, object: *duck*, top to bottom: gt-ball, gt-ellipse, frcnn-ball, frcnn-ellipse

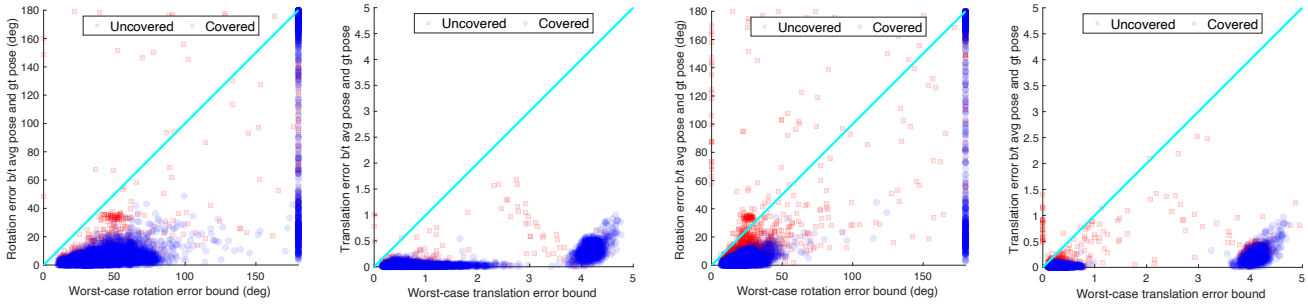


(c) $\epsilon = 0.4$, object: driller, top to bottom: gt-ball, gt-ellipse, frcnn-ball, frcnn-ellipse

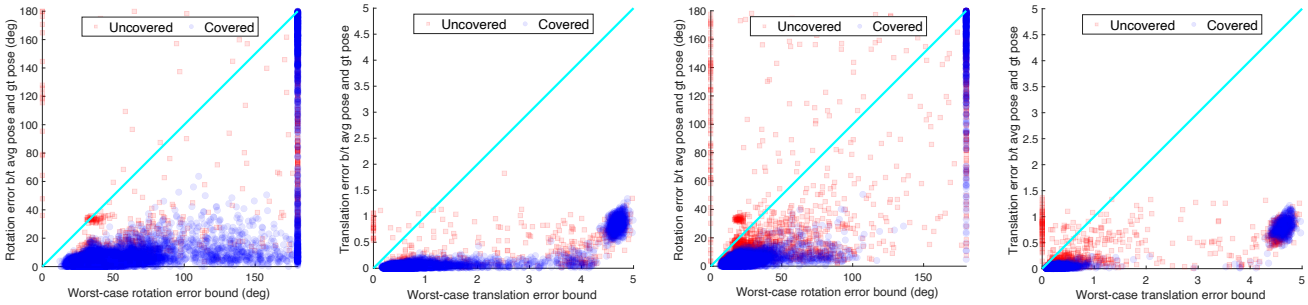


(d) $\epsilon = 0.4$, object: *duck*, top to bottom: gt-ball, gt-ellipse, frcnn-ball, frcnn-ellipse

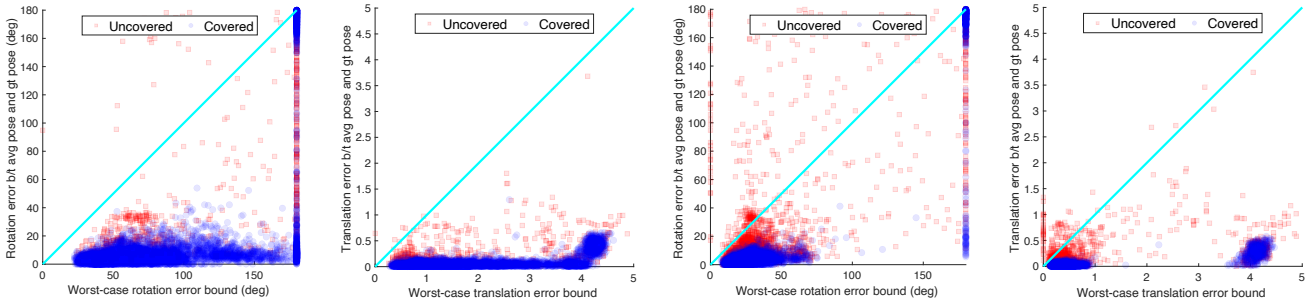
Figure A2. ICP sets on LM-O [1]. Last image of each row overlays *all* groundtruth keypoints (squares) and ICP sets (balls & ellipses) on the original image. Other images overlay the heatmap, groundtruth location, and ICP set of a *single* keypoint on the original image.



(a) gt-ellipse. Left two columns: $\epsilon = 0.1$; right two columns: $\epsilon = 0.4$.



(b) frcnn-ball. Left two columns: $\epsilon = 0.1$; right two columns: $\epsilon = 0.4$.



(c) frcnn-ellipse. Left two columns: $\epsilon = 0.1$; right two columns: $\epsilon = 0.4$.

Figure A3. Worst-case error bounds under (a) gt-ellipse, (b) frcnn-ball, and (c) frcnn-ellipse setups. x-axis represents the worst-case error bounds computed from (12), y-axis represents the actual error between average pose and groundtruth pose. The area below the diagonal $y = x$ indicates correctness of the bounds (*i.e.*, bound $\geq$ error), and points that are closer to the diagonal from below indicate *tighter* bounds (perfect if precisely lie on the diagonal). Blue circles plot cases where the PURSE covers the groundtruth pose and red squares plot cases were the PURSE does not cover the groundtruth. Notice that blue circles never cross the diagonal and our bounds are correct when the PURSE contains the pose (which holds with $1 - \epsilon$ marginal probability).

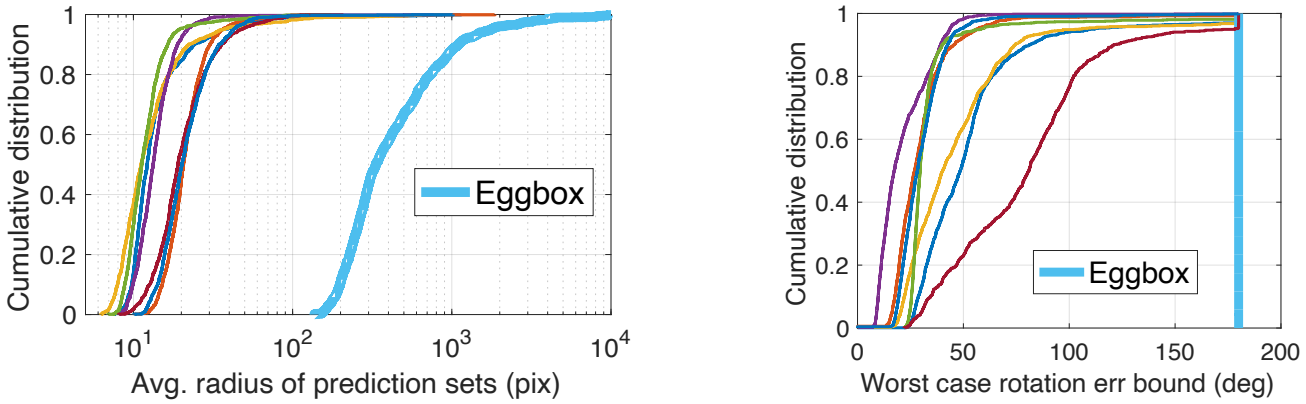


Figure A4. Left: cumulative distribution (CDF) of the average radius of prediction sets (under gt-ball). Right: CDF of the worst-case rotation error bounds; *eggbox* error bounds are mostly 180°.

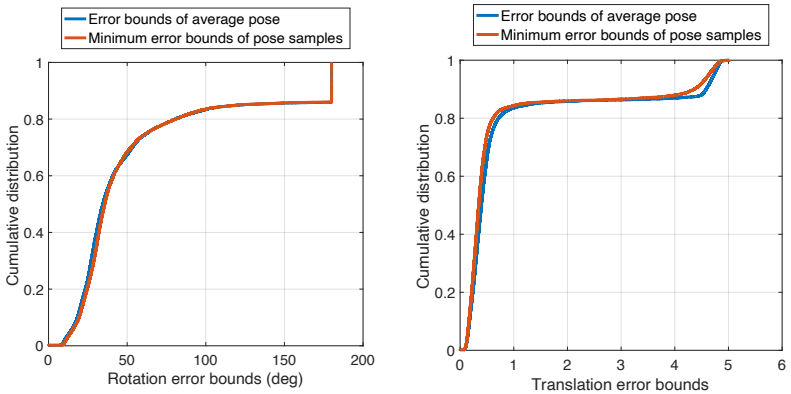


Figure A5. Cumulative distribution function (CDF) of the worst-case error bounds under the gt-ball setup with $\epsilon = 0.1$. Blue curve plots the CDF of the error bounds of the average pose, and red curve plots the CDF of the minimum error bounds of the pose samples (*i.e.* solving (A13)). We can see that the translation error bounds are slightly tightened by selecting the minimum error bounds for multiple pose samples.