– Supplementary Material –
GenFlow: Generalizable Recurrent Flow for 6D Pose Refinement of Novel Objects

Sungphill Moon* Hyeontae Son* Dongcheol Hur Sangwook Kim
NAVER LABS
{sungphill.moon, son.ht, dongcheol.hur, o.s.w.a.l.k}@naverlabs.com

1. Training Details

Implementation We implement our method using PyTorch [7] and Panda3D renderer [2]. The training procedure for our coarse model is as follows: 630K iterations of the Adam optimization [4] with a batch size of 128, a learning rate decayed from 3×10^{-4} to 3×10^{-5} after 450K steps with a warm-up phase of 45K iterations. Our refiner model is trained with another training schedule: 1620K iterations of the Adam with a batch size of 32, a learning rate decayed from 10^{-4} to 10^{-5} after 900K steps with a warm-up phase of 180K iterations, and a gradient clipping to a value of 10^{-2}. It takes 4.5 days and 3.5 days to train our coarse and refiner models respectively using 8 NVIDIA A100 GPUs.

Data Augmentation During training, we apply the random perturbation to the RGB images for our model to be robust to the domain shift. We use the same data augmentation method as CosyPose [6] and MegaPose [5]. It consists of Gaussian blur, sharpness, contrast, brightness, and color filters. We also render the images to be compared to the input crop with randomly positioned lighting sources.

2. Additional Ablations

We conduct two additional experiments on the same coarse estimation results as the GenFlow design ablation in the main paper.

Inner Loop We report the changes in AR score according to the iterations of inner updates (GenFlow updates) for an outer update in Fig 1. The inner updates are performed on two GenFlow modules half and half, and two updates take approximately 18 milliseconds on an RTX 3090 GPU. The results show the tradeoff between accuracy and runtime.

Using Certainty for RGB-D Inputs For the RGB-D input, we utilize the RANSAC-Kabsch algorithm [1, 3] for depth refinement. Specifically, we filter out the 3D-3D correspondences of certainty lower than a threshold before applying the RANSAC-Kabsch. To validate our method, we compare the three different filtering methods: No filtering, confidence-based, and certainty-based filtering. We use all correspondences on the rendered mask for the no filtering method. We use the threshold value 0.5 for certainty-based filtering and $\frac{\max(\mathbf{W})}{2}$ for confidence-based filtering where $\mathbf{W}$ is the confidence weights from the last GenFlow update. Figure 2 shows the AR scores for 5 BOP datasets according to the filtering methods. The results show that our certainty-based filtering outperforms all other methods. The confidence-based filtering is less effective than others concerning most datasets since it is vulnerable to noisy input depth due to the sparseness of high-confidence correspondences.

*These authors contributed equally.

Method	# of renderings (↓)	Mean Average Recall (↑)	Median Runtime (sec) (↓)					
			LM-O	T-LESS	TUD-L	IC-BIN	YCB-V	MEAN
Naïve	576	22.5	0.80	0.80	0.79	0.79	0.77	0.79
Ours	144	31.3	**0.21**	**0.22**	**0.21**	**0.21**	**0.21**	**0.21**
	208	**35.8**	0.30	0.30	0.30	0.30	0.29	0.30

Table 1. Results of ablation study of the pose hypotheses generation strategy for the coarse pose estimation. The median runtime is reported as the median processing time for coarse pose estimation of each detection. Our method requires fewer renderings and extracting scores while achieving better performance than the Naïve method. Thus, ours can take both time efficiency and accuracy.

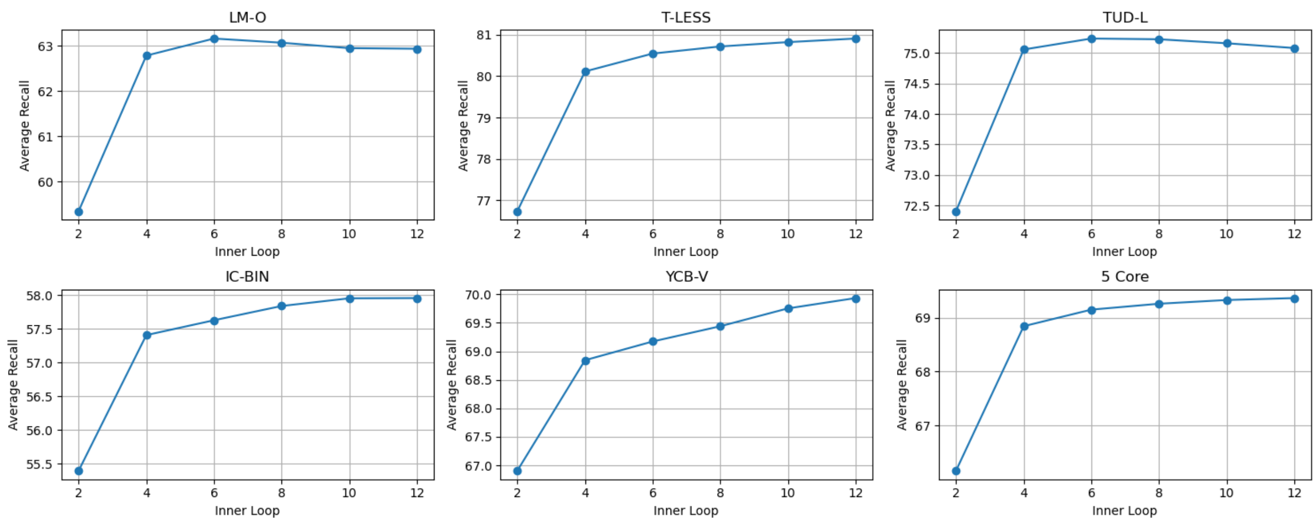

Figure 1. Changes in AR score according to inner updates. The bottom right chart shows the mean of AR score for 5 datasets.

References

[1] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981. 1

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

[3] 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

[4] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 1

[5] 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*. 1

[6] Y. Labbe, J. Carpentier, M. Aubry, and J. Sivic. Cosypose: Consistent multi-view multi-object 6d pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 1

[7] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035, 2019. 1

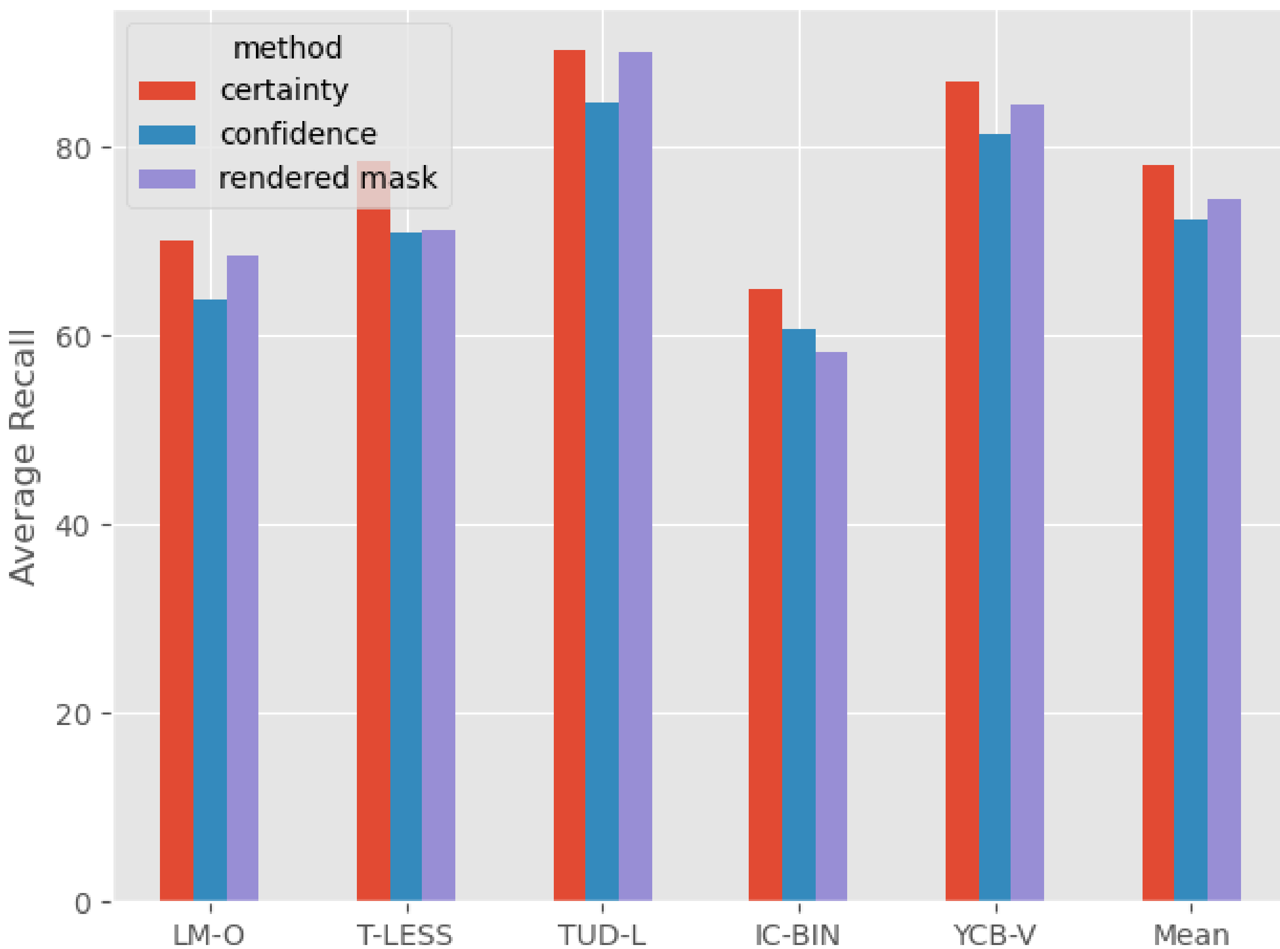

Figure 2. Comparison of methods to filter out the outliers of 3D-3D correspondences.

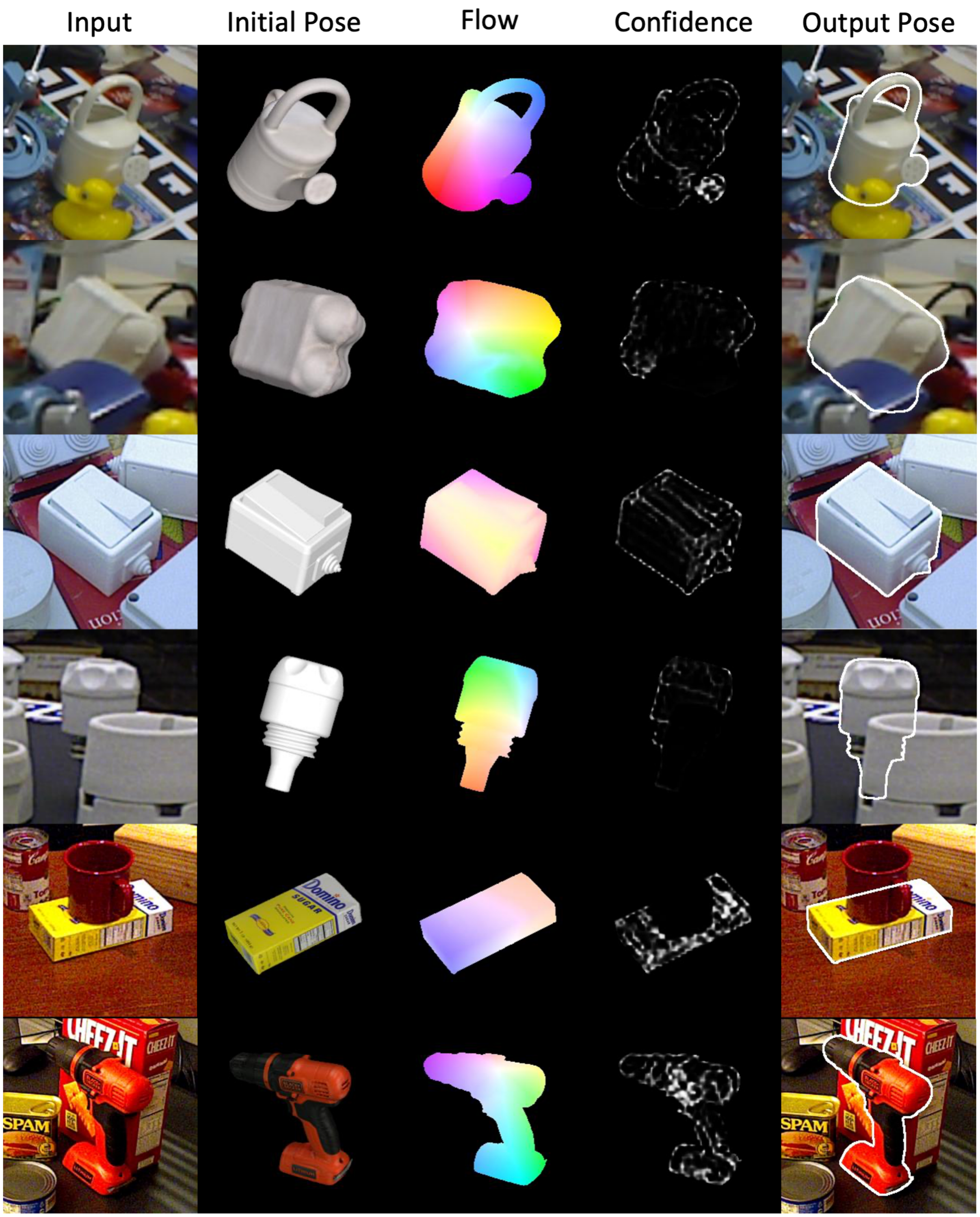

Figure 3. Visualization of the outputs of a single outer update of our refiner. The first column is the input crop, the second column is the synthetic image rendered with the rough initial pose, the third column and fourth column are the optical flow and confidence weights of the last GenFlow update respectively, and the fifth column is the overlay of the input crop and silhouette image with predicted 6D pose.