AlignPose: Generalizable 6D Pose Estimation via Multi-view Feature-metric Alignment

Supplementary Material

Appendix

This appendix contains additional implementation details for our method (Sec. A) and supplementary experimental results supporting our design choices (Sec. B). Additional qualitative results on the HouseCat6D and ITODD-MV datasets are presented in Sec. C. Failure cases are summarized in Sec. D.

A. Implementation Details

This section is organized according to the stages of our pipeline. We first describe how single-view object detections and pose candidates are obtained in Sec. A.1. Details on aggregating these candidates are given in Sec. A.2. The multi-view refinement procedure is split into Sections A.3, A.4 and A.5, that cover the preprocessing, feature extraction, and optimization steps. Finally, we report the hardware setup and timing measurements in Sec. A.6.

A.1. Single-view pose estimation

A.1.1. T-LESS, YCB-V

To evaluate our method on T-LESS and YCB-V dataset, we use single-view pose candidates downloaded from BOP Leaderboard[1], specifically we use the following versions: FoundPose+FeatRef+Megapose-5hyp, GigaPose+GenFlow (5 hypotheses), MegaPose-CNOS_fastSAM+MultiHyp, Co-op (F3DT2D, 5 Hypo). For all ablations on these datasets, we use downloaded single-view pose candidates by Co-op (F3DT2D, 5 Hypo).

A.1.2. HouseCat6D, BOP-Industrial

For HouseCat6D and BOP-Industrial datasets (ITODD-MV, IPD, XYZ-IBD), we do not have access either to 2D detections or single-view 6D pose estimates[2]. Following our goal of building a pose estimation method generalizable to novel objects, we designed a single-view 6D object detection pipeline based on existing generalizable and open-source methods consisting of 2D detection, segmentation and single view pose estimation.

2D detection and segmentation. Inspired by [66], we implemented a three step training-free 2D detection method, called FOSOD (Foundation model based Open-Set Object Detection). First, we detect many object candidates using the open-set 2D detector GroundingDINO [65], using a fairly low threshold of 0.1. We then obtain object masks using SAM2 [71] using detection bounding boxes as box prompts. Finally, we use template matching to match detections to a set of available textured models. In an offline procedure (less than 5 minutes per object), 128 object templates are rendered offline with viewpoints obtained using Fibonnacci sampling of SO(3) [58]. At inference time, 2D detections are used to crop parts of the query image, and a crop descriptor is used to detect an object label following the CNOS methodology [68], [65]. We found that using cosine similarity of DINOv3-B layer 11 CLS tokens as descriptors gave reasonable matching results.

For BOP-Industrial datasets, we use 2D detections by 3PT [61], a transformer-based model trained for CAD-prompted 2D detection, and the winner of the BOP Challenge 2025. Based on these detections as box prompts, we generate segmentations with SAM2 [71].

We evaluate the performance of detection methods and analyze how detection quality impacts 6D pose estimation in Sec. B.5.

Single-view 6D pose estimates. We adopted a coarse-to-fine 6D pose estimation approach. We first compute a coarse pose estimate using FoundPose [70], then use FoundPose feature-metric refinement, and finally refine the results using the MegaPose [63] refiner with 5 iterations. For textureless objects, we adjusted the MegaPose rendering by adding directional lights to accentuate the 3D shape through shading, which would otherwise appear as a featureless gray silhouette. We use this method for all experiments on HouseCat6D and BOP-Industrial datasets.

A.2. Aggregation

Our 3D Non-Maximum Suppression (NMS) is implemented using world-axis-aligned 3D bounding boxes of object pose candidates. We use 3D IoU as an overlap measure with a threshold of 0.4. We apply NMS globally across all object candidates in the scene. We apply NMS at two stages in our pipeline: once during the aggregation stage to filter out redundant initial proposals, and again after the refinement stage to handle cases where object poses may have converged or become significantly overlapping during the refinement. We evaluate different NMS settings in Sec. B.2.

[1] https://bop.felk.cvut.cz/leaderboards/

[2] BOP's default detections and poses are insufficient as they cover only a single view of a scene. For HouseCat6D, we do not have access to model-based RGB pose estimates using a realistic RGB 2D detector.

A.3. Preprocessing

As a preparation for multi-view alignment, we prepare two distinct, fixed representations for each view: a 2D query feature map and a set of 3D registered features. In this section, we describe the generation process inspired by FoundPose [70]. We now describe how the query features and registered features are created for a view associated with camera C and for an object O with a coarse pose estimate $\boldsymbol{T}_{WO}$.

2D query features. Given a 2D bounding box obtained from the coarse pose $\boldsymbol{T}_{WO}$, we crop the image using the perspective cropping method implemented by [70]. For each view, this process yields a cropped image of size 420×420 and a corresponding crop camera C' with pose $\boldsymbol{T}_{WC'}$ in the world coordinates. We then extract a 2D feature map of this image. The cropped image is decomposed into a grid of non-overlapping 14×14 patches, and each of these patches is embedded with a feature descriptor. We use the hidden state of layer 18 of the DINOv2 backbone, which has been empirically found to provide a good balance between positional and semantic information [70]. The resulting feature map is then up-sampled to the crop resolution via bilinear interpolation.

3D registered features. To generate the set of registered features, we render an RGB-D image of the object as it would appear from the crop camera C' given the coarse pose $\boldsymbol{T}_{WO}$. After rendering, we extract a feature descriptor for each patch, analogously to the process used for the query features, but we keep only the descriptors of patches centered on the object rather than the background. Additionally, we lift the 2D descriptors into 3D object space. Each descriptor $\mathbf{p}_i$ corresponding to a patch centered at pixel c_i is assigned the 3D point $\mathbf{x}_i$ in object coordinates that projects to c_i. This yields a set of *registered features* $\mathcal{F}_{C'O} = \{\mathbf{p}_i, \mathbf{x}_i\}$ where $\mathbf{p}_i$ is a patch descriptor and $\mathbf{x}_i$ its corresponding 3D location in object space.

A.4. Feature extraction.

For all feature extraction (except for the feature descriptor ablation study), we use the following configuration: We produce cropped images and rendered images of size 420×420 px and use the DINOv2 ViT-L model with registers to extract 14×14 patch features as our image descriptors. We use layer 18 of the model and apply layer normalization on the output. To save memory and compute, we project the patch features into the top 256 PCA components.

For the feature descriptor ablation study, we use resolutions of 480×480 px for DINOv3 and 432×432 px for RADIO2.5, which are the recommended resolutions and otherwise follow the same process. Both of these models use 16×16 patch size. For the extraction of dense SIFT features, we use the implementation by Kornia [72], with resolution 420×420 px and 14×14 spatial bins.

Table 8. **Our refinement timings.** In this table we report per-object runtime of our multi-view refinement (avg±std).

Method	YCB-V	T-LESS
Refinement time [s]	0.560±0.064	0.556±0.067

Dimensionality reduction. Feature extractors such as DINOv2/v3 produce high-dimensional descriptors (e.g., 1024 dimensions). To make the optimization process tractable, we reduce the dimensionality of these descriptors using Principal Component Analysis (PCA). The principal components are pre-computed for each object type during an offline onboarding stage and then applied to all subsequent patch descriptors of that object.

Specifically, the components are estimated from feature descriptors extracted from renders of the object observed under a diverse set of viewpoints. We render 788 templates per object and extract their feature descriptors, which takes less than 5 minutes per object. The PCA is computed only on the masked part of the renders that contain the object. Note that, contrary to [70], these templates are only used to compute the PCA projection matrix and are not kept in memory at inference time.

A.5. Optimization

Our nonlinear solver is implemented as a Levenberg-Marquardt adapted from [73]. We use the Barron robust loss function with parameters $\alpha = -5$, $c = 0.5$. Using these parameters does not penalize large residuals as aggressively as an ℓ_2 loss, which helps to stabilize the refinement process. A study of the effects of these parameters is presented in Sec. B.3.

A.6. Hardware and timings

We ran all of our experiments on nodes equipped with NVIDIA-L40S GPUs and Intel(R) Xeon(R) 6760P CPUs. Our method takes less than a second per detection to refine aggregated poses. Individual dataset timings are reported in Tab. 8, where we measure time for pose optimization per object candidate in YCB-B and T-LESS datasets. The time scales linearly with the number of object candidates. For reference, CosyPose processes all candidates in a scene jointly (requires at least three objects) and takes on average 0.400±0.695 s *per scene* on YCB-V and 0.277±0.664 s on T-LESS when refining single-view candidates predicted by Co-op.

Table 9. **Multi-view pose estimation of unseen objects on YCB-V and T-LESS.** This table is an extension of Tab. 1 in the main paper. Both our method and CosyPose [62] refine candidates from FoundPose [70], GigaPose [69], MegaPose [63], and Co-Op [67]. Our approach achieves higher performance across datasets.

Dataset	Method	AR	AR_{VSD}	AR_{MSSD}	AR_{MSPD}	AP	AP_{MSSD}	AP_{MSPD}
YCB-V	FoundPose	69.0	60.2	67.0	79.7	63.0	54.5	71.4
	+ CosyPose MV	79.2	73.6	81.4	82.7	76.1	74.3	77.8
	+ Ours	**83.9**	**78.7**	**88.8**	**84.2**	**83.2**	**85.0**	**81.4**
	GigaPose	66.6	57.4	64.2	78.2	63.1	54.2	72.0
	+ CosyPose MV	76.5	70.3	77.8	81.2	70.9	68.3	73.4
	+ Ours	**81.9**	**77.0**	**86.7**	**82.0**	**79.7**	**81.3**	**78.1**
	MegaPose	62.0	53.5	59.7	72.8	56.1	47.8	64.5
	+ CosyPose MV	71.1	65.1	72.3	75.8	64.5	61.8	67.2
	+ Ours	**80.0**	**75.0**	**84.6**	**80.3**	**77.3**	**78.7**	**75.8**
	Co-op	69.7	58.3	66.3	84.6	69.5	57.8	81.2
	+ CosyPose MV	81.0	73.9	83.0	**86.1**	79.2	76.3	**82.1**
	+ Ours	**83.8**	**78.6**	**88.5**	84.2	**83.3**	**84.8**	81.7
T-LESS	FoundPose	57.0	53.6	54.9	62.3	57.0	52.9	61.1
	+ CosyPose MV	66.4	62.4	63.9	67.0	63.0	62.1	64.0
	+ Ours	**84.1**	**80.9**	**85.3**	**85.9**	**88.6**	**88.5**	**88.6**
	GigaPose	58.2	54.9	56.0	63.7	54.3	50.8	57.8
	+ CosyPose MV	61.9	59.5	60.6	65.6	55.9	55.5	57.3
	+ Ours	**81.9**	**79.3**	**83.0**	**83.6**	**84.9**	**84.8**	**85.0**
	Megapose	50.8	48.1	48.5	55.9	50.5	46.6	54.4
	+ CosyPose MV	57.6	55.8	56.7	60.3	56.1	54.9	57.3
	+ Ours	**78.8**	**76.1**	**79.7**	**80.5**	**81.4**	**81.3**	**81.6**
	Co-op	68.2	64.0	65.8	74.8	68.9	64.3	73.4
	+ CosyPose MV	78.7	76.9	78.5	80.7	78.9	78.3	79.4
	+ Ours	**89.6**	**86.3**	**90.9**	**91.5**	**92.4**	**92.3**	**92.5**

Table 10. **HouseCat6D results for the full dataset and metallic and glass categories.** Extended version of Tab. 2 in the main paper. Both our method and CosyPose [62] refine candidates from FoundPose [70]. We demonstrate significant improvements of the pose estimation scores on the groups of glass and metallic objects compared to the single-view and multi-view baselines.

Subset	Method	AR	AR_{VSD}	AR_{MSSD}	AR_{MSPD}	AP	AP_{MSSD}	AP_{MSPD}
Full dataset	FoundPose	80.3	79.1	79.0	82.7	72.2	70.1	74.2
	+ CosyPose MV	86.7	84.5	87.0	88.6	86.6	85.9	87.3
	+ Ours	**88.9**	**85.3**	**90.4**	**91.2**	**89.6**	**89.4**	**89.8**
Glass objects	FoundPose	75.1	66.0	77.6	81.8	71.4	69.2	73.7
	+ CosyPoseMV	84.6	75.8	88.8	89.2	86.8	86.6	87.0
	+ Ours	**88.9**	**79.2**	**92.7**	**95.0**	**92.8**	**91.9**	**93.8**
Metallic objects	FoundPose	19.3	11.2	15.3	31.5	19.0	10.3	27.7
	+ CosyPose MV	25.8	15.9	23.7	37.7	27.1	21.0	33.2
	+ Ours	**43.5**	**27.0**	**49.9**	**53.5**	**46.5**	**45.2**	**47.8**

B. Additional experimental results

B.1. Complete set of results with all metrics

In Tab. 9, 10 and 11, we present additional results of unseen pose estimation, detailing the AR/AP scores of each VSD/MSSD/MSPD metric. Our method is able to refine all single-view pose estimates with a higher accuracy than our baseline CosyPose, as shown by the consistently higher AP/AR scores. An exception to the overall trend appears in Tab. 9 for the MSPD error when refining YCB-V candidates from Co-Op [67]. In this case, our method does not yield an improvement in this single reprojection error

Table 11. **Evaluation on BOP-Industrial datasets.** This table is an extension of Tab. 3 in the main paper. We obtain 2D detections from 3PT [61], generate segmentations by SAM2 [71] and generate candidates using FoundPose [70] official code base.

	Method	AP	AP_{MSPD}	AP_{MSSD}
IPD	FoundPose	31.4	47.3	15.5
	+ CosyPose MV	36.7	48.5	24.8
	+ Ours	**79.8**	**85.2**	**74.3**
XYZ -IBD	FoundPose	32.5	41.4	23.6
	+ CosyPose MV	52.4	56.0	48.8
	+ Ours	**66.5**	**67.4**	**65.7**
ITODD -MV	FoundPose	41.2	49.0	33.4
	+ CosyPose MV	54.5	56.7	52.2
	+ Ours	**76.8**	**75.9**	**77.7**

metric. A likely explanation is that our refinement adjusts the object pose to better align in the spatial/depth direction, thereby reducing MSSD errors (increasing AR_{MSSD} and AP_{MSSD}). However, this stricter alignment in 3D can sometimes lead to a slight increase in the 2D projection MSPD error (decrease of AR_{MSPD} and AP_{MSPD}), for instance when the refined pose corrects depth or orientation errors in ways that shift the projected silhouette. This highlights a trade-off: optimizing for geometric consistency in 3D may not always translate to lower reprojection error in 2D, especially when the input candidates have a very low reprojection error.

Detailed results for HouseCat6D in Tab. 10 show that metallic objects are very challenging for FoundPose, especially their depth alignment (low AR_{MSSD}, AR_{VSD} and AP_{MSSD}). When refined by our method, these metrics improve significantly. Similar trends can be observed for industrial datasets in Tab. 11.

B.2. Non Maximum Suppression ablation

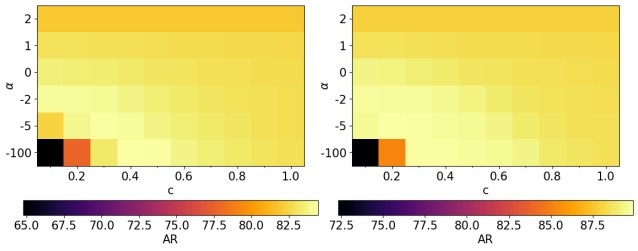
During NMS, each pose candidate is converted to its simple geometric representation which is used to measure overlap. Overlapping detections are then filtered via threshold θ. The ablation of these paremeters is shown in Tab. 12. While low values of θ over-suppress proximal objects and high values of θ retain duplicates, $\theta = 0.4$ provides an optimal balance. At this operating point, the geometric representation for IoU (e.g., axis-aligned boxes vs. bounding spheres) has negligible impact. We employ axis-aligned boxes; bounding spheres are used to approximate the translation-distance NMS of FreezeV2 [60]. The lack of public implementation details for the latter precludes a more direct comparison.

B.3. Cost function ablation

We evaluate our approach under varying settings of the robust cost function. We ablate the Barron general loss [59] with respect to the robustness (α) and scale (c) parameters

Table 12. NMS with different object representation types (AA: axis-aligned boxes; S: bounding spheres) and IoU thresholds for the T-LESS dataset.

Type	$\theta = 0.2$	$\theta = 0.4$	$\theta = 0.6$	$\theta = 0.8$
AA	90.9	91.6	91.4	91.2
S	90.6	91.6	91.6	90.6



(a) YCB-V (b) T-LESS

Figure 5. **Ablation of the cost function parameters.** We ablate the value of parameters α (robustness) and c (scale) of the Barron[59] robust cost function. The maximum is reached at $\alpha = -5$ and $c \in [0.2, 0.3]$.

Table 13. **Ablation of the scoring function.** We compare the proposed scoring based on the average feature-metric loss across views with a minimum or maximum score across views.

Scoring	YCB-V		T-LESS	
	AR	AP	AR	AP
Max	82.7	81.6	87.0	90.5
Min	83.4	82.7	88.4	91.4
Average	**83.8**	**83.3**	**89.6**	**92.4**

in Fig. 5. This family spans several common losses, such as ℓ_2 loss ($\alpha = 2$). Decreasing α, and in particular using negative α, reduces the influence of outliers and increases robustness. We obtain the best performance for $\alpha = -5$. The difference in the final score between the ℓ_2 loss and the optimal robust configuration is within 3%–4%. In all experiments, we use $\alpha = -5$ and $c = 0.5$. Note that very negative α (e.g., $\alpha = -100$) combined with small c (e.g., 0.1) flattens the loss, effectively treating all points as outliers and hindering optimization (see minima of Fig. 5). For further details on the Barron loss, please refer to the original work [59].

B.4. Scoring function ablation

We present an ablation study to assess the importance of the scoring function described in Sec. 3.3 of the main paper. We replace the *average* per-view feature-metric loss score by the *minimal* or the *maximal* score across views. The results are reported in Tab. 13. The chosen average feature-metric loss scoring is better than the alternatives.

Table 14. **Impact of 2D object detection quality on 6D pose estimation.** We compare 3PT [61], FOSOD (our implementation of [66]), and CNOS (SAM) [68] detection methods.

Dataset	Detection method	2D Detection			6D Pose (AP)	
		AP	AR_{10}	AR_{100}	FoundPose	AlignPose (ours)
IPD	3PT	62.8	85.0	85.9	31.4	79.8
	FOSOD	24.0	55.7	56.7	26.7	61.2
	CNOS (SAM)	20.7	26.5	26.5	17.1	37.0
XYZ -IBD	3PT	54.2	35.3	72.4	32.5	66.5
	FOSOD	25.8	22.0	46.4	24.4	50.8
	CNOS (SAM)	27.6	22.7	34.7	28.3	52.4
ITODD -MV	3PT	51.7	78.3	80.2	41.2	76.8
	FOSOD	40.2	61.1	62.8	31.7	69.3
	CNOS (SAM)	31.2	49.3	49.7	32.9	65.8

B.5. Impact of detection quality ablation

To assess how 2D detection quality influences 6D pose estimation, we compare three detectors: 3PT [61], FOSOD (our implementation of [66]), and CNOS (SAM) [68]. For 2D detection, we report standard BOP/COCO metrics: AP, AR_{10}, and AR_{100} [64]. Tab. 14 shows that high-quality 2D detections are essential for 6D pose estimation. Across all datasets, better detection performance consistently leads to higher pose estimation scores. The results confirm that our multi-view method AlignPose significantly outperforms single-view FoundPose across all detection qualities, as it can partially compensate for imperfect detections through multi-view aggregation. The results also illustrate that industrial datasets remain challenging for general-purpose detectors like CNOS and FOSOD.

C. Qualitative results

We provide additional qualitative results for HouseCat6D in Fig. 6 and for ITODD-MV in Fig. 7.

The main trend in HouseCat6D is that single-view predictions are often slightly misaligned with the ground truth due to inaccurate estimation of the distance from the camera. Such errors are especially prominent for the transparent glass objects. The cutlery category presents an even greater challenge because the objects are thin, metallic, and textureless. Single-view predictions are often inconsistent in both rotation and translation, and in some cases objects are not detected in all viewpoints. In these conditions, CosyPose may fail to perform its geometric multi-view refinement. Our method overcomes these issues by refining pose estimates with respect to visual evidence. We recover globally consistent poses even when individual per-view pose estimates are sparse, noisy, or missing entirely.

The ITODD-MV dataset does not provide publicly available ground-truth object poses, therefore, we cannot show a full 3D scene visualization. Instead, in Fig. 7, we show the object contours corresponding to the predicted poses, as seen from four available views. The single-view predictions appear accurate in the "Main View" (the source view used for single-view pose estimation), but become highly misaligned when re-projected to the other views. In contrast, multi-view methods produce poses that are consistent across all views and our method demonstrates better pixel-wise alignment compared to both single-view and CosyPose baselines.

D. Failure Cases and Limitations

We identified the following three failure cases. First, we observe that when all available views come from a similar angle, our method cannot leverage multi-view information fully. This issue becomes more noticeable in challenging situations involving occlusions (Fig. 9). The second failure case appears when the input single-view pose candidate has a very large depth error, causing major cross-view projection misalignment. We provide an example in Fig. 10 where thin cutlery appears correctly placed from one viewpoint, while a second view reveals a large translation error. Our method fails to correct this estimate because the projected pose does not overlap with pixels corresponding to the cutlery. Finally, nearly symmetric objects introduce ambiguity that can cause the optimization to become trapped in local minima. We illustrate this with an example of a cup from the HouseCat6D dataset, which features identical text on both sides (see Fig. 11). As a result, the optimization may converge to an incorrect orientation of the cup.

In addition to these failure cases, our approach shares a common limitation with other methods in this category: it assumes accurately calibrated camera extrinsics. This limitation can be relaxed in future work by extending the objective to jointly optimize both object and camera poses.

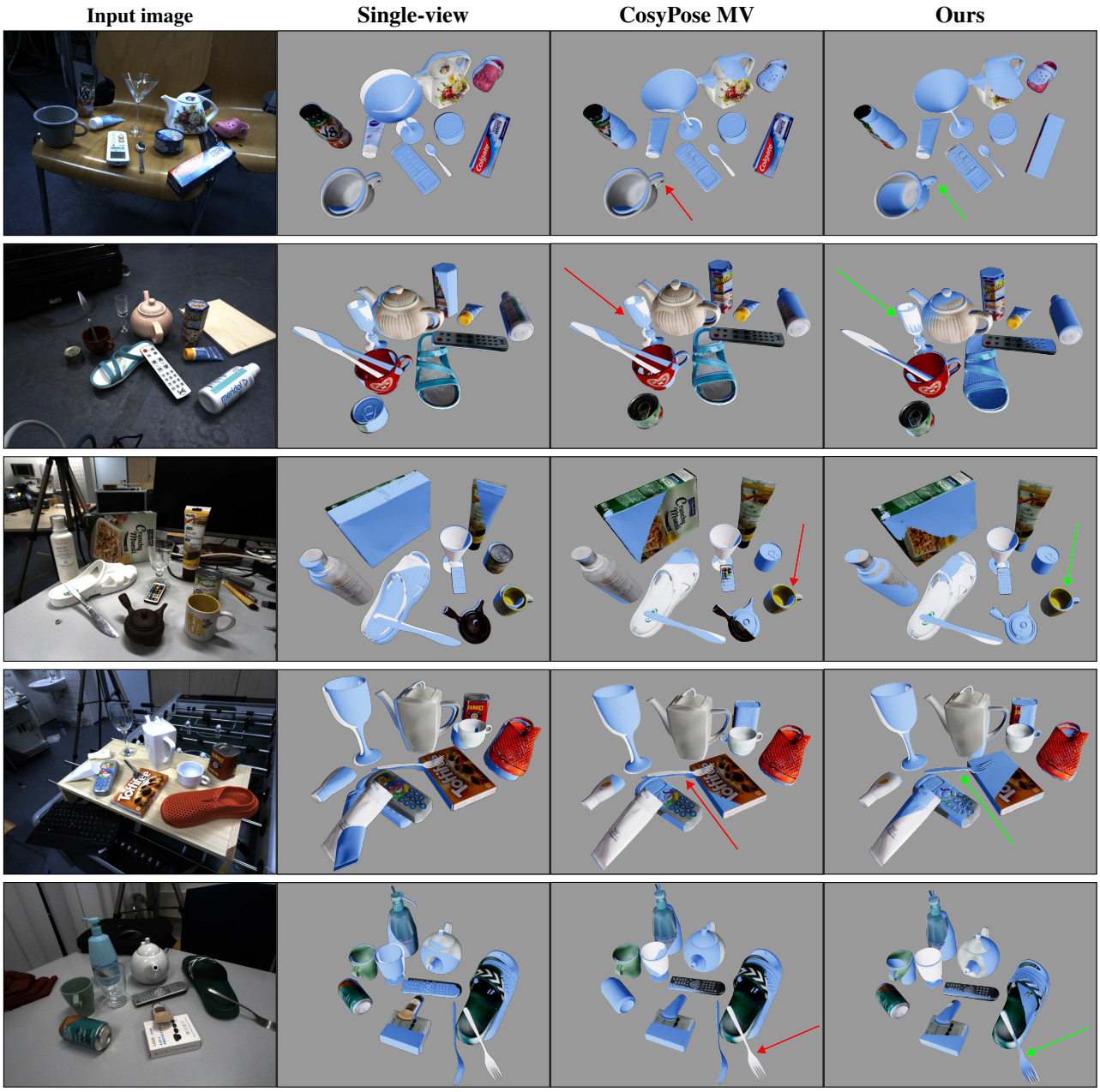


Figure 6. **Qualitative results of multi-view refinement on the HouseCat6D dataset.** Column 1: Representative input image from the four-view set. Column 2: Single-view estimates (blue) overlaid on ground-truth (textured models). Column 3: Multi-view baseline (CosyPose). Column 4: Our proposed refinement. Both CosyPose and our method utilize single-view estimates from four input views. We highlight the improved pose accuracy of our method over CosyPose using green and red arrows respectively across several examples: "mug" (rows 1, 3), "glass" (row 2), and "cutlery" (rows 4, 5). Note that while single-view estimates for cutlery are often significantly misaligned and left uncorrected by CosyPose, our method successfully recovers the poses by leveraging cross-view visual information. With the exception of occasional axial rotations for cutlery, our method achieves highly accurate alignment across all object categories.

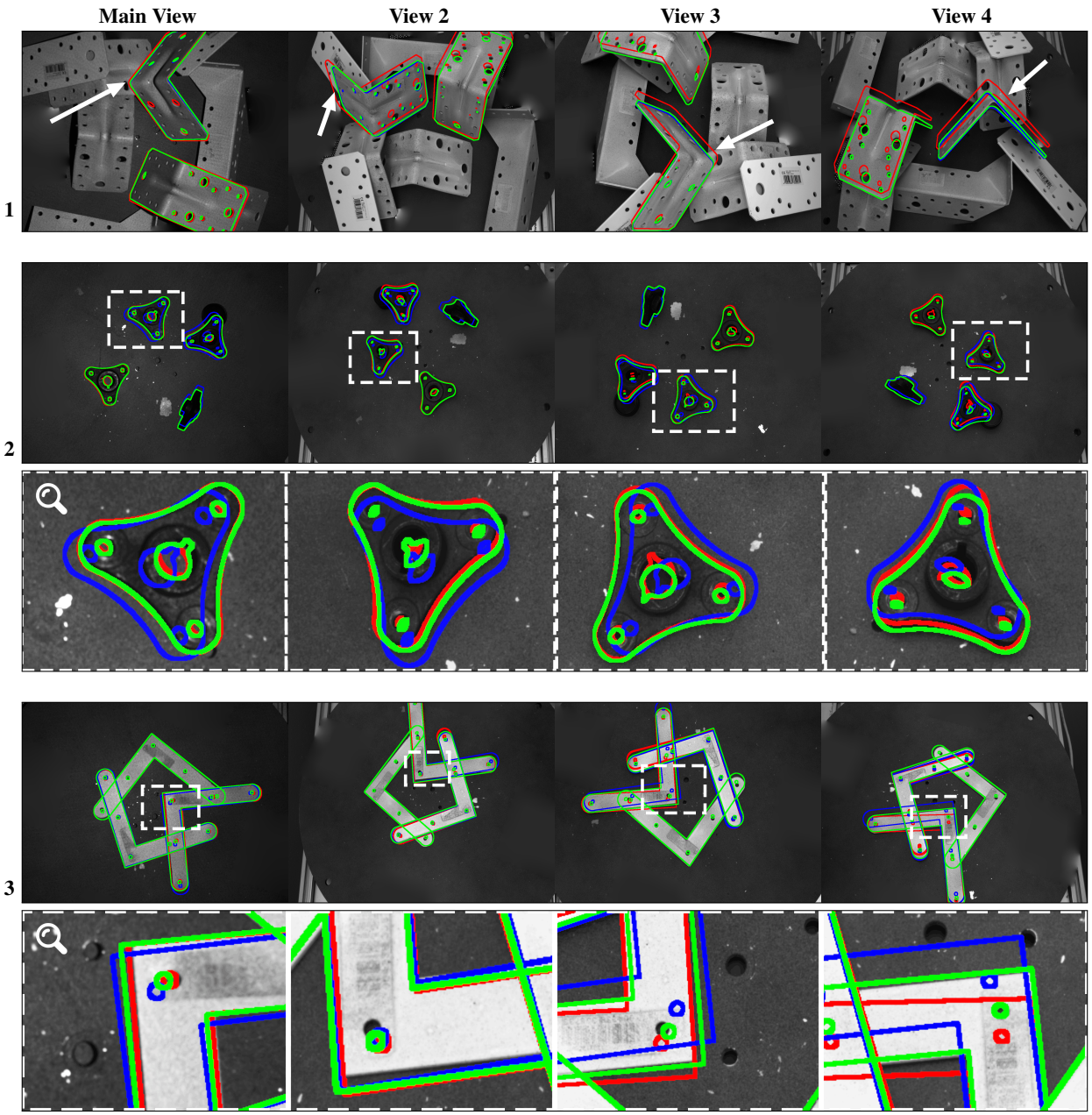


Figure 7. **Qualitative results of multi-view refinement on the ITODD dataset.** Each row shows one scene observed from four viewpoints. Contours indicate predicted object poses. Single-view pose estimates (red) are computed using only the Main View; CosyPose multi-view estimates are shown in blue; and our multi-view estimates are shown in green. Both our method and CosyPose use all four views. **Example 1:** The single-view method produces a reasonable pose in the Main View, but it becomes inaccurate when projected to the other views, as highlighted by the white arrows. In contrast, both multi-view methods yield consistent poses across all four views. **Examples 2 and 3:** Our method achieves higher accuracy than CosyPose: please pay attention to the slight misalignment of the blue contours, which is visible in the zoomed-in images.

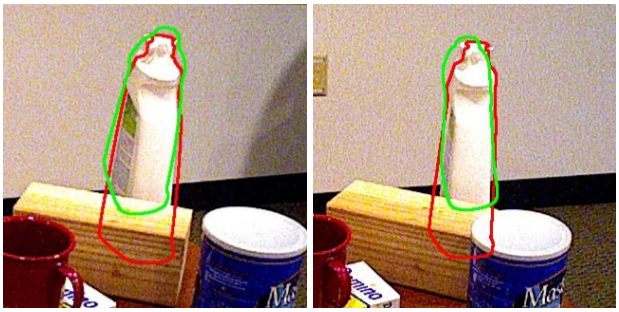


Figure 9. **Failure case I.: Similar viewpoints.** All cameras have similar viewpoints, causing the bleach bottle to be occluded by the wooden block in all views. The predicted pose (green) does not align with the ground truth pose (red) in the occluded regions.

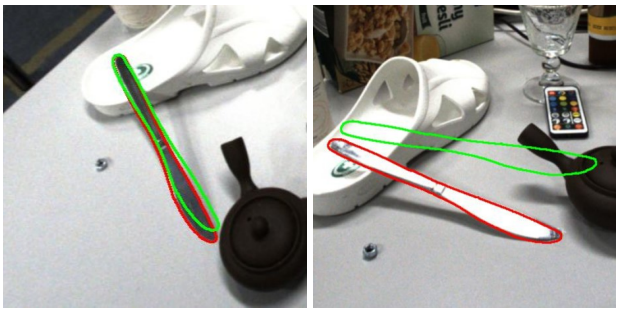


Figure 10. **Failure case II.: Large depth errors.** Pose prediction (green) appears correct from the first view (left), but the second view (right) reveals a translation error too large for optimization.

(a) Front view (b) Back view

(c) Ground truth pose (d) AlignPose prediction

Figure 11. **Failure case III.: Nearly symmetric objects.** Object "cup-new_york" has the same texture on both sides, optimization gets stuck in local minimum.

Supplementary References

[58] Marc Alexa. Super-Fibonacci Spirals: Fast, Low-Discrepancy Sampling of SO(3). In *CVPR*, pages 8291–8300, 2022.

[59] Jonathan T Barron. A General and Adaptive Robust Loss Function. In *CVPR*, pages 4331–4339, 2019.

[60] Andrea Caraffa, Davide Boscaini, and Fabio Poiesi. Accurate and efficient zero-shot 6d pose estimation with frozen foundation models, 2025.

[61] Agastya Kalra, Tim Salzmann, Guy Stoppi, Dmitrii Marin, Rishav Agarwal, Vage Taamazyan, Martin Bokeloh, Stefan Hinterstoisser, Anton Boykov, Alberto Dall'Olio, Pravin Dangol, Kartik Venkataraman, and Huaijin Chen. 3D-Object Perception Transformer (3PT). In *CVPR*, June 2026. To appear.

[62] Yann Labbé, Justin Carpentier, Mathieu Aubry, and Josef Sivic. CosyPose: Consistent Multi-View Multi-Object 6D Pose Estimation. In *ECCV*, pages 574–591, 2020.

[63] Yann Labbé, Lucas Manuelli, Arsalan Mousavian, Stephen Tyree, Stan Birchfield, Jonathan Tremblay, Justin Carpentier, Mathieu Aubry, Dieter Fox, and Josef Sivic. MegaPose: 6D Pose Estimation of Novel Objects via Render & Compare. In *CoRL*, 2022.

[64] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *ECCV*, pages 740–755, 2014.

[65] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection. In *ECCV*, pages 38–55, 2024.

[66] Yangxiao Lu, Yunhui Guo, Nicholas Ruozzi, Yu Xiang, et al. Adapting Pre-Trained Vision Models for Novel Instance Detection and Segmentation. In *IROS*, pages 13341–13348. IEEE, 2025.

[67] Sungphill Moon, Hyeontae Son, Dongcheol Hur, and Sangwook Kim. Co-Op: Correspondence-Based Novel Object Pose Estimation. In *CVPR*, pages 11622–11632, 2025.

[68] Van Nguyen Nguyen, Thibault Groueix, Georgy Ponimatkin, Vincent Lepetit, and Tomas Hodan. CNOS: A Strong Baseline for CAD-Based Novel Object Segmentation. In *ICCV*, pages 2134–2140, 2023.

[69] Van Nguyen Nguyen, Thibault Groueix, Mathieu Salzmann, and Vincent Lepetit. GigaPose: Fast and Robust Novel Object Pose Estimation via One Correspondence. In *CVPR*, pages 9903–9913, 2024.

[70] Evin Pınar Örnek, Yann Labbé, Bugra Tekin, Lingni Ma, Cem Keskin, Christian Forster, and Tomas Hodan. FoundPose: Unseen Object Pose Estimation with Foundation Features. In *ECCV*, pages 163–182, 2024.

[71] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. SAM 2: Segment Anything in Images and Videos. In *ICLR*, pages 28085–28128, 2025.

[72] Edgar Riba, Dmytro Mishkin, Daniel Ponsa, Ethan Rublee, and Gary Bradski. Kornia: An Open Source Differentiable Computer Vision Library for PyTorch. In *WACV*, pages 3674–3683, 2020.

[73] Paul-Edouard Sarlin, Ajaykumar Unagar, Mans Larsson, Hugo Germain, Carl Toft, Viktor Larsson, Marc Pollefeys, Vincent Lepetit, Lars Hammarstrand, Fredrik Kahl, et al. Back to the Feature: Learning Robust Camera Localization from Pixels to Pose. In *CVPR*, pages 3247–3257, 2021.