Active Coarse-to-Fine Segmentation of Moveable Parts from Real Images Supplementary Material

Ruiqi Wang[1], Akshay Gadi Patil[1], Fenggen Yu[1], Hao Zhang[1,2]
[1]Simon Fraser University [2]Amazon

We present additional results, both qualitative and quantitative, here in the supplementary material. We begin by showcasing differences in examples of real scene images from our dataset and compare them with the kind of real images from OPDReal [1] dataset and OPDMulti[3] dataset (Section 1). We then present additional qualitative and quantitative results (Section 2). We conclude the supplementary material by providing the details of our human-in-the-loop framework (Section 3). We also attach a video in the supplementary package that demonstrates human feedback in the active learning setup.

1. Dataset Visualization

We show more visualizations of raw images as well as annotated segmentation masks from OPDReal dataset, OPDMulti dataset and ours, in Fig 1, 2, 3, respectively.

We can observe from the visualization that the segmentation masks for parts in OPDReal are not correctly annotated, with clear misalignments at part boundaries, as shown in Fig 1 (a). Similarly, OPDMulti also suffers from annotation inaccuracies, as both of them obtain the 2D segmentation masks from projection of 3D annotations, a process prone to reconstruction and re-projection errors; see Fig 2 (b). In addition, some images are labeled with noisy annotations, especially on objects with reflective surface such as the glass door in Fig 1 (a) row 4, 8, 9 and Fig 2 (b) row 5. As mentioned in our paper, OPDReal focus on images that contain a single articulated object. Therefore, although there are multiple articulated objects in an image, only one type of them will be annotated as seen in Fig 1 (b). Specifically, some parts of the single object in images are missed such as Fig 1 (b) row 5, where only 2 drawers of the 4-layered cabinet are annotated. Again, OPDMulti has the same issue. Although it claims to be generalized on multiple articulated objects in images, some objects and parts are still unlabeled as shown in Fig 2 (a).

Apart from annotation quality, both OPDReal and OPDMulti datasets are not guaranteed to have high-resolution images with appropriate viewpoints. Image data of both datasets are captured through video frames, so some images have motion blur as seen in Fig 1 (c) row 7, 8 and Fig 2 (c) row 10. Many images in OPDReal and OPDMulti are not captured from clean viewing angles, which makes the target part look obscure, as shown in Fig 1 (c), (d) and Fig 2 (c).

Table 1: Quantity comparison of models pre-trained on synthetic data only. Results on test set of different datasets.

	segm mAP(↑)		
Dataset ↓ / Method →	OPD-C	OPDFormer-C	Ours$_{w/oAL}$
OPDReal	1.6	2.0	**3.0**
OPDMulti	2.3	3.0	**4.2**
Ours	12.7	18.8	**23.4**

In our dataset, images data are captured from photograph and directly labeled in 2D. Hence, we have a rich image dataset quality and contain clear annotations of moveable parts for multiple articulated objects presented in an image. In addition, in the images of our dataset, objects are comprehensively displayed with their main structures fully contained, ensuring that more than just a fragmentary part is visible. We present visualization of individual categories of objects in our dataset in Fig 3.

2. More Visualizations w/o Active Learning

As mentioned in our paper, bridging the synthetic-real data gap in our task is a significant challenge. To address this, we introduce a *coarse-to-fine* active learning approach, specifically designed to achieve high-accuracy segmentation on real images. Table 1 shows the performance of different methods trained exclusively on synthetic data, and test on the test set of different datasets. Despite using extensive training data from rendered images of 3D synthetic models, the accuracy on all real datasets does not exceed 25%. This underscores the necessity and relevance of our work.

In our paper, we assert that the data skewness in OPDReal leads to low overall performance on its test set. From category-wise results on OPDReal test set in Table 2, accuracy on storage category is significantly higher than others, reflecting its predominance with more than 90% of the data. Conversely, on our dataset, the category-wise results are more evenly distributed, as shown in Table 3. The only exception is the oven and microwave category, which exhibits

Table 2: Category-level results on OPDReal test set with training with OPDReal train set.

	segm mAP (↑)			
	Grounded-SAM	OPD-C	OPDFormer-C	Ours$_{w/oAL}$
storage	17.0	70.2	73.9	**79.5**
dishwasher	22.3	43.4	45.8	**49.3**
fridge	22.9	42.4	43.4	**49.6**
oven&micro	9.4	35.1	36.2	**39.5**
washer	10.7	31.4	32.2	**40.2**

Table 3: Category-level results on our dataset, with 80% as training data and 20% as test data.

	segm mAP (↑)			
	Grounded-SAM	OPD-C	OPDFormer-C	Ours$_{w/oAL}$
storage	25.8	76.5	81.4	**89.5**
dishwasher	36.7	78.3	84.0	**93.2**
fridge	25.3	71.5	75.0	**86.8**
oven&micro	16.6	65.3	71.8	**74.1**
washer	3.8	89.4	92.6	**98.0**

lower performance. This is attributed to scenes containing both microwaves and ovens being more complex and typically featuring a variety of different category objects.

We provide additional qualitative results on OPDReal, OPDMulti and ours dataset across different methods in Fig 4, 5 and 6 respectively. We also show qualitative comparison of $Ours_{w/oAL}$ and our final model with *coarse-to-fine* active learning in Fig 7.

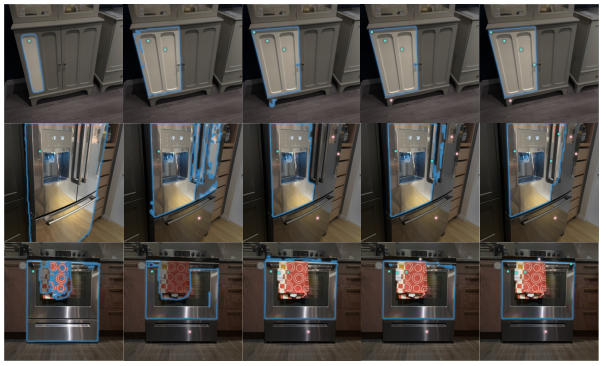
3. More Results with Active Learning

3.1. Comparison with other AL methods

Different from previous papers using AL to improve the segmentation results through giving minimum guidance on correcting the predictions, our work optimizes the entire AL pipeline, especially, the sampling process in a *coarse-to-fine* manner. In this section, we will explain why we did not evaluate our task on other AL methods.

As mentioned in our paper, previous works such as [4, 6] employed AL to refine initial 2D segmentation prediction through key point or region selection. However, [6] use region impurity and prediction uncertainty for domain adaptive semantic segmentation, while we focus on instance segmentation without learning from synthetic data, making them not really comparable. [4] does perform instance segmentation, via *point supervision*, but only a preliminary code was released to "help you understand how it works". Therefore, we cannot use their implementation for a meaningful comparison. Other AL methods aim to close the domain gap, and we explained in the main paper that we cannot borrow such methods on our task due to the large feature differences between current synthetic and our real data.

We notice that Segment Anything Model (SAM) [2] also supports "click" guidance for segmenting parts. In the below figure, we show several examples of such point-supervised segmentation on our data.



In the top row, segmenting moveable parts such as a door from a cabinet often requires multiple "add" and "remove" clicks. However, these clicks are insufficient for segmenting the door of the fridge with light reflections in the middle row. In real-world scenarios, segmentation accuracy from clicks is compromised by scene noise, such as a rag on an oven handle in the bottom row. We show more examples of using click-assisted labeling tool in the supplementary video.

The main reason why AL using point-supervision is difficult in our tasks is that the texture separation around moveable parts is often far less pronounced than those encountered in "bird-in-sky" or "bee-over-flower" segmentation tasks. Therefore, traditional AL methods reducing the labeling efforts cannot apply on our task with a lot of uncertainties (e.g. point selection for best guidance in labeling). All these limitations motivated our *coarse-to-fine* AL approach in optimizing the AL sampling process, instead of the labeling.

3.2. Details of our AL sampling

In our active learning (AL) setup where we use feedback from humans in the loop, one of the major tasks is to judge the quality of annotations output from the model used in the setup. In *coarse* AL, we verify the prediction of interaction direction, which is intuitive to be correct or wrong. In *fine* AL, as mentioned in the main paper, we categorize the outputs into three types, and based on this categorization, images are then moved to respective parts of the pipeline as described below.

- **Perfect:** If the prediction precisely segments all moveable parts of articulated objects in the image, the prediction will be considered as perfect. It will be automatically moved to the training set for the next iteration. Note that not all articulated objects will be segmented in our task – if the object is too small or only partly shown in the image, it will be ignored as they are not meaningful in future downstream tasks.

Table 4: Different data and efficiency statistics over each iteration of the active learning process on our 500-image set using OPDRCNN-C (OPD-C). Columns represents iteration, amount of training data, amount of test data, amount of perfect predictions, amount of bad predictions (numbers of images and annotations), and time required for human efforts in hours (metric total lab time described in the main paper).

Iter.	Train	Test	Perfect	Bad (img/anno)	Time (hr)
0	50	500	-	-	-
1	129	421	17	62 / 211	1.435
2	207	343	36	42 / 143	1.064
3	293	257	46	40 / 145	0.985
4	384	166	52	39 / 128	0.819
5	450	100	32	34 / 108	0.634
6	512	38	35	27 / 95	0.507
7	550	-	21	17 / 57	0.280
Total:				261 / 887	5.724

- **Fair:** If the prediction outputs partial segmentation masks with some minor mistakes (such as holes in the mask, noisy boundary, or wrong prediction on the background/side face of the object), the prediction will be considered as fair. It will remain in the test set for the next iteration. Note that we keep most of outputs as fair aside from perfect since we aim to minimize the amount of bad predictions which require manual annotation.

- **Bad:** If the prediction fails to segment moveable parts or incorrectly segments a non-moveable part (such as combing two moveable parts together, wrong segmentation boundary, or completely missing too many moveable parts), the prediction will be considered as bad. It will be manually annotated and merged to the training set for the next iteration. Note than at every iteration, completely missed predictions (no predictions with $\geq$ 75% confidence) will be automatically selected as bad prediction. In addition, as we aim to minimize the amount of bad predictions to save human efforts, when humans review the predictions, we try to select bad predictions from different scenes and not all bad predictions from the same scenes will be selected.

Fig 8 shows a comparison of perfect, fair and bad predictions based on our model employed in the *fine* AL setup.

3.3. Data and efficiency statistics

We present data and efficiency statistics over each iteration of AL process in annotating segmentation masks for our 500-image set using OPD-C (Table 4), OPDFormer-C (Table 5), Ours$_{f-AL}$ (Table 6) and ours (*coarse-to-fine* AL) (Table 7).

Table 5: Different data and efficiency statistics over each iteration of the active learning process on our 500-image set using OPDFormer-C.

Iter.	Train	Test	Perfect	Bad (img/anno)	Time (hr)
0	50	500	-	-	-
1	256	294	176	30 / 106	0.997
2	339	211	58	25 / 88	0.693
3	398	152	32	27 / 95	0.630
4	437	113	27	12 / 42	0.344
5	471	79	15	19 / 68	0.409
6	505	45	14	20 / 71	0.384
7	550	-	20	25 / 90	0.425
Total:				158 / 560	3.882

Table 6: Different data and efficiency statistics over each iteration of the active learning process on our 500-image set using Ours$_{f-AL}$.

Iter.	Train	Test	Perfect	Bad (img/anno)	Time (hr)
0	50	500	-	-	-
1	340	210	260	30 / 97	0.824
2	445	105	82	23 / 53	0.385
3	550	-	75	30 / 89	0.466
Total:				83 / 239	1.675

Table 7: Different data and efficiency statistics over each iteration of the active learning process on our 500-image set using ours (*coarse-to-fine* AL). Columns represent iteration, amount of training data, amount of test data, amount of correct and wrong (c/w) predictions of interaction direction, amount perfect predictions, amount of bad prediction (numbers of images and annotations), and time required for human efforts in hours.

Iter.	Train	Test	Pose check (c/w)	Perfect	Bad (img/anno)	Time (hr)
0	50	500	-	-	-	-
1	382	168	367 / 133	305	23 / 64	0.964
2	490	60	-	93	15 / 43	0.315
3	550	-	-	38	22 / 64	0.321
Total:					64 / 184	1.6

We observe that OPD-C and OPDFormer-C take 7 iterations in the AL setup, whereas both variants of our model takes only 3 iterations. Among the two competing methods, OPDFormer-C consumes less time over OPD-C, indicating that the predictions from OPDFormer-C are superior to OPD-C. The only difference between these two methods is the inclusion of an additional pose check (interaction direction check) in the first iteration of ours. As a result, our *coarse-to-fine* AL requires more time in iteration 1 for this extra verification step on all 2000 predictions. However, the total time taken by our method is still less than that of Ours$f - AL$.

Table 8: Different data and efficiency statistics over each iteration of the active learning process on our 2000-image set using Ours$_{f-AL}$.

Iter.	Train	Test	Perfect	Bad	Time (hr)
0	550	2000	-	-	-
1	1811	739	1143	118 / 392	3.856
2	2305	245	392	102 / 341	2.242
3	2449	101	102	42 / 156	0.922
4	2550	-	74	27 / 94	0.504
Total:				289 / 983	7.524

Table 9: Different data and efficiency statistics over each iteration of the active learning process on our 2000-image set using ours (*coarse-to-fine* AL)

Iter.	Train	Test	Pose check (c/w)	Perfect	Bad (img/anno)	Time (hr)
0	550	2000	-	-	-	-
1	2073	477	1457/543	1436	87 / 285	3.97
2	2368	182	-	214	85 / 276	1.55
3	2515	35	-	106	41 / 135	0.71
4	2550	-	-	19	16 / 62	0.29
Total:					229 / 758	6.52

We present data and efficiency statistics over each iteration of AL process in annotating segmentation masks for our 2000-image set using Ours$_{f-AL}$ (Table 8) and ours (*coarse-to-fine* AL) (Table 9). We observe that the the benefits of employing the *coarse-to-fine* AL become increasingly apparent when handling extensive amounts of unlabeled data. The *coarse-to-fine* AL reduces the number of bad predictions of final moveable part segmentation in the *fine* stage, and the time for verifying interaction directions is significantly less than that required for manual annotation of bad predictions. This is because the time invested in verifying interaction directions is considerably less than what would be required to annotate bad predictions manually. We provide details of time spent for each operation in Section 3.4.

3.4. User study

We conduct an in-lab user study to (i) obtain timing in verifying and correcting the prediction of interaction direction in the first iteration, (ii) obtain timing in reviewing the final moveable mask segmentation predictions after each iteration, and (iii) obtain timing for manually annotating the bad predictions. We selected 7 users with basic computer operating skills. We provided them with the same tools and instructions as used for the justification of predictions and labeling of moveable parts in the AL process. They were asked to verify the predicted interaction directions from visualization of pose coordinates and value ($\pm x, \pm y, \pm z$), and distinguish perfect, fair and bad predictions from visualization of the final results. Finally, they need to annotate moveable parts for all bad predictions. In our AL setup, each sample image will only be labeled by one user.

We use labelme [5] as the labeling tool. The average times taken by human for different tasks in our AL model are as follows: time to verify and correct the interaction direction $\tau_{chk-dir} = 1.4s$, time to click the corresponding selection button in our interface $\tau_{click} = 0.6s$, time to identify whether the prediction is perfect, fair or bad $\tau_{chk-msk} = 2.4s$, with same time for selection $\tau_{click} = 0.6s$, and average per-annotation labeling time $\tau_{anno} = 14.8s$. We also attach a video in the supplementary material showing the interface of our AL setup for sample selection, correction and annotation.

We show the user interface for verification in both *coarse* and *fine* stage, as well as the manual annotation in Fig 9, 10, 11 separately.

References

[1] Hanxiao Jiang, Yongsen Mao, Manolis Savva, and Angel X Chang. Opd: Single-view 3d openable part detection. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIX*, pages 410–426. Springer, 2022.

[2] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023.

[3] Xiaohao Sun, Hanxiao Jiang, Manolis Savva, and Angel X. Chang. OPDMulti: Openable part detection for multiple objects. In *Proc. of 3D Vision*, 2024.

[4] Chufeng Tang, Lingxi Xie, Gang Zhang, Xiaopeng Zhang, Qi Tian, and Xiaolin Hu. Active pointly-supervised instance segmentation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVIII*, pages 606–623. Springer, 2022.

[5] Kentaro Wada. labelme: Image polygonal annotation with python. `https://github.com/wkentaro/labelme`, 2018.

[6] Binhui Xie, Longhui Yuan, Shuang Li, Chi Harold Liu, and Xinjing Cheng. Towards fewer annotations: Active learning via region impurity and prediction uncertainty for domain adaptive semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8068–8078, 2022.

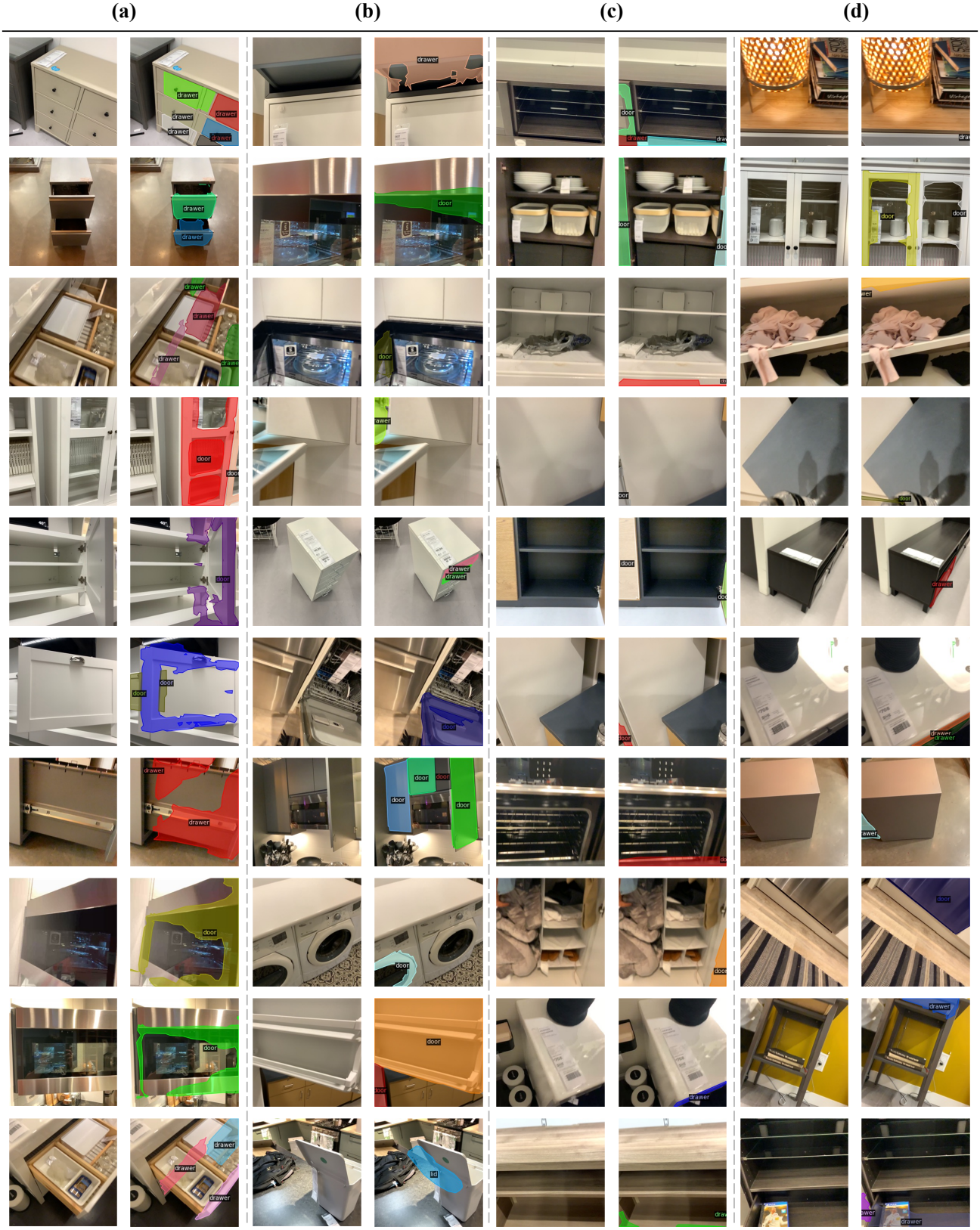


Figure 1: Visualizations of dataset annotations of OPDReal.

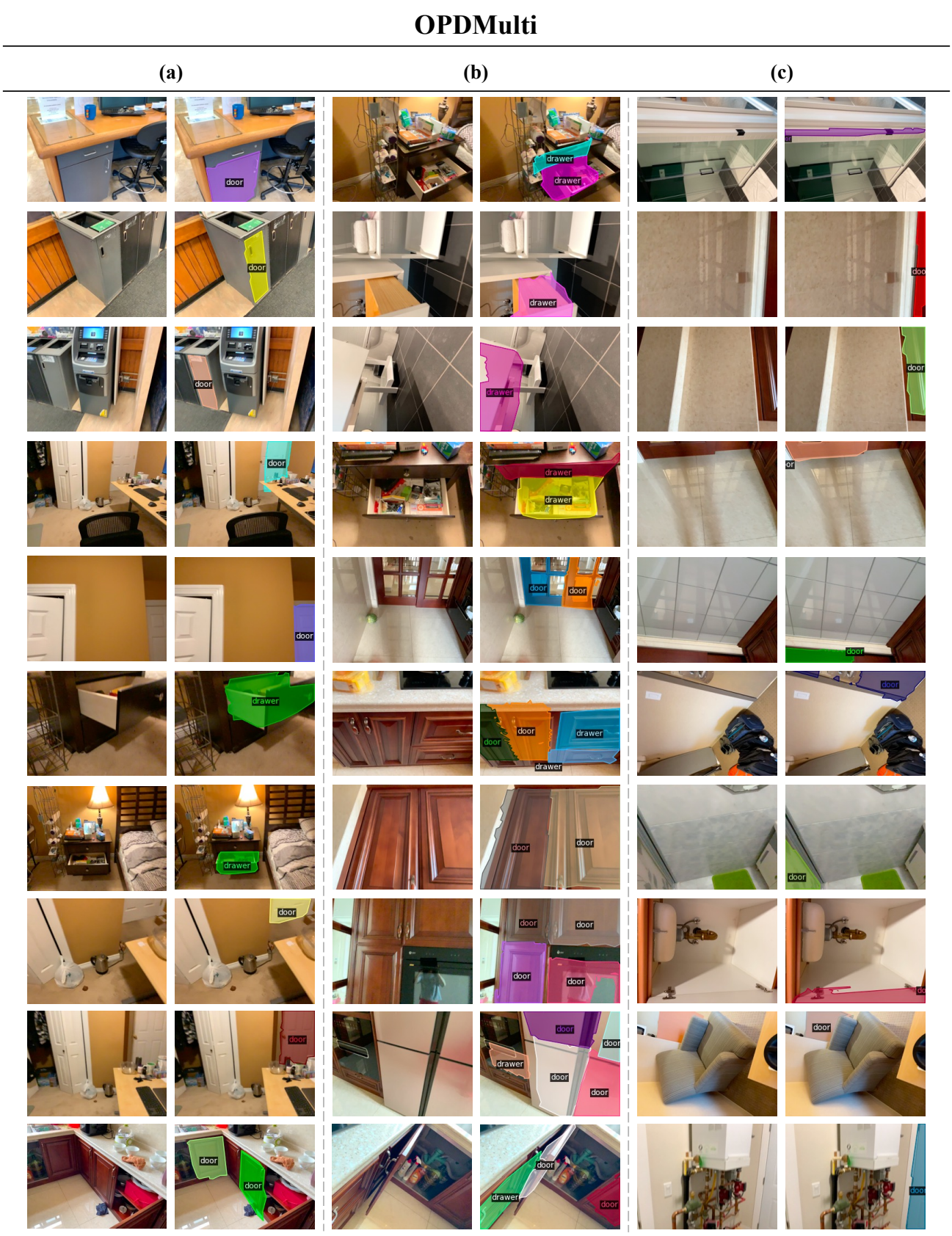


Figure 2: Visualizations of dataset annotations of OPDMulti.

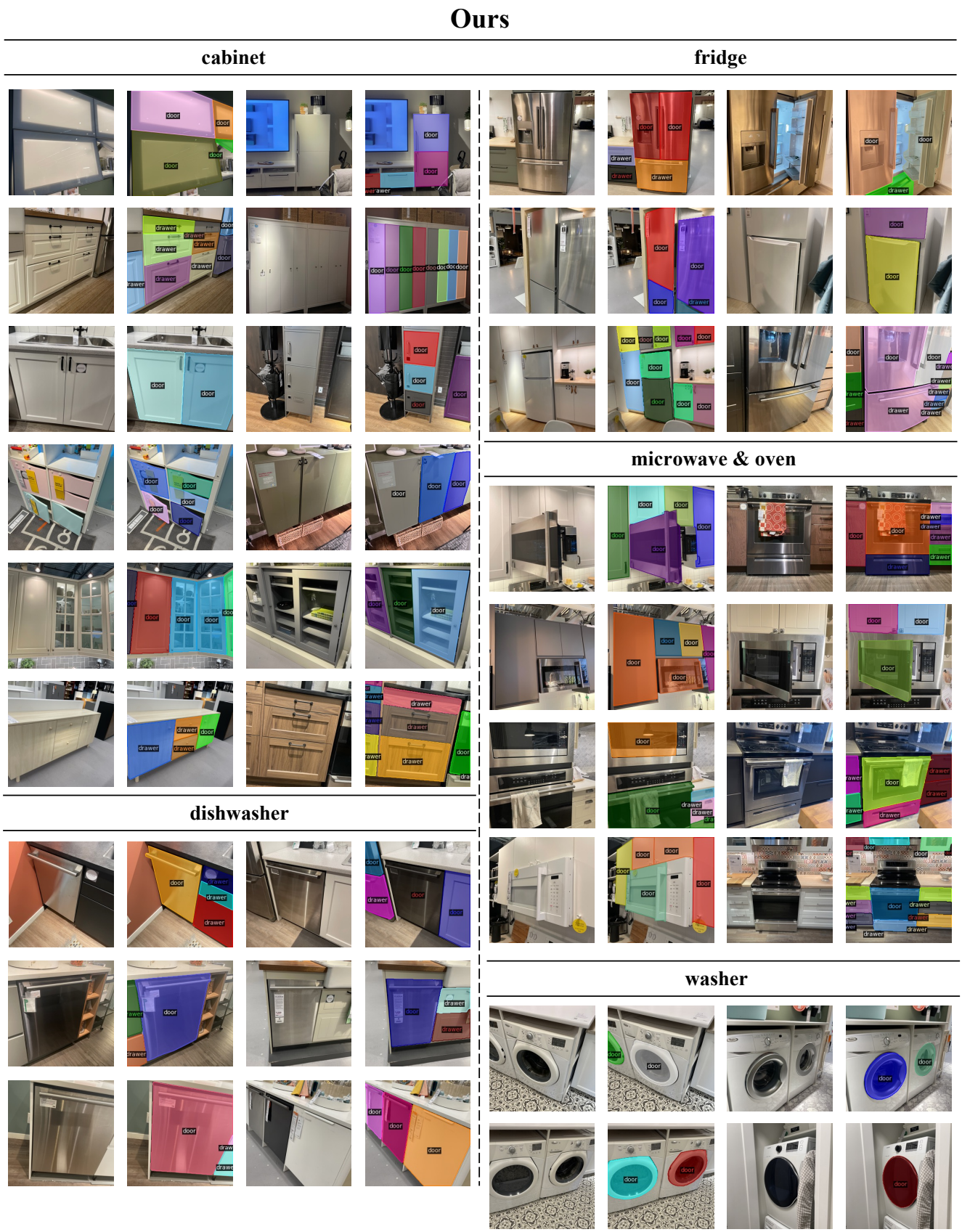


Figure 3: Visualizations of dataset annotations of our dataset.

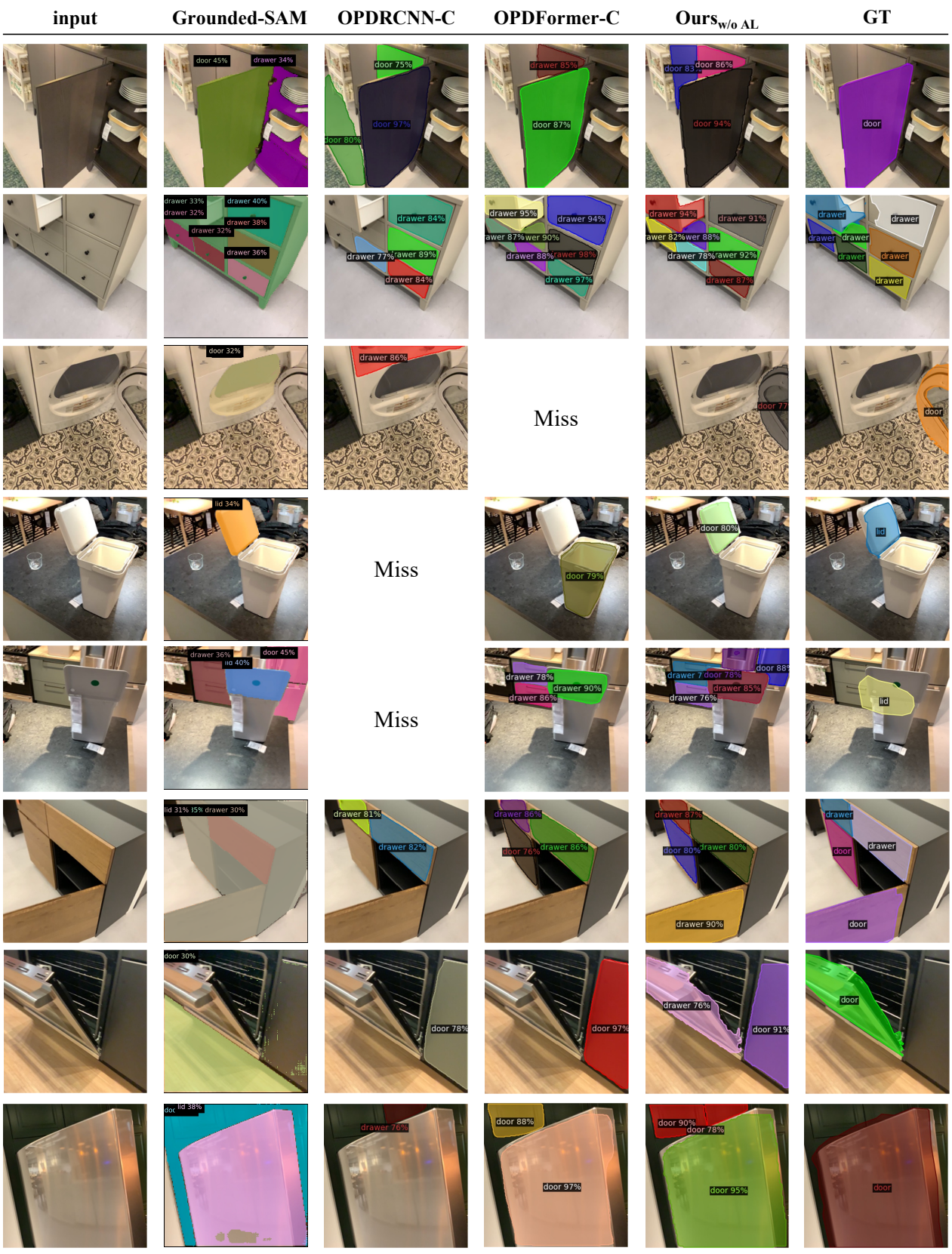


Figure 4: Qualitative results of different methods on OPDReal test set. "Miss" represents the absence of part segmentation with $\geq 75\%$ confidence.

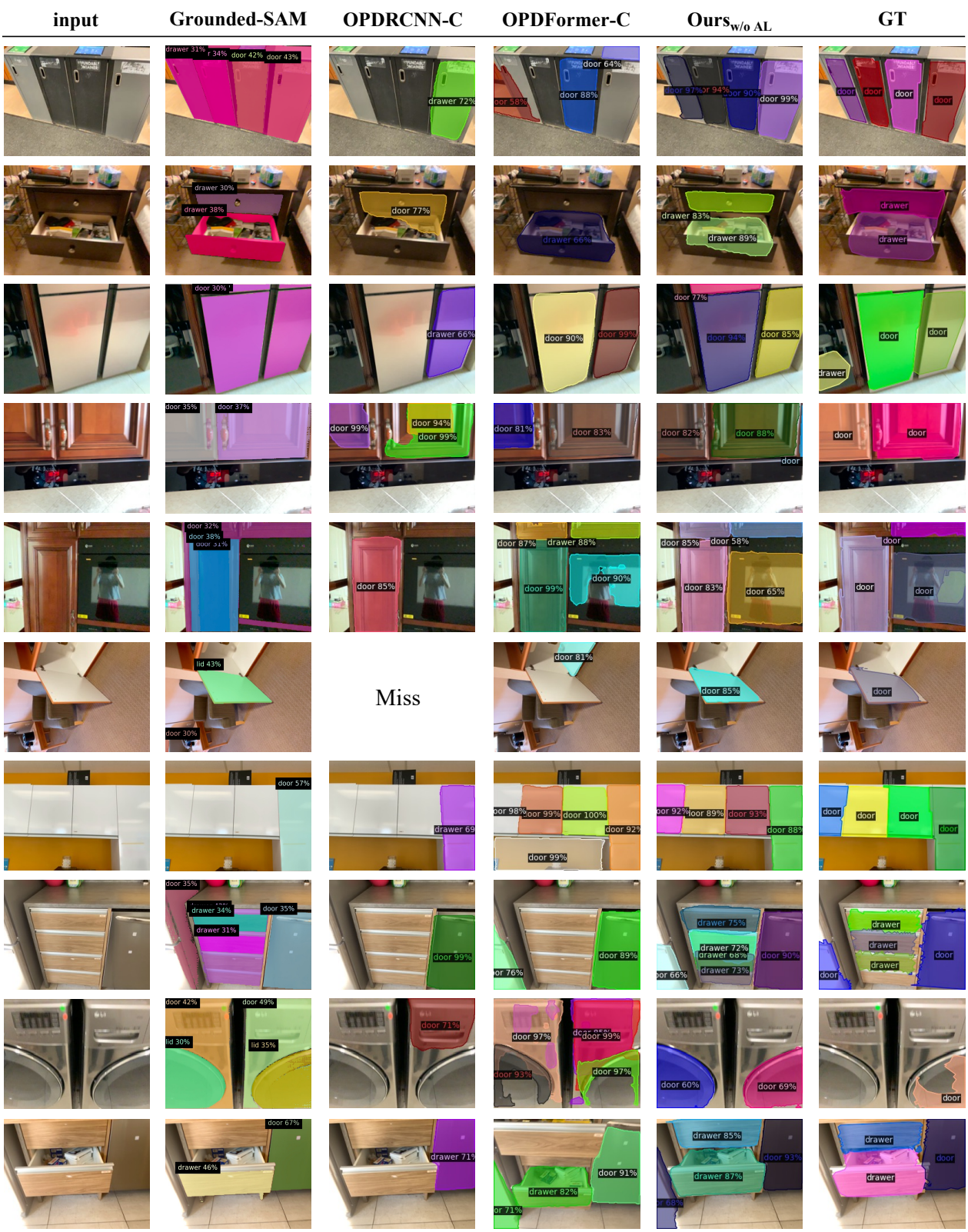


Figure 5: Qualitative results of different methods on OPDMulti test set. "Miss" represents the absence of part segmentation with $\geq$ 65% confidence. Note than we lower the confidence threshold for OPDMulti as most scores for predictions are below 75%.

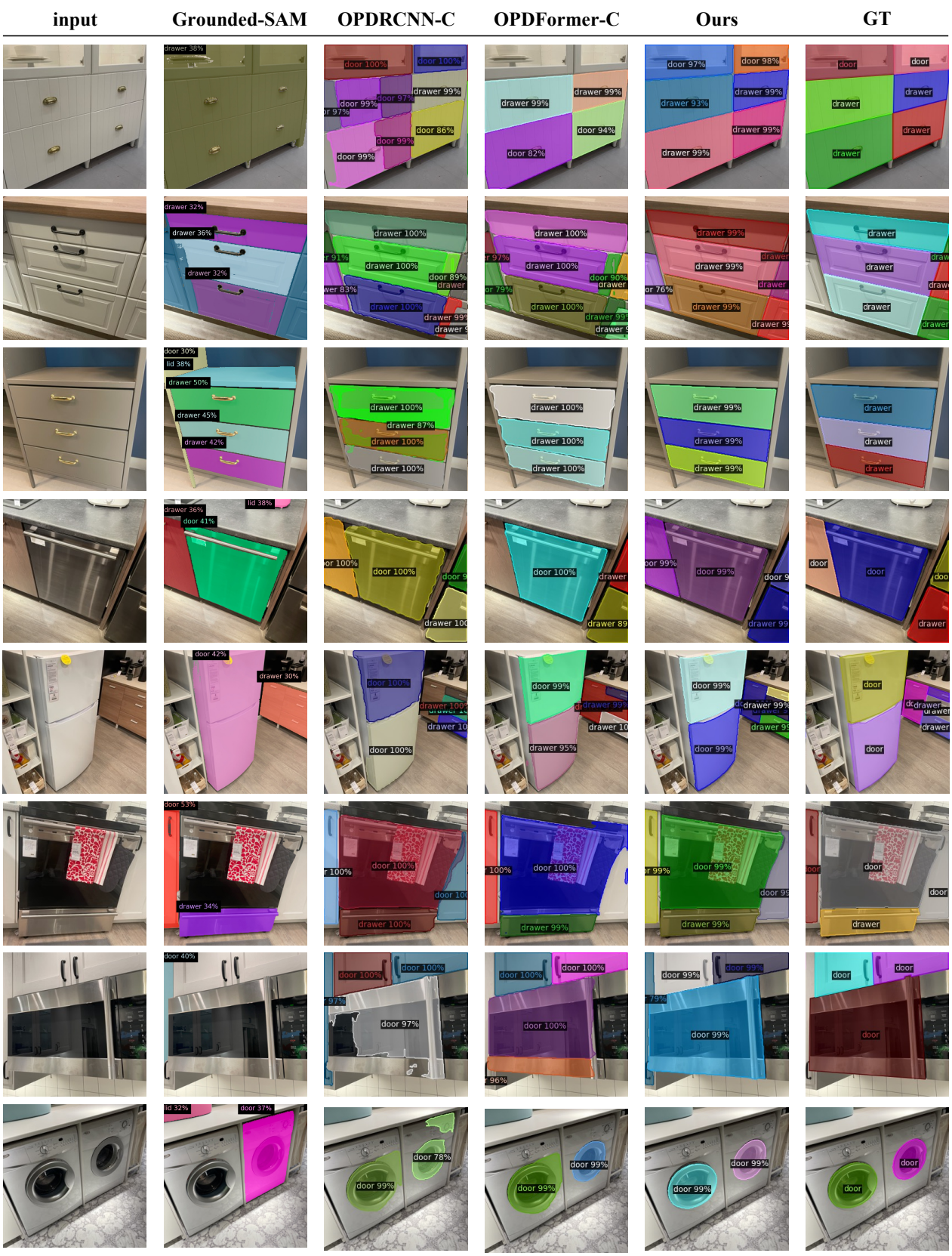


Figure 6: Qualitative results of different methods on our test set.

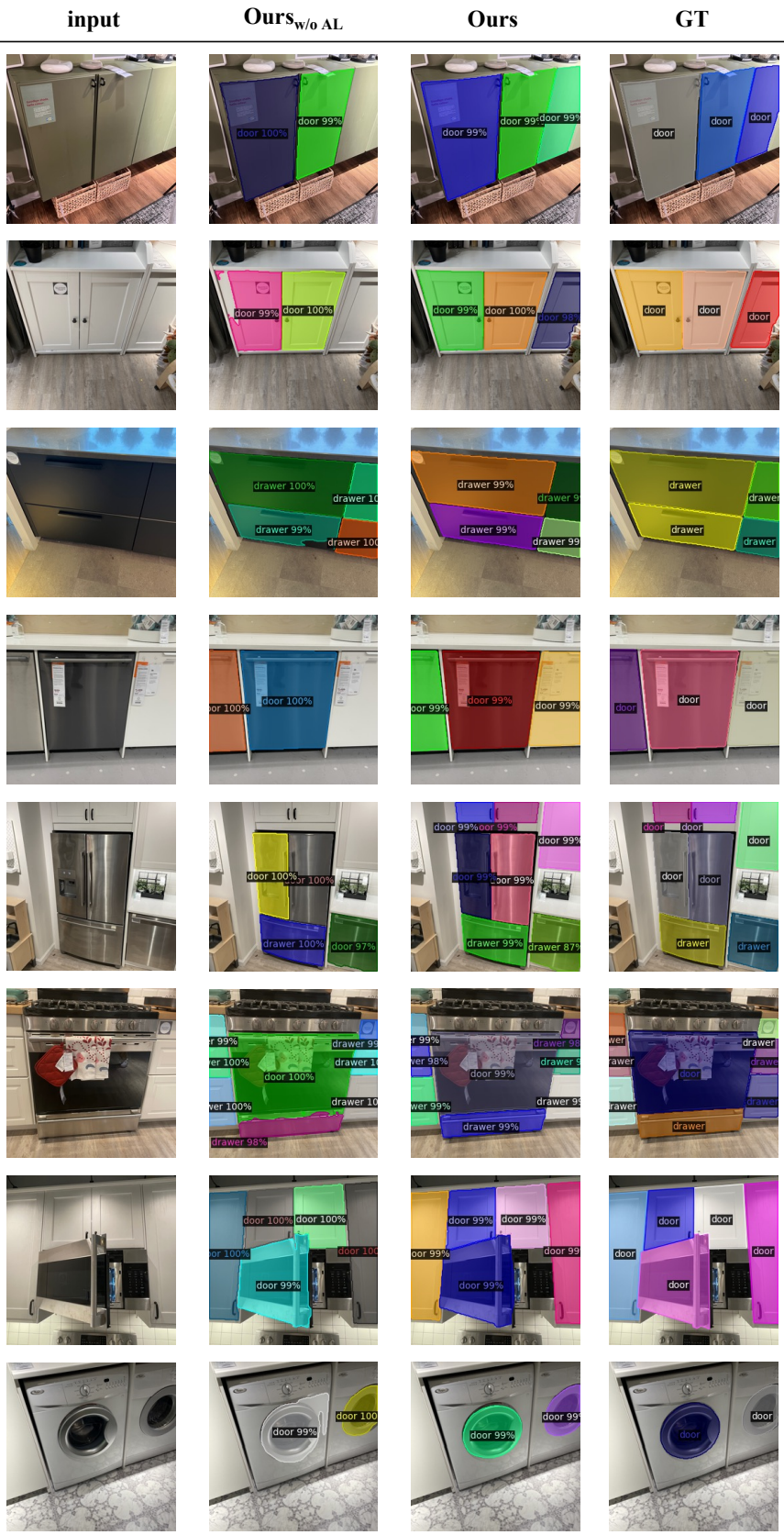


Figure 7: Qualitative results of Ours$_{w/oAL}$ and our final model on our test set.

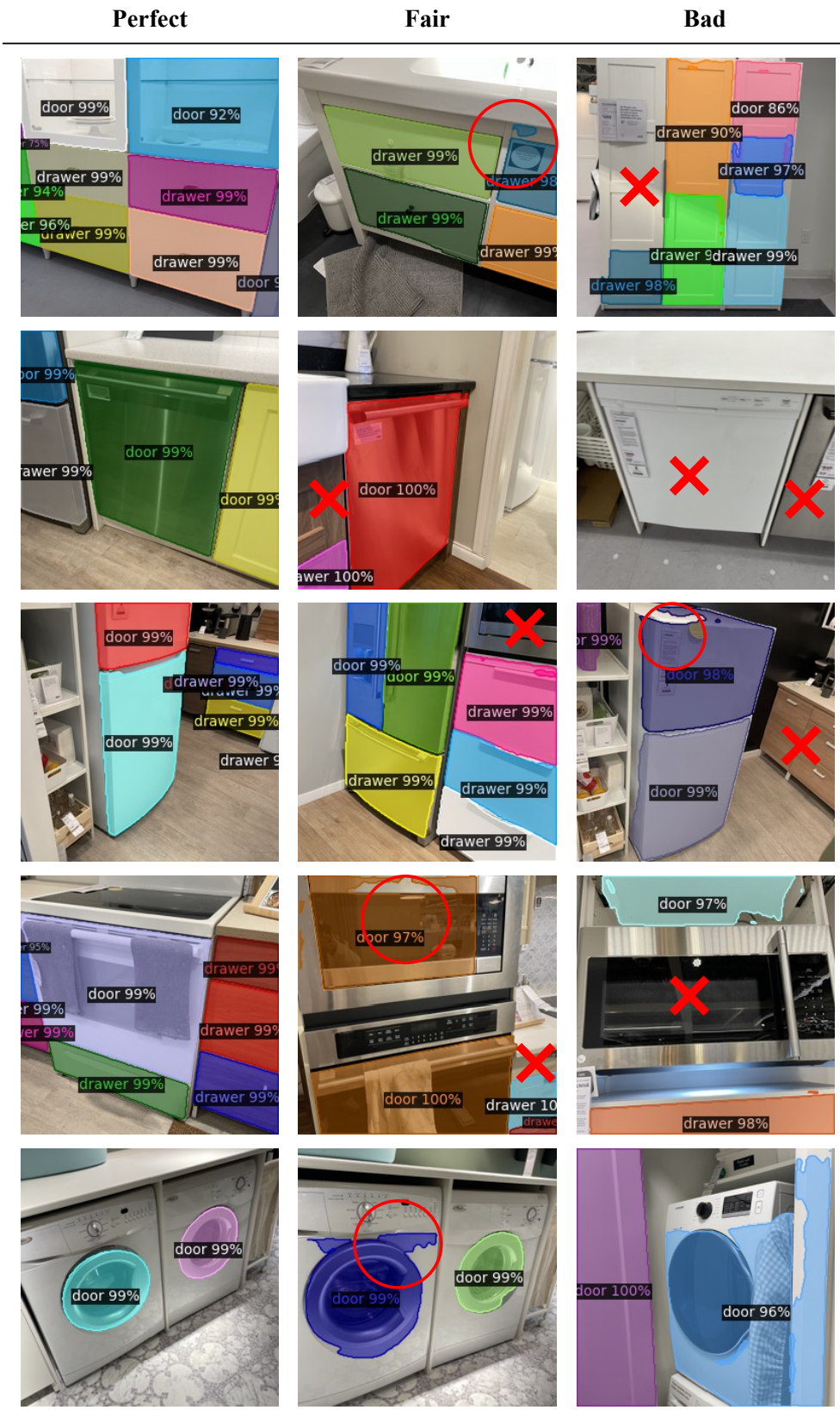


Figure 8: Comparison of perfect, fair, and bad predictions. We use red circle to highlight those segmentation with minor mistakes(small holes or noisy boundaries) and red $\times$ to indicate erroneous or missed segmentation.

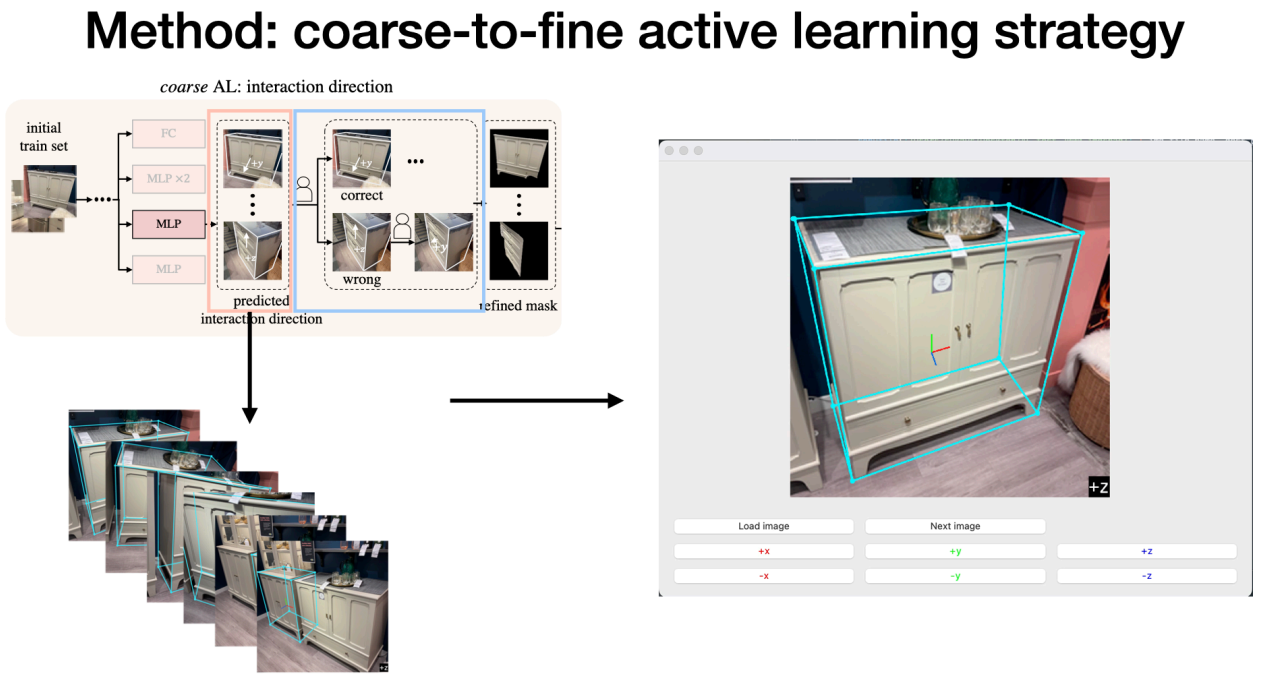


Figure 9: Pipeline and user interface of the coarse-stage verification.

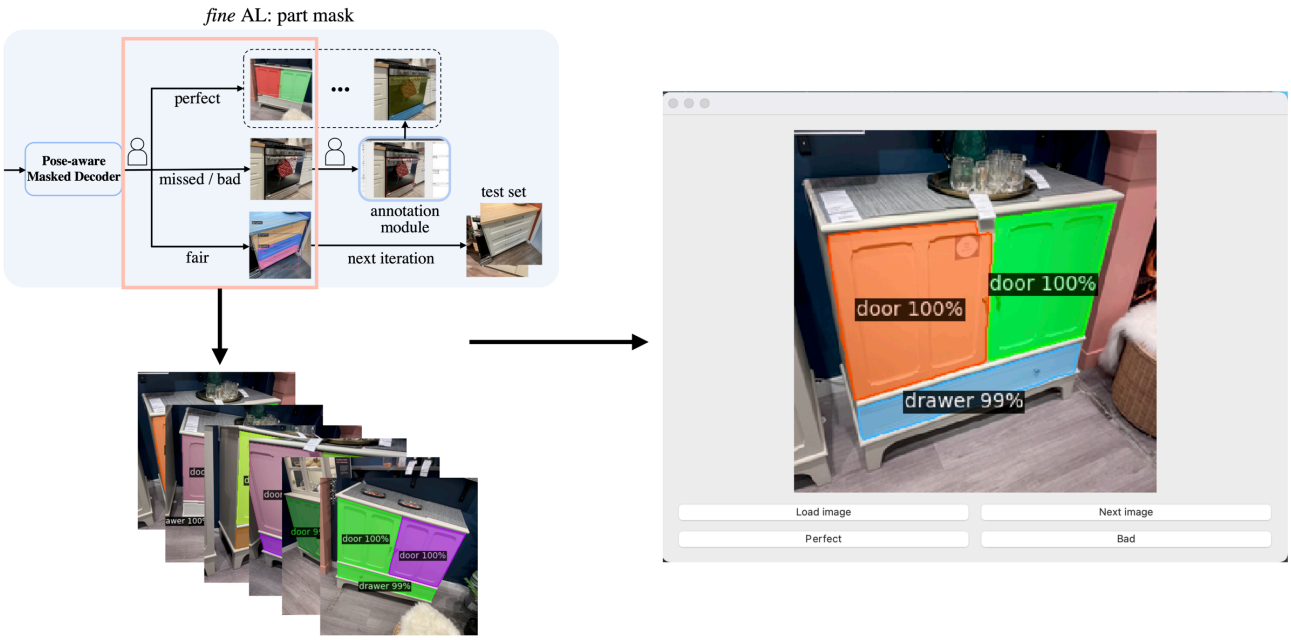


Figure 10: Pipeline and user interface of the fine-stage verification.

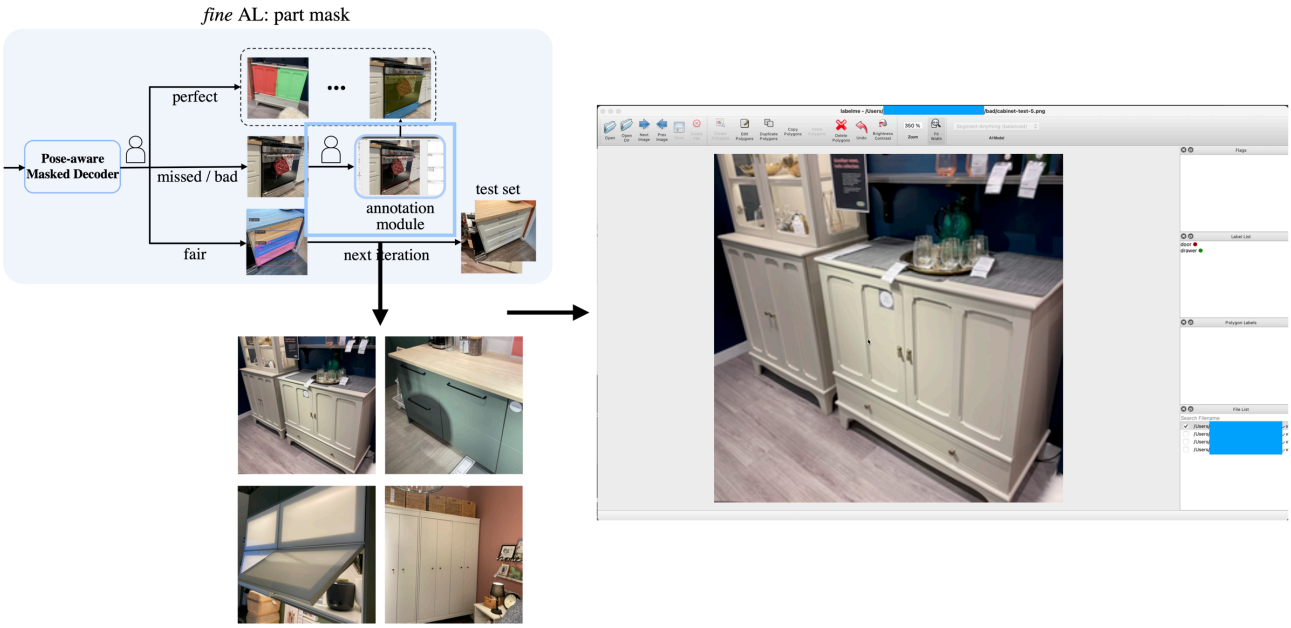


Figure 11: Pipeline and user interface of the human annotation.