Co-op: Correspondence-based Novel Object Pose Estimation

Supplementary Material

1. Training Details

In this section, we provid the model configurations, learning rates, and training schedules used for training each model. All three models of Co-op are based on CroCo [19]. Each model takes two images as input and extracts features using a ViT [4] encoder with shared weights. The ViT decoder then processes these two sets of features together and estimates the required outputs through each model's respective head. We implemented our method using PyTorch [18] and the Panda3D renderer [5]. All models are trained using eight NVIDIA A100 GPUs, and we used AdamW [13] as the optimizer.

1.1. Coarse Model

For the coarse model, we use ViT-Large for the encoder and ViT-Base for the decoder. We train the model with a batch size of 64 over 450 epochs, where each epoch consists of 1,800 iterations. The learning rate decreases from 2.0×10^{-5} to 2.0×10^{-6} after 250 epochs, with a warm-up period during the first 50 epochs. Training the coarse model takes approximately two days.

1.2. Refiner Model

The refiner model uses the same encoder and decoder architecture as the coarse model. During training, the batch size is 32, and we use the same learning rate and training schedule as the coarse model. Unlike the coarse model, we use gradient clipping to prevent excessively large gradients caused by unstable correspondences during the early training stages. The gradient clipping value is set to 10^{-2}. Training the refiner model takes approximately three days.

Pose Loss To train the refiner model, we need a pose loss $\mathcal{L}_{pose}$ in addition to the $\mathcal{L}_{cert}$ and $\mathcal{L}_{flow}$ losses described in the main paper. We define $\mathcal{L}_{pose}$ as a disentangled point matching loss, the same as in GenFlow [14].

Given the estimated pose $\mathbf{P} = [\mathbf{R} \mid [\mathbf{t}_x, \mathbf{t}_y, \mathbf{t}_z]^T]$ and the ground truth pose $\mathbf{P}_{gt} = [\bar{\mathbf{R}} \mid [\bar{\mathbf{t}}_x, \bar{\mathbf{t}}_y, \bar{\mathbf{t}}_z]^T]$, $\mathcal{L}_{pose}$ is defined as follows:

$$\begin{aligned} \mathcal{L}_{pose} = \mathcal{D}([\mathbf{R}|[\bar{\mathbf{t}}_x, \bar{\mathbf{t}}_y, \bar{\mathbf{t}}_z]^{\mathrm{T}}], \mathbf{P}_{gt}) \\ +\mathcal{D}([\bar{\mathbf{R}}|[\mathbf{t}_x, \mathbf{t}_y, \bar{\mathbf{t}}_z]^{\mathrm{T}}], \mathbf{P}_{gt}) \\ +\mathcal{D}([\bar{\mathbf{R}}|[\bar{\mathbf{t}}_x, \bar{\mathbf{t}}_y, \mathbf{t}_z]^{\mathrm{T}}], \mathbf{P}_{gt}). \end{aligned} \tag{1}$$

Here, $\mathcal{D}$ is the average L1 distance between two sets of 3D points obtained by transforming the 3D points on the object's surface using the poses $\mathbf{P}$ and $\mathbf{P}_{gt}$, respectively.

1.3. Selection Model

The selection model uses a smaller encoder and decoder architecture compared to the coarse and refiner models. Specifically, we use ViT-Base for the encoder and ViT-Small for the decoder. We train the model for 200 epochs with a batch size of 16, where each epoch consists of 7,200 iterations. The learning rate is 2.0×10^{-5}, and similar to the other models, we have a warm-up period during the first 50 epochs to stabilize the training process. Training the selection model takes approximately 1.5 days.

1.4. Data Augmentation

During model training, we apply random perturbations to the training RGB images to enhance robustness against domain shifts. We employ the same data augmentation techniques as MegaPose [9], which include Gaussian blur, contrast adjustment, brightness adjustment, and color filtering using the Pillow library [1].

2. Additional Experiments

2.1. BOP Benchmark Results Using RGB-D

Although Co-op was trained for object pose estimation using single RGB image inputs, it can also be applied to pose estimation using RGB-D inputs. This is because it estimates the pose based on correspondences between two images. To achieve this, in the coarse estimation stage, we use semi-dense correspondences and employ the Kabsch algorithm [7] instead of PnP [10] for pose estimation. In the refinement stage, instead of returning the estimated pose from the differentiable PnP layer, we estimate the pose using the model's outputs: dense correspondences and certainty. Certainty represents the degree to which the flow from the rendered image reaches the object surface of the query image; therefore, we use only correspondences with a certainty of 0.5 or higher, thereby excluding the influence of occluded correspondences. To obtain the final pose, we utilize MAGSAC++ [2] in our implementation.

Table 1 presents our pose estimation results on the seven core datasets of the BOP challenge using RGB-D inputs. GenFlow [14], the best overall method in the 2023 BOP challenge, reported results only for CNOS [15] detection, while FoundationPose [20] reported results solely for SAM-6D [11] detection. To ensure a fair comparison, we conducted separate experiments for each detection method and reported our results individually. Additionally, we included the submission results that achieved the highest scores among various experimental settings for each compared

#	Method	Detection / Segmentation	LM-O	T-LESS	TUD-L	IC-BIN	ITODD	HB	YCB-V	Mean	Time(sec)
1	MegaPose [9]	CNOS [15]	62.6	48.7	85.1	46.7	46.8	73.0	76.4	62.8	141.965
2	SAM-6D [11]		65.1	47.9	82.5	49.7	56.2	73.8	81.5	65.3	1.254
3	GenFlow [14]		63.5	52.1	86.2	53.4	55.4	77.9	83.3	67.4	34.578
4	GigaPose [16] + GenFlow [14]		67.8	55.6	81.1	56.3	57.5	79.1	82.5	68.6	11.140
5	FreeZe [3]		68.9	52.0	**93.6**	49.9	56.1	79.0	85.3	69.3	13.474
6	Co-op (Coarse)		70.0	64.2	87.9	56.4	56.6	84.2	85.3	72.1	0.978
7	Co-op (1 Hypotheses)		71.5	64.6	90.5	57.5	58.2	85.7	87.4	73.6	2.331
8	Co-op (5 Hypotheses)		**73.0**	**66.4**	90.5	**59.7**	**61.3**	**87.1**	**88.7**	**75.2**	7.162
9	SAM-6D [11]	SAM-6D [11]	69.9	51.5	90.4	58.8	60.2	77.6	84.5	70.4	4.367
10	FreeZe [3]		71.6	53.1	**94.9**	54.5	58.6	79.6	84.0	70.9	11.473
11	FoundationPose [20]		**75.6**	64.6	92.3	50.8	58.0	83.5	**88.9**	73.4	29.317
12	Co-op (Coarse)		70.0	65.4	89.2	56.6	58.4	83.7	85.5	72.7	0.911
13	Co-op (1 Hypotheses)		71.6	65.8	91.5	57.6	60.1	85.1	87.7	74.2	2.265
14	Co-op (5 Hypotheses)		73.0	**67.8**	91.5	**59.8**	**63.1**	**87.5**	**88.9**	**75.9**	7.236

Table 1. **BOP Benchmark Results Using RGB-D** Results on the seven core datasets of the BOP challenge using RGB-D inputs.

#	Method	*LM-O*	*T-LESS*	*TUD-L*	*IC-BIN*	*ITODD*	*HB*	*YCB-V*	*Mean*
1	Ours (Proposed)	65.5	64.8	72.9	54.2	49.0	84.9	68.9	65.7
2	Ours (Same as GenFlow)	65.0	64.7	73.0	53.5	48.7	84.8	69.0	65.5
3	GenFlow [14]	65.1	64.5	72.6	53.0	47.5	84.7	69.0	65.2

Table 2. **Selection Model Ablation Study.** Ablation study on the impact of positive rotation thresholds and backbone in pose selection models.

method. Specifically, we cite MegaPose [9] results using 10 hypotheses and Teaser++ [22], and GenFlow results using 16 hypotheses.

2.2. Pose Selection Ablation

To enhance the pose selection model's ability to identify poses close to the ground truth, we used a smaller rotation threshold when defining positives and negatives. We conducted comparative experiments to verify this approach, as shown in Table 2.

GenFlow reuses the coarse estimation model for pose selection, employing a rotation threshold of 15 degrees. For comparison, we trained a pose selection model with a ConvNeXt [12] architecture, following GenFlow's settings (row 3). We also trained our model using the same configuration as GenFlow (row 2). Comparing these models with our proposed pose selection model (row 1), we confirm that a tighter positive threshold improves accuracy in precise pose estimation.

3. Qualitative Results

3.1. Coarse Estimation

Fig. 1 presents the qualitative results of the coarse estimation. From left to right, the first and second columns display the query image and the template with the highest similarity score, respectively. The third and fourth columns display the segmentation mask from CNOS [15] and the patches that our model did not classify as "no match." From the fifth to the final columns, we present the semi-dense correspondences between the query image and the most similar template, along with the estimated poses derived from them. We can observe that our proposed method accurately estimates correspondences across various datasets, including YCB-V [21], which consists of texture-rich objects, and T-LESS [6], which consists of low-texture objects.

As demonstrated in the third and fourth rows, our proposed method is robust to detection errors. Segmentation masks from CNOS or SAM-6D [11] rely on SAM [8], which does not consider object information, leading to over-segmentation (see row 5) or under-segmentation (see row 8). As reported in the ablation experiments of FoundPose [17], such segmentation errors can significantly affect pose estimation accuracy. In contrast, our model—similar to detector-free methods in correspondence estimation—jointly considers the query image and the template. This approach allows our model to estimate poses robustly against detection errors.

3.2. Pose Refinement

Fig. 2 presents a visualization of the pose refinement process. From left to right, the first two columns display the query image and the rendered image of the initial pose, respectively. The third and fourth columns show the visualization of flow and confidence, which are inputs to the differentiable PnP layer. The fifth to seventh columns show the flow probability, certainty, and sensitivity, which are used to calculate the confidence. The last column shows the model's output—the refined pose of the object.

As seen in the fifth and sixth rows, the flow probability lowers the confidence in ambiguous areas such as the object's self-occlusion. Similarly, the certainty reduces confidence in areas where it is difficult to trust correspondences because of occlusions. Conversely, the sensitivity increases confidence in regions with rich texture (see the eighth row) or at the object's edges (see the sixth row) to achieve accurate pose refinement.

References

[1] The pillow imaging library. https://github.com/python-pillow/pillow. 1

[2] Daniel Barath, Jana Noskova, Maksym Ivashechkin, and Jiri Matas. MAGSAC++, a fast, reliable and accurate robust estimator. In *Conference on Computer Vision and Pattern Recognition*, 2020. 1

[3] Andrea Caraffa, Davide Boscaini, Amir Hamza, and Fabio Poiesi. Freeze: Training-free zero-shot 6d pose estimation with geometric and vision foundation models. In *ECCV*, 2024. 2

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 1

[5] M. Goslin and M.R. Mine. The panda3d graphics engine. *Computer*, 37(10):112–114, 2004. 1

[6] Tomáš Hodaň, Pavel Haluza, Štěpán Obdržálek, Jiří Matas, Manolis Lourakis, and Xenophon Zabulis. T-LESS: An RGB-D dataset for 6D pose estimation of texture-less objects. *WACV*, 2017. 2

[7] Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography*, 32(5):922–923, 1976. 1

[8] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, pages 4015–4026, 2023. 2

[9] 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. 1, 2

[10] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o(n) solution to the pnp problem. *International Journal Of Computer Vision*, 81:155–166, 2009. 1

[11] Jiehong Lin, Lihua Liu, Dekun Lu, and Kui Jia. Sam-6d: Segment anything model meets zero-shot 6d object pose estimation. In *CVPR*, pages 27906–27916, 2024. 1, 2

[12] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, pages 11976–11986, 2022. 2

[13] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2017. 1

[14] Sungphill Moon, Hyeontae Son, Dongcheol Hur, and Sangwook Kim. Genflow: Generalizable recurrent flow for 6d pose refinement of novel objects. In *CVPR*, pages 10039–10049, 2024. 1, 2

[15] Van Nguyen Nguyen, Thibault Groueix, Georgy Ponimatkin, Vincent Lepetit, and Tomas Hodan. Cnos: A strong baseline for cad-based novel object segmentation. In *ICCVW*, pages 2134–2140, 2023. 1, 2, 4

[16] 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. 2

[17] Evin Pınar Örnek, Yann Labbé, Bugra Tekin, Lingni Ma, Cem Keskin, Christian Forster, and Tomáš Hodaň. Foundpose: Unseen object pose estimation with foundation features. In *ECCV*, 2024. 2

[18] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 1

[19] Philippe Weinzaepfel, Thomas Lucas, Vincent Leroy, Yohann Cabon, Vaibhav Arora, Romain Brégier, Gabriela Csurka, Leonid Antsfeld, Boris Chidlovskii, and Jérôme Revaud. Croco v2: Improved cross-view completion pretraining for stereo matching and optical flow. In *ICCV*, pages 17969–17980, 2023. 1

[20] Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects. In *CVPR*, pages 17868–17879, 2024. 1, 2

[21] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. In *RSS*, 2018. 2

[22] H. Yang, J. Shi, and L. Carlone. TEASER: Fast and Certifiable Point Cloud Registration. *IEEE Trans. Robotics*, 2020. 2

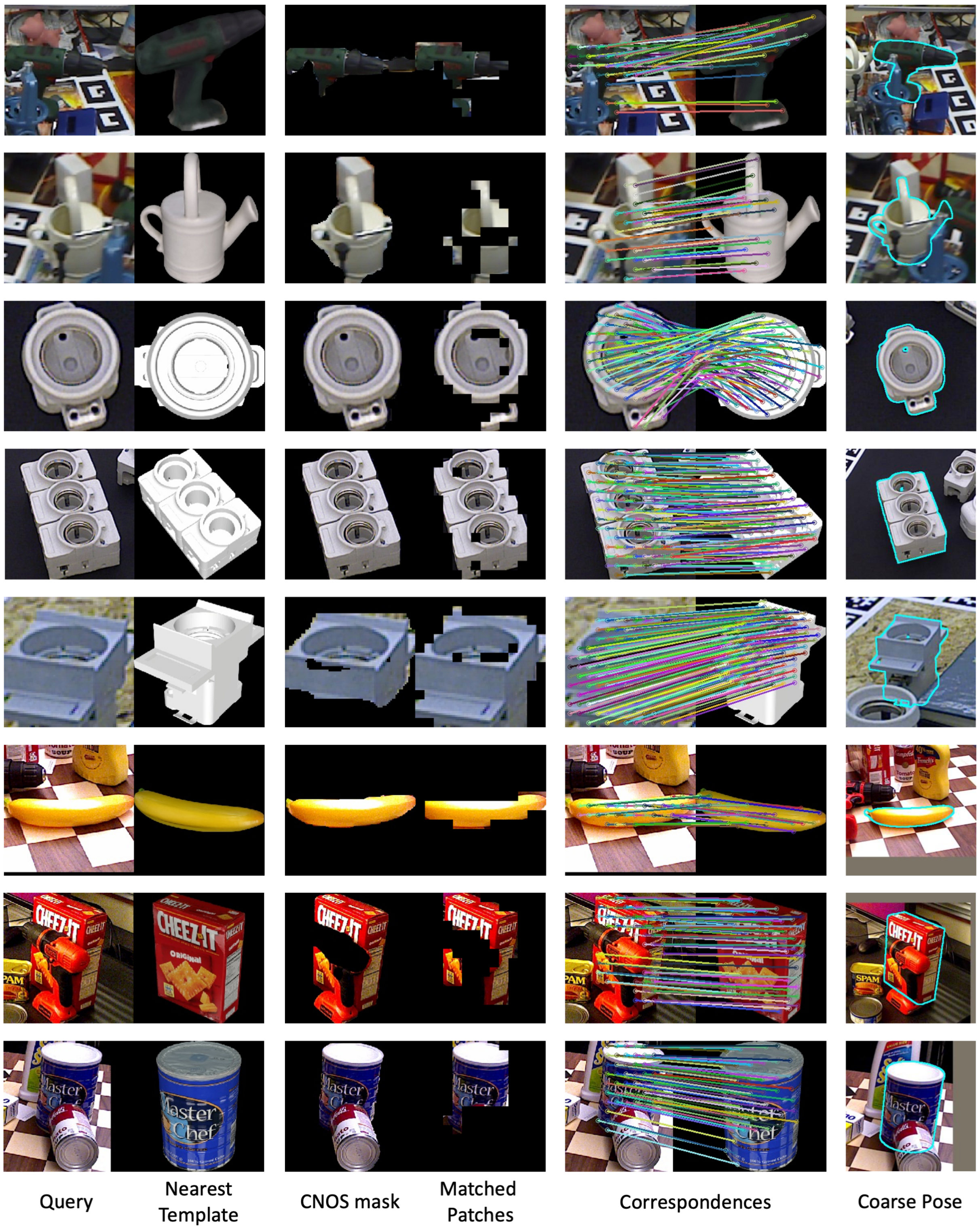

Figure 1. **Qualitative Results of Coarse Estimation.** The first two columns on the left display the model's query image and the template with the highest similarity score to the query image. The third and fourth columns compare the CNOS [15] segmentation mask with patches that the model did not classify as 'no-match'. From the fifth to the last columns, the correspondences between the query image and the template, as well as the resulting pose estimation results, are shown.

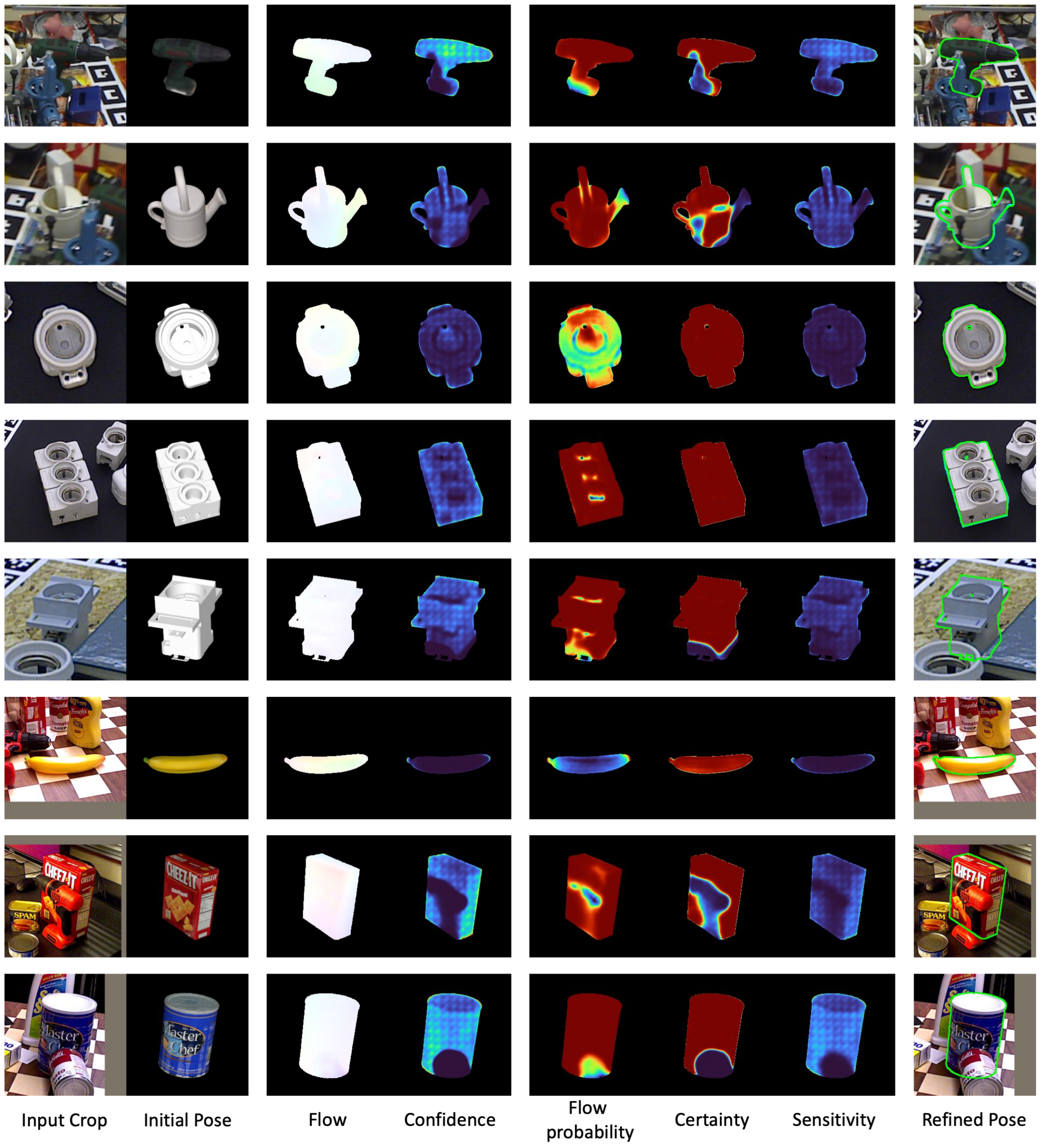

Figure 2. **Qualitative Results of Pose Refinement.** From left to right: query image, initial pose rendering, flow, confidence, flow probability, certainty, sensitivity, and the refined pose (legend: 0.0 ▬ 1.0). The flow probability and certainty reduce confidence in ambiguous or occluded areas, while sensitivity increases confidence in textured regions and object edges to improve pose refinement.