PartSTAD: 2D-to-3D Part Segmentation Task Adaptation — Supplementary Material

Hyunjin Kim[1†] and Minhyuk Sung[2]

[1] KRAFTON Inc., South Korea
[2] KAIST, South Korea
rlaguswls98@krafton.com, mhsung@kaist.ac.kr

In this supplementary material, we present additional implementation details (Sec. S.1), additional results on the ablation studies (Sec. S.2, Sec. S.3, Sec. S.4, Sec. S.5, and Sec. S.6), how bounding box weights change compared to GLIP confidence scores (Sec. S.7), the reason for not replacing GLIP with SAM (Sec. S.8), more analysis on results (Sec. S.9), outcomes with scanned data (Sec. S.10), and comprehensive results covering all PartNet-Mobility categories (Sec. S.11, Sec. S.12, and Sec. S.13).

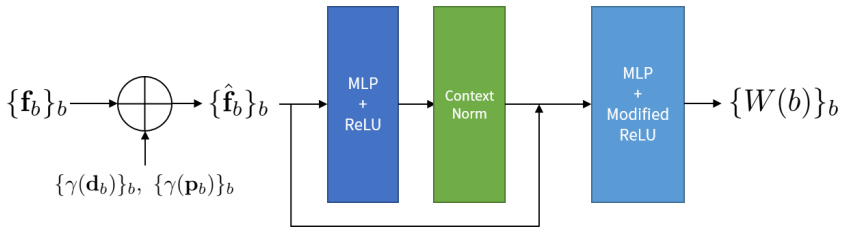


Figure S1: Architecture of the weight prediction network used in PartSTAD. $\mathbf{f}_b$, $\mathbf{d}_b$, and $\mathbf{p}_b$ represent the bounding box feature, view direction, and position in the 2D image of the bounding box b, respectively. γ denotes the positional encoding function. This network takes all bounding box features of a single object as input and outputs bounding box weights.

S.1 Additional Implementation Details

Network Architecture. Fig. S1 shows the detailed network architecture of the weight prediction network of PartSTAD. As mentioned in Sec. 4.2 of the main paper, the network consists of small shared two-layer MLP with the context normalization [10] layer between them to embed context information. This light architecture design is inspired by LoRA [1] and PartSLIP [5] which add small learnable parameters while keeping the original pretrained model parameters.

We add positional encoded vectors to each bounding box feature to embed the positional information. For given bounding box feature $\mathbf{f}_b$, we do not directly

[†] This work was conducted when the author was at KAIST.

feed $\mathbf{f}_b$ to the network but feed $\hat{\mathbf{f}}_b$ which concatenates $\mathbf{f}_b$, positional encoding of view direction $\mathbf{d}_b$, and positional encoding of 2D bounding box position $\mathbf{p}_b$. Positional encoding γ is defined as below:

$$\gamma(x_1, x_2, ..., x_n) = \bigoplus_{i=1}^{n} (x_i, \sin(2^0 \pi x_i), \cos(2^0 \pi x_i), ..., \sin(2^{L-1} \pi x_i), \cos(2^{L-1} \pi x_i)), \tag{1}$$

where $\oplus$ indicates the concatenation operation and L is set to 10 in our experiments. Thus the input $\hat{\mathbf{f}}_b$ of the weight prediction network can be written as below:

$$\hat{\mathbf{f}}_b = \mathbf{f}_b \oplus \gamma(\mathbf{d}_b) \oplus \gamma(\mathbf{p}_b). \tag{2}$$

We initialize all network parameters $\theta_i \sim \mathcal{N}(0, \epsilon)$, where ϵ is a very small number (ϵ is set to 0.0001 in our experiments) so that the initial MLP output becomes 0. Since the last layer is modified ReLU layer ϕ (Eq. 7 of the main paper), the initial weight is set to τ. This is inspired by zero convolution of ControlNet [11], and this makes the training more stable by preventing drastic weight changes.

Instead of using an attention-based network to consider the relations between bounding boxes, we opt to add context normalization [10] between two MLP networks in the weight prediction network. This allowed us to keep the network lightweight while still considering the relations between bounding boxes.

Design of Modified ReLU. The modified ReLU function (Eq. 7 of the main paper) is designed to set the initial value of the bbox weight, which is the output of the weight prediction network, to a value $\tau > 0$.

Training Details. We generate 2D images of size 800×800 from 10 fixed viewpoints for each object that is normalized to fit in a unit sphere using the Pytorch3D point cloud renderer with the fixed camera distance 2.2 following the same procedure as described in PartSLIP [5]. After rendering, we obtain bounding boxes for each image using the GLIP [4]. Subsequently, all bounding box features corresponding to bounding boxes from a single object are simultaneously fed into the weight prediction network to calculate the weights. Training is conducted using a single RTX 3090 GPU.

S.2 Results with Varying Parameters

Number of Views. Tab. S1 presents the results of the ablation study on the number of views. It is observed that as the number of views increases, mIoU also increases, with the most significant difference observed when the view changes from 5 to 10. This highlights that having too few samples of bounding boxes used in training can lead to suboptimal results.

Table S1: Ablation study on the number of views.

# of Views	5	10	15	20
mIoU	59.4	65.0	66.3	**67.2**

Table S2: Ablation study on the number of training data. The experiment is conducted on the StorageFurniture category as it only has more than 128 shapes (346 in total).

# of Training Data	8	16	32	64	128
mIoU	57.0	56.7	58.0	57.8	**60.1**

Table S3: Ablation study on the hyperparameter τ. The mIoU is measured for five object categories: Chair, Table, StorageFurniture, Faucet, and TrashCan, which are the five categories with the most test data.

Initial Weight (τ)	1	5	10	15	20
mIoU (5 categories)	54.8	55.6	**55.8**	55.6	55.6

Number of Training Data. Tab. S2 presents the results of the ablation study based on the number of training data. Only the StorageFurniture category has more than 128 data, so the experiment is conducted only for this category. There is a tendency for mIoU to increase as the number of training data increases, but the difference is not significant. This demonstrates that even using only 8 data points can yield sufficiently good results.

Hyperparameter τ. Tab. S3 shows the ablation results for the hyperparameter τ. The results are best when τ is 10, but they also demonstrate that the τ value does not significantly impact the results when it is greater than zero.

S.3 Random Viewpoints vs. Fixed Viewpoints

Table S4: Comparison between cases of rendering with random viewpoints and fixed viewpoints.

Category	Chair	Kettle	Lamp	Suitcase	Scissors
Random	**85.5**	82.5	66.8	**71.3**	64.8
Fixed (Ours)	85.3	**84.2**	**68.4**	68.3	**68.5**

Tab. S4 shows the results with random viewpoints and shows that the outcomes are not sensitive to the choice of viewpoints.

S.4 Cross-Entropy Loss vs. mRIoU Loss

Table S5: Comparison with the cases of using cross-entropy loss and mRIoU loss (Ours).

Method	mIoU (%)
PartSLIP [5] + *SAM Mask Integration* (Baseline)	61.9
PartSTAD + Cross-Entropy - mRIoU	63.5
PartSTAD + Cross-Entropy	64.5
PartSTAD (Ours)	**65.0**

As mentioned in Sec. 4.1 of the main paper, the use of mRIoU loss is crucial for achieving significant improvement in our task adaptation. To demonstrate the effectiveness of our mRIoU loss, we conduct an experiment comparing it with the alternative, cross-entropy loss.

Tab. S5 shows the ablation results for different loss types. Baseline at the 1st row in Tab. S5 represents the result which only applies *SAM* [3] *mask integration* to PartSLIP [5] (same as our method *without weight prediction*). When using the commonly used cross-entropy loss for segmentation tasks, the mIoU decreases by 1.5%p compared to using the mRIoU loss. Even when both losses are used together, the mIoU decreases by 0.5%p. This indicates that the mRIoU loss is more effective for 3D segmentation task adaptation, and it shows its highest effectiveness when used alone.

S.5 Vanilla GLIP vs. Finetuned GLIP

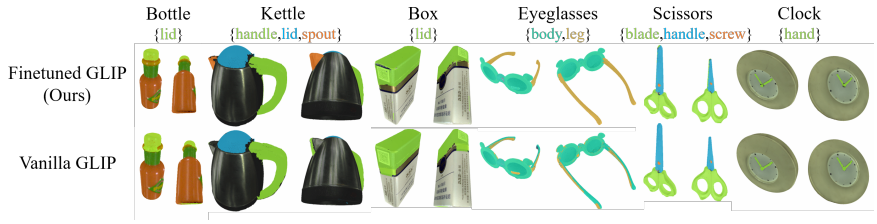


Figure S2: Qualitative comparison between vanilla GLIP and finetuned GLIP on OmniObject3D [8], a real-scanned dataset.

Tab. S6 compares results using vanilla GLIP and finetuned GLIP. Vanilla GLIP yields significantly worse results, emphasizing the substantial impact of

Table S6: Quantitative comparison between vanilla GLIP and finetuned GLIP on PartNet-Mobility [9] dataset.

Method	Vanilla GLIP	Finetuned GLIP (Ours)
PartSLIP [5]	27.2	58.0
PartSTAD (Ours)	48.9 (+ 21.7)	**65.0 (+ 7.0)**

bounding box prediction on the final outcome. At the same time, when we use the vanilla GLIP, our weight prediction network significantly improves the mIoU from 27.2 to 48.9. This indicates that our weight prediction network is more effective when the 2D prediction is inaccurate.

In our experiments with scanned objects (Sec. 5.4 of the main paper), we used a finetuned GLIP instead of the vanilla GLIP, as it exhibited better performance in detecting the parts, even for real images, due to its finetuning for the specific parts. Fig. S2 illustrates the superior performance of the finetuned GLIP compared to the vanilla GLIP for the OmniObject3D dataset.

S.6 GLIP Confidence Score vs. Mask Weight

Table S7: Comparison with the cases of using GLIP confidence score as weight and predicted mask weight (Ours).

Method	mIoU (%)
PartSLIP [5] + *SAM Mask Integration* (Baseline)	61.9
PartSTAD + GLIP Conf.	53.3
PartSTAD + Normalized GLIP Conf.	62.3
PartSTAD (Ours)	**65.0**

It is worth noting that the GLIP [4] model also outputs a confidence score for each predicted bounding box. This implies that we can consider using the GLIP confidence scores as weights in the voting scheme ($W(b)$ in Eq. 5 of the main paper). Thus, we compare the results when using GLIP confidence scores and the predicted mask weights from our method.

Tab. S7 presents the comparison results between using GLIP confidence as weights and using weights predicted from the network. We compare those methods with the baseline (1st row in Tab. S7) which only applies *SAM* [3] *mask integration* to PartSLIP [5] (same as our method *without weight prediction*). As the confidence scores from GLIP range in $[0, 1]$, utilizing them directly as weights causes an overall score reduction, resulting in segments are not generated. Consequently, the outcome is notably poor, with a 53.3 mIoU. To ensure a

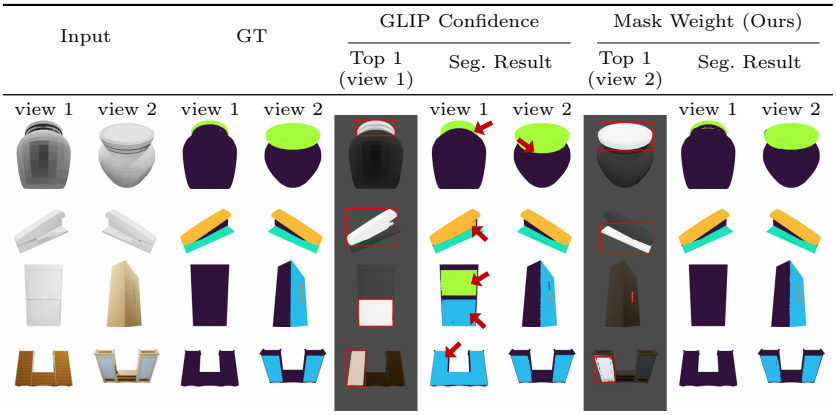


Figure S3: Comparison of results using GLIP confidence scores and Mask Weights. The 5th and the 8th columns depict masks with the highest scores (weights), where red rectangles represent bounding boxes from GLIP, and the white regions are segmentation masks after integrating SAM. From the top row to the bottom, each corresponds to the Bottle, Stapler, and two StorageFurniture categories. When using GLIP confidence, the highest score mask for Bottle (1st row) and Stapler (2nd row) includes an incorrect region, leading to inaccurate segmentation (denoted as red arrows). In contrast, our method assigns the highest score to the correct mask, indicating that the incorrect mask has a lower score. Additionally, when using the GLIP confidence score, the highest score mask for the 3rd and the 4th rows each indicates a completely wrong part (the backside of StorageFurniture). However, our method assigns the highest score to the handle and the correct door part at the front side for the 3rd row and the 4th row, respectively.

fair comparison, we normalize the weights to maintain the same sum as before. With these normalized weights, as presented in the second row of Tab. S7, the result becomes 62.3 mIoU, a slight increase of 0.4%p compared to the baseline. However, this is still 2.7%p lower than utilizing predicted mask weights from a network trained with 3D mRIoU loss. In conclusion, our method provides results more optimized for 3D segmentation than GLIP confidence scores, demonstrating the effectiveness of our method.

As shown in Fig S3, GLIP confidence scores occasionally assign the highest score to incorrect bounding boxes, leading to suboptimal segmentation results. In contrast, the weights predicted by our method consistently assign the highest weight to the correct regions. For instance, using the GLIP confidence score as the weight results in the highest score masks for Bottle (1st row) and Stapler (2nd row) including incorrect regions, leading to inaccuracies in segmentation (indicated by red arrows). In contrast, our method assigns the highest score to the correct mask, indicating that the incorrect mask has a lower score. Additionally, with the GLIP confidence score, the highest score masks for the 3rd and the 4th rows each indicate completely wrong parts (the backside of StorageFurniture). However, our method assigns the highest score to the handle and the correct door part at the front side for the 3rd row and the 4th row, respec-

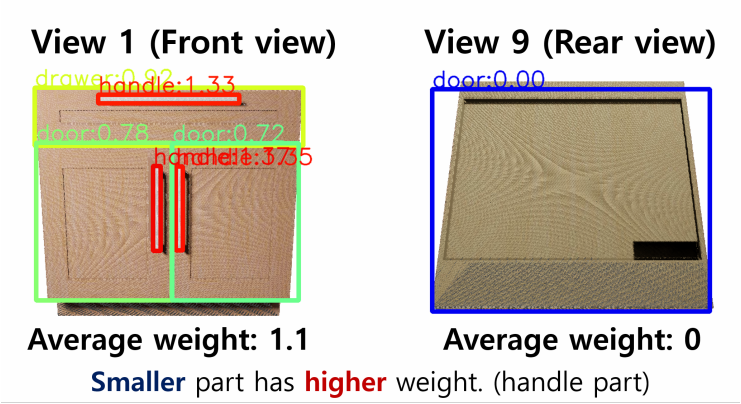


Figure S4: The smaller part (handle) has a higher weight compared to the bigger part (door), and the rear view (fewer GT parts) has a lower weight compared to the front view (more GT parts).

tively. Those results demonstrate that utilizing the mask weight predicted from the network trained with 3D mRIoU loss produces more accurate predictions compared to using the GLIP confidence score as the weight.

S.7 Learned Bounding Box Weights

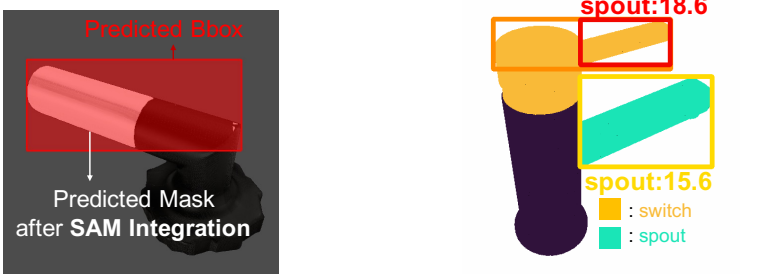
As seen in the example of Fig. S4 for the storage furniture category, smaller parts like handles tend to have relatively higher weights compared to larger parts like doors. Additionally, in rear views where there are no ground truth parts, there is a tendency for the average weight to be lower compared to front views with many ground truth parts. This indicates that learned weight is influenced by both view direction and part labels, unlike the GLIP confidence score, which has a uniform average without distinct trends regarding view and parts.

S.8 Reasons for Not Replacing GLIP [4] with SAM [3]

SAM [3] allows text prompts as inputs, which could enable direct replacement of GLIP with SAM. However, since the pretrained model supporting text prompts has not been released, we resort to an alternative approach. We serialize GLIP and SAM by using a bounding box predicted by GLIP as an input prompt for SAM.

Note that Grounded-SAM [7] and other recent 2D segmentation methods based on text prompts (e.g., SAM-HQ [2]) also involve the serialization of a bounding box prediction network (like GLIP) and SAM. Grounded-SAM specifically uses GroundingDINO [6] instead of GLIP. We believe that the selection of the bounding box prediction network should not impact our contributions.

S.9 More Analysis on Results



(a) Failure case of SAM Integration. **(b)** Failure case of Weight Prediction.

Figure S5: Failure cases of PartSTAD.

The qualitative and quantitative results in the main paper demonstrate that our PartSTAD provides more specialized 2D predictions tailored to 3D segmentation compared to PartSLIP [5]. However, in some categories, there are cases where removing specific components from PartSTAD leads to better results or even where PartSLIP outperforms PartSTAD (e.g., Faucet category). Fig. S5 illustrates cases where each component performs worse predictions than the baseline.

When the initially given bounding box contains few wrong points but includes many correct points, there are cases where the new mask obtained through SAM does not include the previously contained correct points (Fig. S5a). In such cases, the performance may deteriorate when *SAM mask integration* is applied. In the Faucet category, both the switch part and the spout part protrude prominently, resulting in the initial bounding box containing few irrelevant points. Therefore, it appears that the performance deteriorates when correct points are excluded rather than irrelevant points through SAM mask integration.

Secondly, in visually similar parts, weights might be inaccurately predicted. Fig. S5b illustrates the predicted bounding boxes and weights of the Faucet object, showing that the switch part is predicted as the spout with the highest bounding box weight. In such cases where parts are not visually distinguishable, the weight prediction may not be properly learned. Additionally, adding weight prediction in these cases may lead to a decrease in performance.

Note that such cases are rare and do not significantly impact the overall improvement, as shown in Tab. 1 and Tab. 2 in the main paper.

Additionally, for some parts such as door and drawer sometimes have extremely low IoU. This is mainly caused by GLIP, as it fails to detect the parts due to a lack of data. For example, in the Table class, there is no training data that includes the door part.

Figure S6: Qualitative results on real-world scan data. In the highlighted red circle, it is evident that our method achieves more accurate segmentation than PartSLIP [5].

S.10 Results on Real-World Scanned Data

Fig. S6 illustrates the results of semantic segmentation on the real-world scan data used in PartSLIP [5], which is captured by smartphone. As seen in the figure, our method demonstrates its robustness by successfully predicting not only with higher-quality real-world scans like OmniObject3D [8] as illustrated in Fig. 5 of the main paper but also with lower-quality scan data. Also, our method provides more accurate segmentation than PartSLIP [5]. In the case of the Chair, our method accurately segments the arm part, while PartSLIP fails to do so. For the Kettle, our method better identifies the spout compared to PartSLIP. Additionally, in the cases of StorageFurniture and TrashCan, PartSLIP segments

parts that should not be segmented (the backside of StorageFurniture and the lid of TrashCan). On the other hand, for the KitchenPot, while PartSLIP finds the lid part that our method misses, its boundary is still not perfect. We demonstrate that our approach identifies more accurate parts while simultaneously predicting more precise boundaries.

S.11 Complete Quantitative Results of PartSTAD

Tab. S8, Tab. S9, and Tab. S10 show the full table of quantitative results for semantic segmentation, part-aware instance segmentation, and part-agnostic instance segmentation, respectively. Overall, our method demonstrates the best results across whole categories and parts. Please refer to the complete table on the subsequent page for comprehensive information. Moreover, after quantitative result tables, additional qualitative results are illustrated in Sec. S.12 and Sec. S.13.

Table S8: Full table of semantic segmentation results on the PartNet-Mobility [9] dataset.

		Baselines			Ablations		
Category	Part	SATR	SATR +SP	PartSLIP	*w/o Weight Pred.*	*w/o SAM Integ.*	**Ours**
Bottle	lid	50.0	55.5	80.7	**85.9**	81.0	83.6
Box	lid	50.6	54.9	77.9	**84.3**	82.8	81.1
Bucket	handle	6.6	7.5	21.1	**85.4**	52.4	83.6
Camera	button	10.2	20.8	46.1	48.0	**53.3**	51.6
	lens	24.1	27.5	**79.1**	77.8	76.6	77.1
Cart	wheel	30.8	37.2	78.8	**85.4**	81.2	85.0
Chair	arm	22.1	24.4	65.4	68.8	63.5	**69.7**
	back	52.3	59.4	**88.8**	88.0	88.2	87.5
	leg	39.8	45.2	90.8	93.1	92.1	**93.9**
	seat	39.1	46.6	78.8	**83.1**	78.4	82.7
	wheel	12.1	12.8	90.5	91.9	**93.4**	92.8
Clock	hand	3.9	16.1	40.5	**54.3**	46.2	47.4
Coffee Machine	button	2.9	3.6	5.9	5.9	5.8	**6.3**
	container	41.5	44.9	**54.3**	52.0	50.6	44.8
	knob	5.6	9.9	28.8	29.0	30.6	**31.4**
	lid	30.2	34.7	59.9	59.4	**61.6**	60.7
Dish-washer	door	21.3	20.7	68.6	**70.3**	67.2	68.1
	handle	4.6	16.9	49.5	52.4	51.3	**53.0**
Dispenser	head	34.5	36.3	55.2	56.6	59.1	**63.2**
	lid	42.0	46.4	**86.4**	84.4	84.9	84.2
Display	base	26.8	33.9	96.7	**97.0**	95.4	96.8
	screen	35.4	40.0	70.3	73.1	**76.3**	70.7
	support	19.4	28.0	78.6	**80.8**	72.1	79.4
Door	frame	14.3	15.1	19.5	20.0	58.4	**61.0**
	door	55.2	61.4	68.7	72.4	68.9	**74.4**
	handle	5.9	15.3	47.3	48.2	47.4	**48.9**
Eyeglasses	body	69.9	78.5	88.2	86.8	**95.9**	95.8
	leg	57.1	69.8	90.2	**90.6**	88.3	89.1
Faucet	spout	43.4	46.2	75.3	75.2	74.0	**75.3**
	switch	30.9	33.4	**57.1**	55.7	54.1	55.3
Folding Chair	seat	58.1	63.7	83.6	90.4	90.4	**91.6**
Globe	sphere	83.1	83.5	93.0	**98.1**	95.7	93.5
Kettle	lid	48.3	55.1	71.8	83.9	**89.4**	84.1
	handle	17.2	17.5	70.2	89.3	81.3	**89.4**
	spout	11.3	10.3	78.7	**82.6**	73.1	79.2
Keyboard	cord	6.3	9.0	89.1	**99.0**	87.0	91.0
	key	39.4	42.4	55.9	50.6	72.2	**73.8**
Kitchen Pot	lid	50.1	54.7	76.7	**87.0**	74.4	72.5
	handle	13.6	14.5	56.9	70.1	64.7	**74.5**
Knife	blade	36.3	39.9	62.6	64.0	**64.4**	63.8
Lamp	base	64.9	72.9	88.1	89.3	82.8	**89.3**
	body	41.5	45.1	77.7	79.6	75.2	**82.1**
	bulb	0.0	0.0	13.1	**13.3**	13.0	12.7
	shade	51.9	55.2	86.6	89.4	83.8	**89.4**
Laptop	keyboard	19.3	24.2	68.3	**68.9**	68.3	67.9
	screen	18.2	20.7	55.6	58.3	68.5	**74.4**
	shaft	0.2	0.1	2.7	3.1	**3.9**	3.4
	touchpad	**17.0**	14.1	16.2	15.3	14.9	15.5
	camera	1.3	3.1	**13.1**	11.6	12.2	11.6
Lighter	lid	32.4	32.5	69.8	**72.3**	67.8	70.9
	wheel	2.0	3.8	51.5	63.0	53.0	**64.1**
	button	1.6	2.2	58.3	**64.3**	63.8	62.6
Microwave	display	1.5	0.0	28.4	28.3	27.9	**51.0**
	door	17.7	16.6	41.8	52.2	**62.2**	58.8
	handle	4.5	25.4	77.2	86.9	**90.0**	84.9
	button	0.7	0.5	23.6	25.7	25.0	**41.5**
Mouse	button	7.7	6.7	16.7	**21.0**	8.5	8.8
	cord	29.1	31.0	63.5	65.9	**66.2**	65.9
	wheel	29.4	64.1	50.9	52.4	45.8	**71.9**
Oven	door	22.7	22.0	73.6	72.2	**78.9**	73.4
	knob	2.6	6.1	67.2	67.5	69.3	**70.3**
Pen	cap	61.4	60.9	**68.6**	59.3	60.4	50.7
	button	40.8	53.7	69.1	69.2	69.9	**70.5**
Phone	lid	60.5	63.4	80.0	86.8	80.7	**89.8**
	button	15.8	20.4	29.6	28.6	34.9	**36.5**
Pliers	leg	70.2	75.1	46.7	35.5	**99.3**	99.3
Printer	button	1.2	4.0	5.9	6.3	6.8	**7.9**
Refrige-rator	door	23.2	23.1	**57.0**	56.4	52.2	52.6
	handle	8.6	24.4	49.7	53.4	40.8	**54.5**
Remote	button	17.2	28.0	36.5	36.6	46.8	**53.4**
Safe	door	21.2	24.5	68.0	**74.0**	62.9	66.5
	switch	3.4	17.0	33.8	35.7	33.4	**39.2**
	button	6.0	**8.2**	4.9	5.5	1.2	4.7
Scissors	blade	39.7	47.6	73.3	72.7	75.5	**76.1**
	handle	59.5	66.5	86.7	87.1	87.8	**88.4**
	screw	11.2	15.1	23.7	22.7	39.2	**40.9**
Stapler	body	74.1	**87.3**	85.7	86.6	84.6	86.0
	lid	65.2	78.5	68.8	76.7	75.2	**85.6**
Storage Furniture	door	31.6	36.6	50.4	55.1	61.0	**63.9**
	drawer	25.6	35.5	35.7	**37.0**	30.5	36.7
	handle	4.7	14.5	71.0	77.7	70.5	**77.9**
Suitcase	handle	30.3	43.3	77.0	**87.4**	81.5	82.3
	wheel	20.6	24.2	51.1	52.7	52.3	**54.2**
Switch	switch	21.4	37.0	52.1	51.9	53.0	**57.9**
Table	door	**0.0**	0.0	0.0	0.0	0.0	0.0
	drawer	20.0	25.3	36.7	37.2	38.8	**41.8**
	leg	42.9	48.6	70.9	72.2	70.3	**72.6**
	tabletop	46.8	54.4	71.8	79.3	78.6	**82.0**
	wheel	26.5	30.9	**68.0**	61.3	63.9	63.7
	handle	3.8	8.7	19.9	20.1	22.6	**26.4**
Toaster	button	6.4	21.3	51.3	52.0	52.0	**52.4**
	slider	7.3	17.8	48.6	50.8	**66.2**	64.8
Toilet	lid	36.6	40.9	69.6	75.9	69.5	**77.9**
	seat	8.8	11.1	22.6	**31.6**	16.4	28.6
	button	7.5	14.5	59.1	62.3	62.3	**65.9**
TrashCan	footpedal	0.0	0.0	0.0	0.0	0.1	**0.3**
	lid	34.1	47.7	62.0	**66.2**	54.6	54.6
	door	0.8	0.1	8.0	6.8	0.4	**8.7**
USB	cap	26.1	27.8	55.1	**55.6**	52.0	49.9
	rotation	34.4	39.0	49.1	58.6	55.2	**69.8**
Washing Machine	door	13.4	18.1	57.3	56.7	**63.6**	56.4
	button	1.3	2.6	32.1	32.7	39.0	**39.9**
Window	window	67.7	69.2	78.3	72.6	**78.6**	76.1
Mean		29.3	34.8	58.0	61.9	62.1	**65.0**

Table S9: Full table of part-aware instance segmentation results on the PartNet-Mobility [9] dataset.

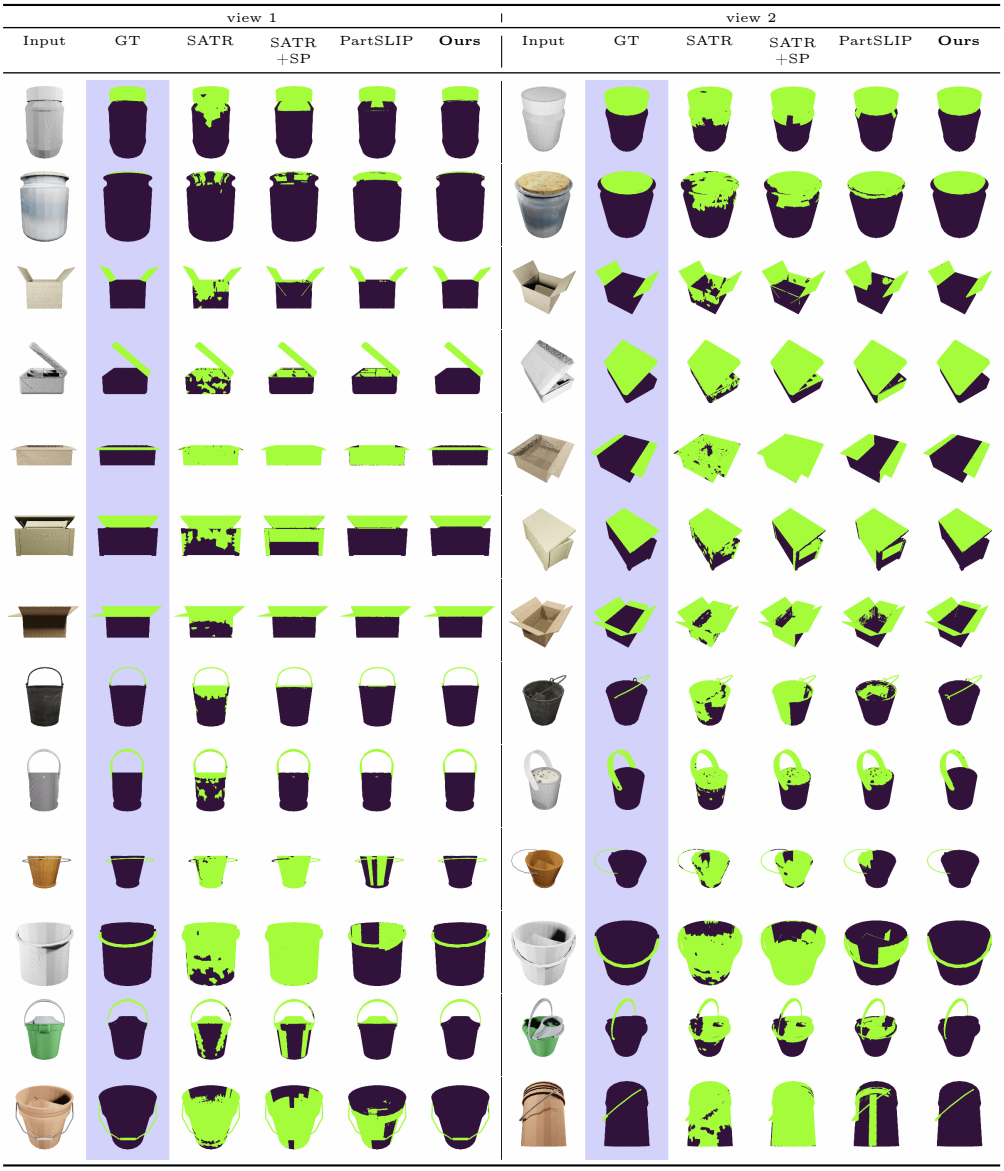
		Baselines	Ablations		
Category	Part	PartSLIP	*w/o Weight Pred.*	*w/o SAM Integ.*	**Ours**
Bottle	lid	73.4	74.9	**79.5**	77.4
Box	lid	56.3	**72.2**	60.7	62.9
Bucket	handle	10.2	**78.2**	39.3	78.2
Camera	button	37.6	36.7	**39.7**	38.7
	lens	**35.8**	32.7	29.6	29.8
Cart	wheel	69.5	**74.0**	68.3	71.6
Chair	arm	**59.1**	50.2	58.9	50.2
	back	94.2	92.3	**94.8**	93.2
	leg	72.8	79.8	72.0	**81.8**
	seat	90.1	**95.0**	89.1	91.1
	wheel	94.8	95.3	**96.1**	96.1
Clock	hand	18.7	**32.3**	18.6	18.5
Coffee Machine	button	1.3	1.3	1.3	**1.4**
	container	23.6	23.2	**24.2**	20.6
	knob	**13.4**	12.5	12.5	11.0
	lid	24.8	**24.9**	24.0	22.3
Dishwasher	door	49.0	49.0	48.8	**53.5**
	handle	31.7	34.4	40.6	**47.1**
Dispenser	head	36.1	39.6	**49.8**	44.3
	lid	76.6	81.8	85.0	**86.6**
Display	base	96.1	94.0	**98.6**	94.1
	screen	49.9	59.0	68.0	**70.7**
	support	**60.9**	52.0	45.6	51.9
Door	frame	5.1	5.5	8.5	**11.2**
	door	13.9	15.1	16.5	**20.2**
	handle	23.9	23.8	27.7	**28.1**
Eyeglasses	body	57.2	51.8	55.7	**60.6**
	leg	82.5	**85.2**	84.9	83.5
Faucet	spout	**54.4**	50.8	46.4	43.6
	switch	**31.5**	28.9	30.5	27.0
Folding Chair	seat	86.4	89.6	**100.0**	91.3
Globe	sphere	92.1	**100.0**	98.8	84.3
Kettle	lid	67.7	**88.9**	88.9	82.0
	handle	66.0	**74.9**	74.9	74.9
	spout	**68.6**	68.6	57.3	66.3
Keyboard	cord	78.6	80.2	**95.2**	91.6
	key	34.4	31.2	47.4	**49.0**
KitchenPot	lid	**95.1**	95.1	69.8	76.4
	handle	39.9	53.8	49.6	**57.4**
Knife	blade	**44.5**	41.3	39.2	32.3
Lamp	base	**84.4**	81.5	71.6	76.5
	body	85.2	85.2	70.4	**85.5**
	bulb	**15.8**	15.8	7.6	7.6
	shade	89.7	**90.2**	86.6	90.2
Laptop	keyboard	54.5	67.6	62.4	**70.1**
	screen	24.0	42.2	42.2	**60.0**
	shaft	2.0	2.0	3.2	**3.5**
	touchpad	7.4	9.0	9.4	**10.0**
	camera	**1.0**	1.0	1.0	1.0
Lighter	lid	**38.9**	38.9	17.2	29.8
	wheel	34.9	**70.1**	33.3	56.1
	button	28.6	**35.8**	31.3	30.1
Microwave	display	33.7	33.7	33.7	**38.1**
	door	39.0	23.2	**42.7**	27.2
	handle	50.5	50.5	**60.4**	50.5
	button	12.0	12.0	12.9	**13.6**
Mouse	button	**5.0**	3.0	5.0	5.0
	cord	**66.3**	66.3	66.3	66.3
	wheel	**50.5**	50.5	50.5	50.5
Oven	door	27.9	38.1	32.0	**42.5**
	knob	66.0	69.4	**72.2**	71.7
Pen	cap	51.0	30.3	**54.6**	25.9
	button	48.0	48.5	**52.4**	52.1
Phone	lid	**28.8**	28.1	16.5	23.0
	button	34.2	34.8	35.0	**35.7**
Pliers	leg	3.2	4.1	**31.1**	31.1
Printer	button	1.1	1.1	**1.7**	1.6
Refrige-rator	door	**27.2**	25.6	20.4	19.6
	handle	36.5	**39.3**	21.3	30.0
Remote	button	19.9	20.3	29.6	**33.7**
Safe	door	70.7	**76.2**	65.4	71.1
	switch	19.3	19.3	20.2	**21.8**
	button	**1.0**	1.0	0.0	1.0
Scissors	blade	13.2	14.8	13.0	**15.5**
	handle	52.9	**55.7**	47.1	54.0
	screen	4.8	6.0	9.2	**9.9**
Stapler	body	86.1	86.6	**87.0**	87.0
	lid	73.4	83.2	80.0	**94.1**
Storage Furniture	door	16.2	20.6	21.8	**25.6**
	drawer	8.9	9.2	7.7	**9.9**
	handle	62.1	**71.5**	58.7	71.1
Suitcase	handle	74.3	72.0	**74.9**	73.1
	wheel	39.4	33.7	30.1	**43.0**
Switch	switch	21.2	**22.2**	19.7	22.1
Table	door	**0.0**	0.0	0.0	0.0
	drawer	8.9	10.9	**11.6**	10.1
	leg	40.2	**42.4**	40.4	40.4
	tabletop	61.4	63.1	67.7	**68.3**
	wheel	**73.0**	54.3	61.9	66.6
	handle	12.1	**14.3**	12.5	13.9
Toaster	button	36.8	30.1	**37.1**	31.6
	slider	29.6	32.1	**49.0**	45.6
Toilet	lid	49.7	54.9	47.5	**57.2**
	seat	2.1	**6.1**	1.4	5.0
	button	56.9	**61.3**	59.2	59.5
TrashCan	footpedal	**0.0**	0.0	0.0	0.0
	lid	32.8	34.6	**37.0**	32.0
	door	2.0	2.0	1.0	**2.9**
USB	cap	17.1	15.3	**25.3**	19.6
	rotation	24.7	19.5	24.2	**33.4**
Washing Machine	door	35.3	27.4	**35.6**	22.9
	button	12.4	12.4	**17.3**	17.3
Window	window	23.6	19.3	**26.1**	20.0
Mean		41.6	44.7	44.2	**45.6**

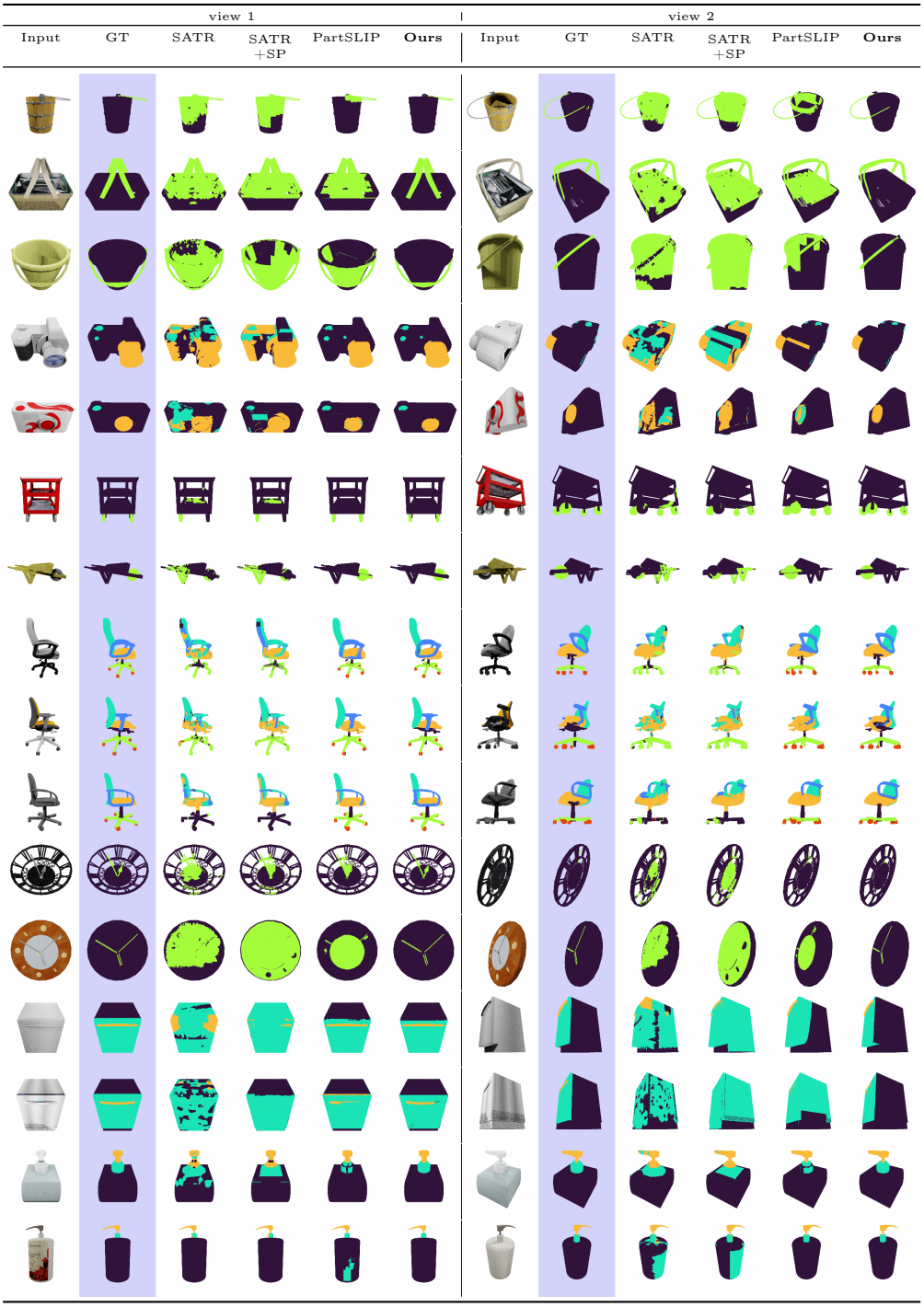
Table S10: Full table of part-agnostic instance segmentation results on the PartNet-Mobility [9] dataset.

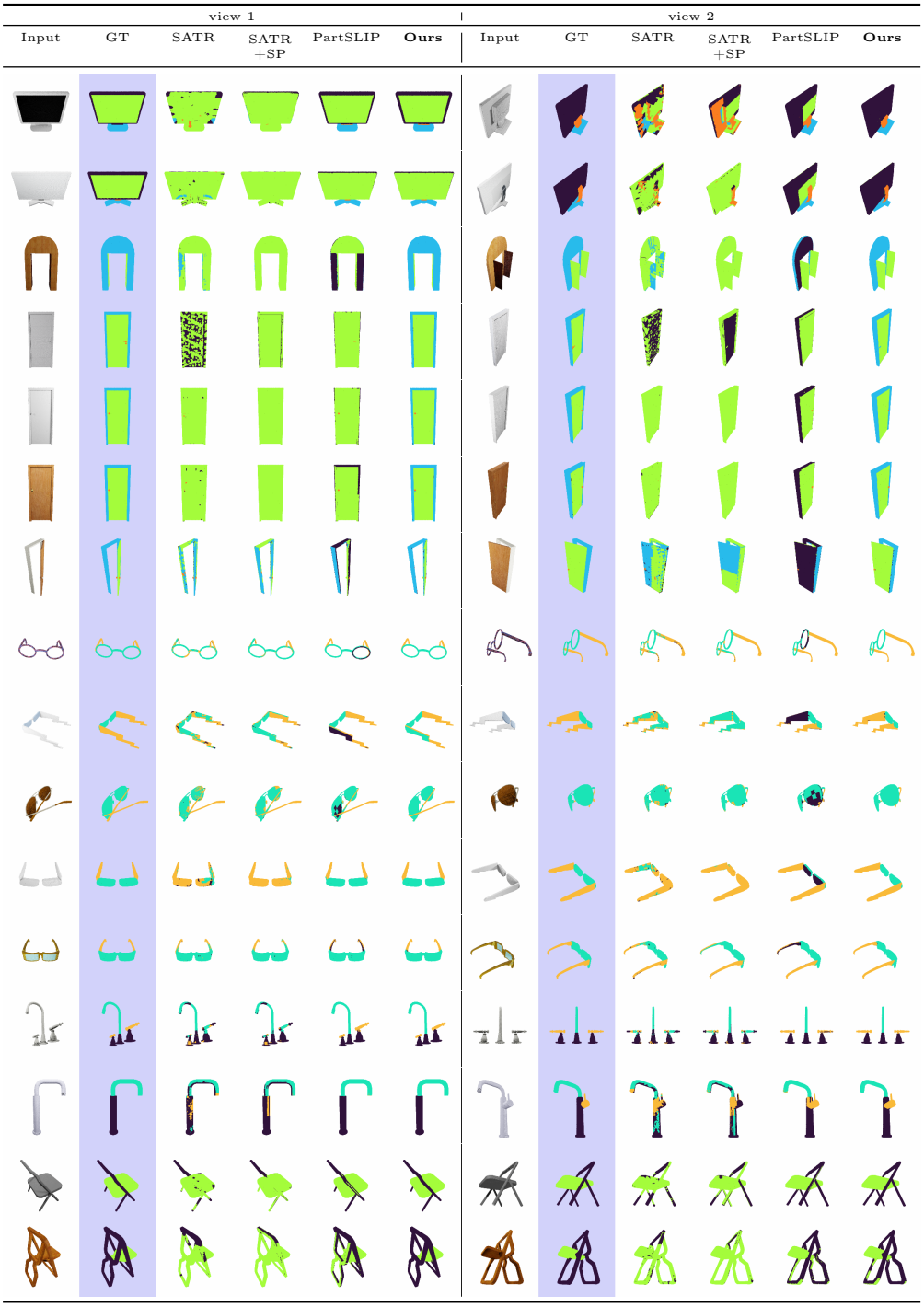
	Baselines		Ablations		
Category	SAM3D	PartSLIP	*w/o Weight Pred.*	*w/o SAM Integ.*	**Ours**
Bottle	15.3	73.4	74.9	**79.5**	76.2
Box	18.2	56.3	**72.2**	60.7	69.6
Bucket	12.0	10.2	**78.2**	39.3	78.2
Camera	2.2	37.1	35.5	**37.8**	37.0
Cart	8.5	69.5	**74.0**	68.3	71.2
Chair	5.7	80.7	81.1	81.2	**83.3**
Clock	1.0	18.7	**32.3**	18.6	18.5
Coffee Machine	3.6	**9.8**	9.5	9.4	9.4
Dishwasher	6.7	39.0	41.9	47.1	**49.7**
Dispenser	20.9	59.0	59.9	**69.0**	65.5
Display	34.1	61.5	66.4	66.0	**68.4**
Door	**17.2**	11.5	11.6	13.9	16.0
Eyeglasses	27.6	71.8	72.1	72.6	**73.9**
Faucet	23.7	**42.8**	37.1	37.9	35.6
Folding Chair	27.2	86.4	89.6	**100.0**	91.3
Globe	4.2	92.1	**100.0**	98.8	84.3
Kettle	41.8	69.5	75.3	**76.3**	75.9
Keyboard	1.0	34.1	31.0	47.3	**48.9**
KitchenPot	31.9	55.4	**66.2**	52.3	65.4
Knife	2.2	**44.5**	41.3	39.2	32.1
Lamp	48.4	77.0	**77.7**	66.5	75.9
Laptop	1.9	19.5	26.5	27.4	**34.1**
Lighter	8.9	32.7	**49.9**	26.6	39.8
Microwave	1.4	15.0	12.9	**18.7**	16.2
Mouse	2.5	12.9	11.2	20.5	**21.0**
Oven	1.0	46.8	55.6	56.4	**62.0**
Pen	8.2	40.8	35.8	**47.2**	34.5
Phone	1.1	31.0	33.3	31.6	**34.7**
Pliers	22.2	3.2	4.1	**31.1**	31.1
Printer	1.0	1.1	1.1	**1.7**	1.5
Refrigerator	1.8	27.7	**30.5**	19.4	22.0
Remote	1.0	19.9	20.3	29.6	**33.7**
Safe	2.0	15.5	**15.6**	14.1	15.2
Scissors	7.4	27.0	**29.4**	24.9	26.2
Stapler	45.1	76.0	80.4	**85.0**	82.7
Storage Furniture	1.2	29.7	35.9	34.8	**41.7**
Suitcase	1.8	**47.3**	46.2	47.1	43.8
Switch	5.0	21.2	22.2	19.7	**22.4**
Table	10.8	28.7	**29.2**	27.6	28.2
Toaster	1.8	34.8	35.9	38.6	**39.2**
Toilet	3.6	35.4	40.0	36.6	**41.0**
TrashCan	7.6	22.6	**24.5**	24.3	22.8
USB	23.4	20.5	15.2	**27.3**	23.5
Washing Machine	2.8	17.5	14.6	**21.4**	19.6
Window	18.0	23.6	19.3	**26.1**	19.5
Mean	12.1	38.9	42.6	42.6	**44.1**

S.12 Additional Qualitative Results for Semantic Segmentation

We present additional part segmentation results below, encompassing semantic segmentation. Each row shows the same object with different views.

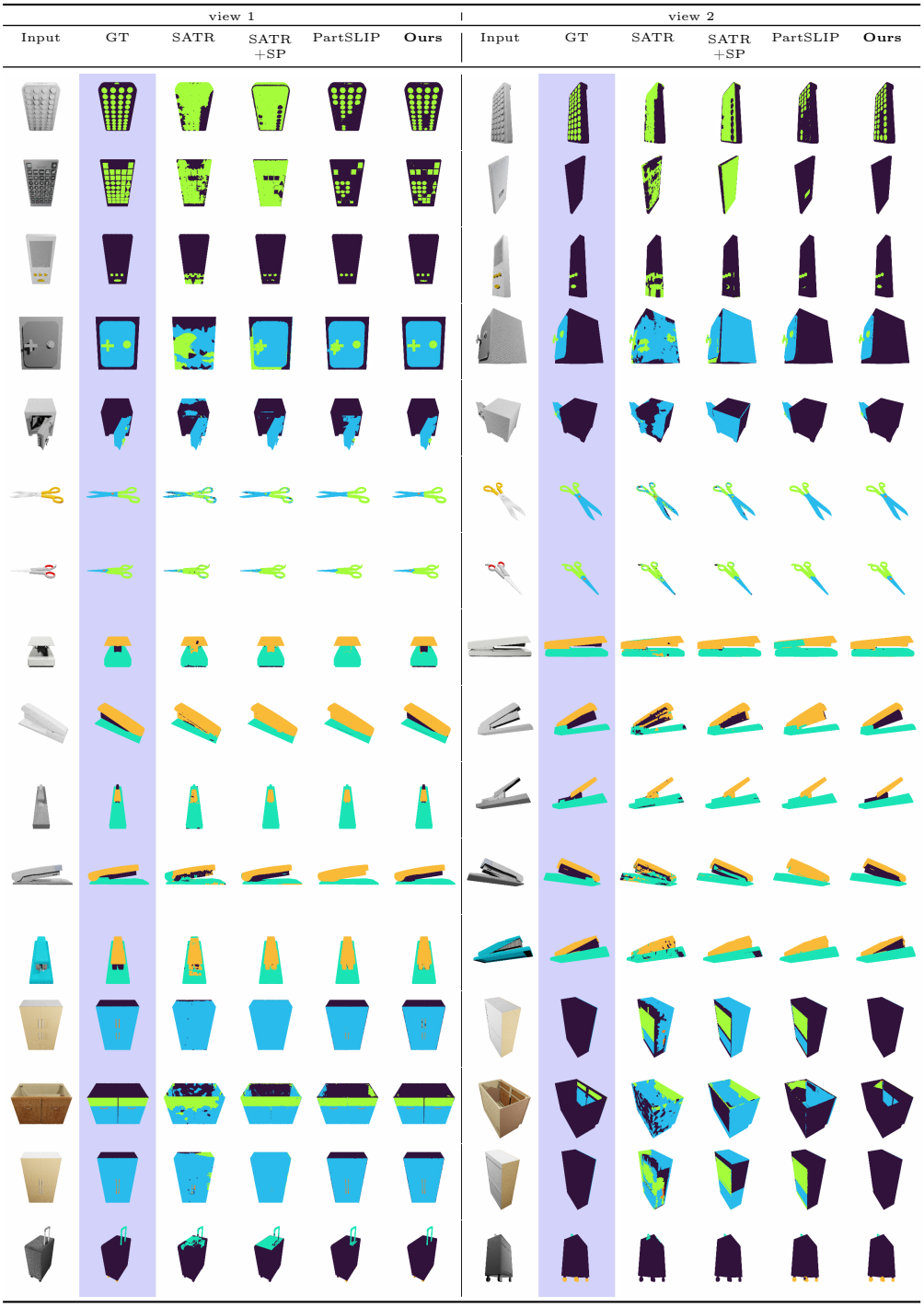


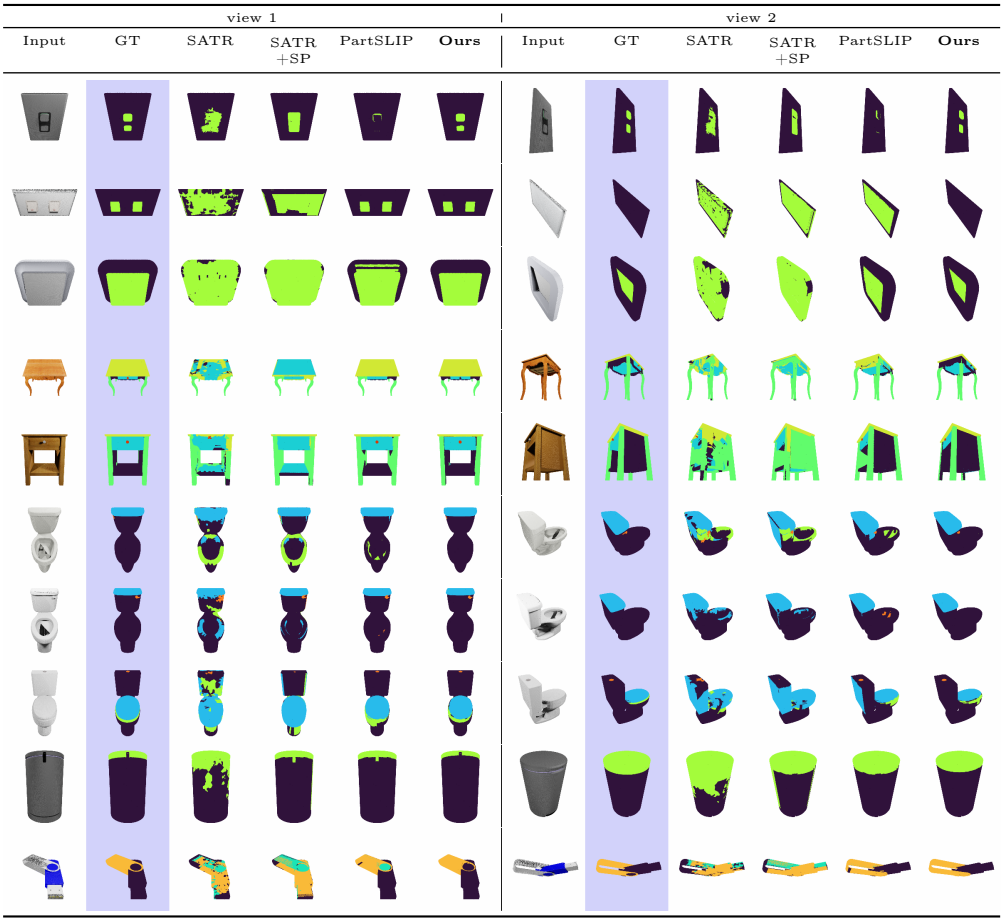

view 1
Input
GT
SATR
SATR +SP
PartSLIP
Ours
view 2
Input
GT
SATR
SATR +SP
PartSLIP
Ours


view 1
view 2
Input
GT
SATR
SATR +SP
PartSLIP
Ours
Input
GT
SATR
SATR +SP
PartSLIP
Ours


view 1
view 2
Input
GT
SATR
SATR +SP
PartSLIP
Ours
Input
GT
SATR
SATR +SP
PartSLIP
Ours

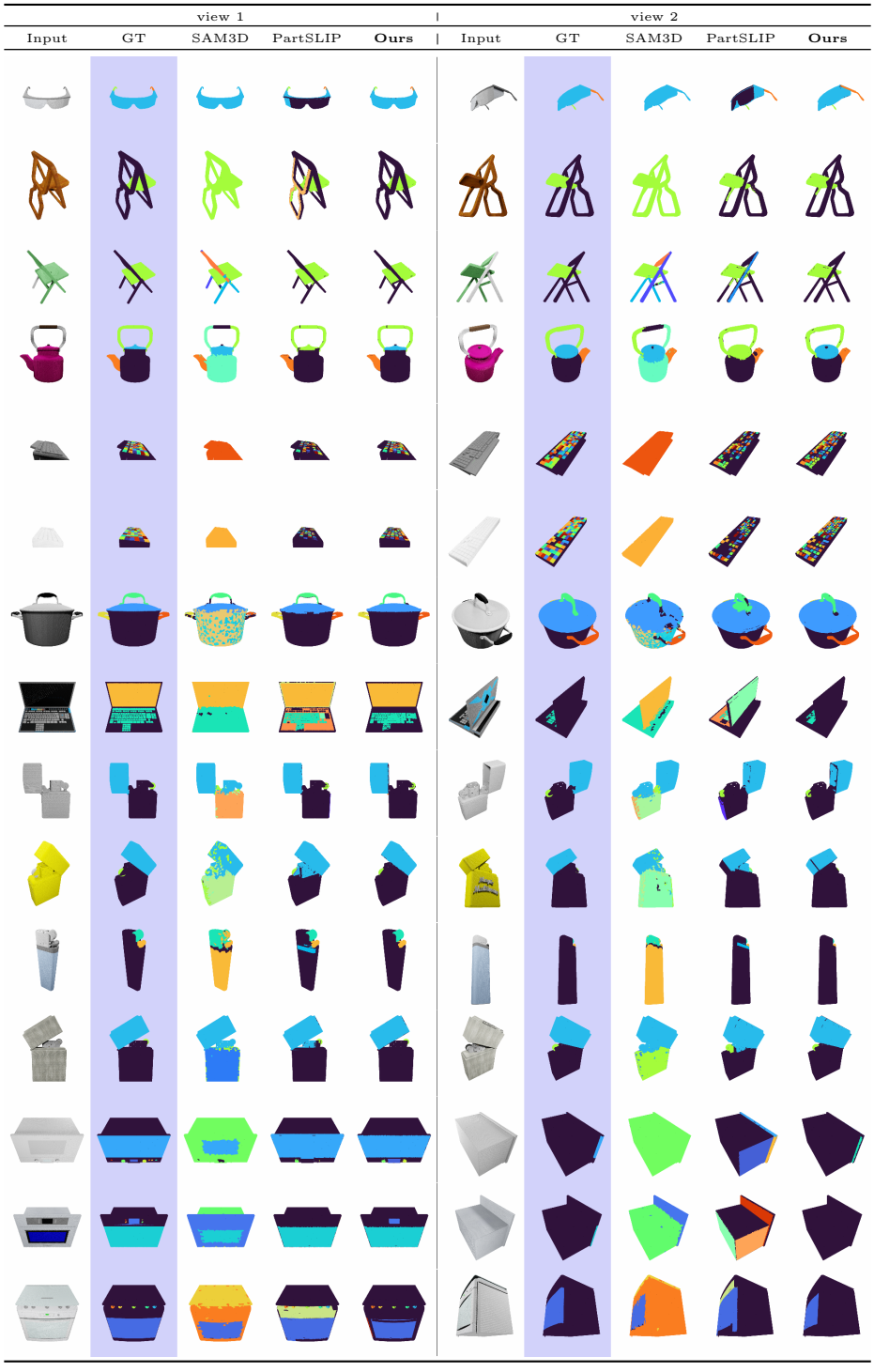

view 1
view 2
Input
GT
SATR
SATR +SP
PartSLIP
Ours
Input
GT
SATR
SATR +SP
PartSLIP
Ours

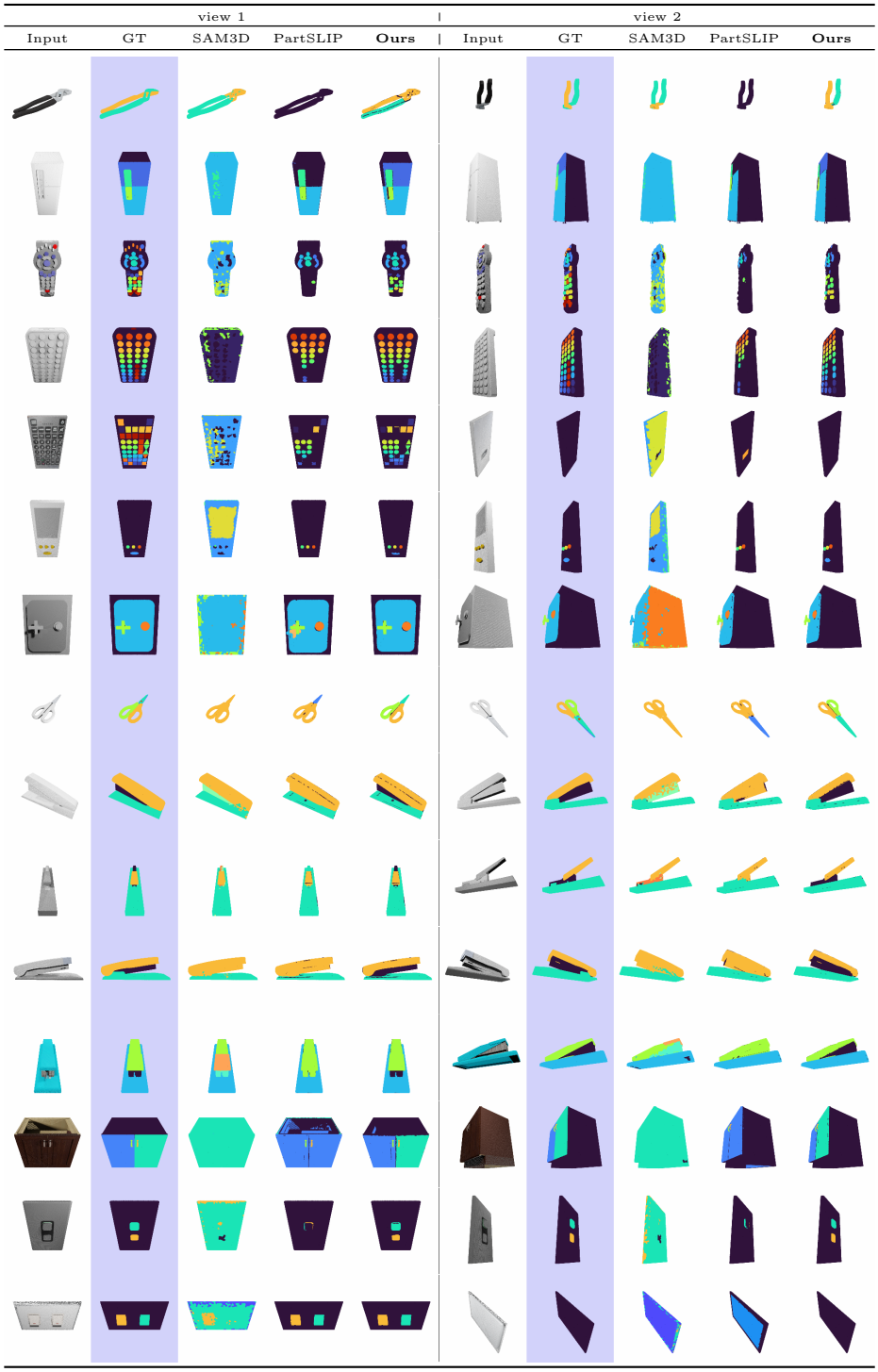

view 1
view 2
Input
GT
SATR
SATR +SP
PartSLIP
Ours
Input
GT
SATR
SATR +SP
PartSLIP
Ours

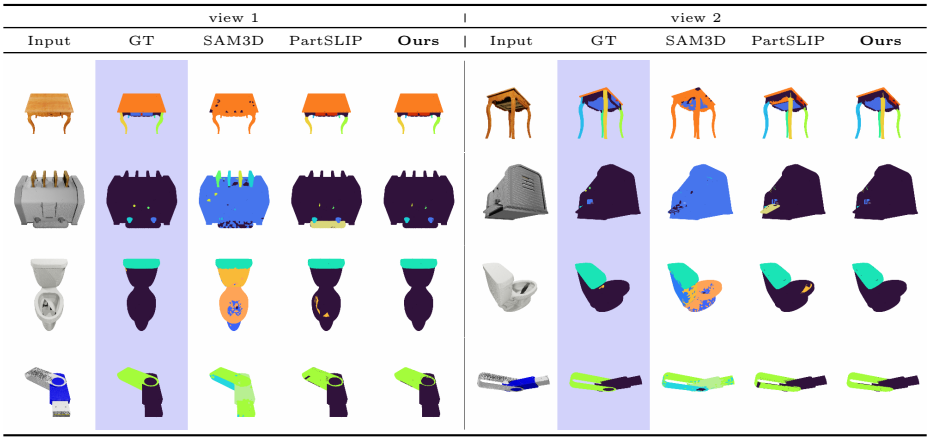

view 1
Input
GT
SATR
SATR +SP
PartSLIP
Ours
view 2
Input
GT
SATR
SATR +SP
PartSLIP
Ours

S.13 Additional Qualitative Results for Instance Segmentation

We present additional part segmentation results below, encompassing instance segmentation. Each row shows the same object with different views.


view 1
view 2
Input
GT
SAM3D
PartSLIP
Ours
Input
GT
SAM3D
PartSLIP
Ours


view 1
view 2
Input
GT
SAM3D
PartSLIP
Ours
Input
GT
SAM3D
PartSLIP
Ours


view 1
view 2
Input
GT
SAM3D
PartSLIP
Ours
Input
GT
SAM3D
PartSLIP
Ours

References

1. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: LoRA: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685 (2021)
2. Ke, L., Ye, M., Danelljan, M., Liu, Y., Tai, Y.W., Tang, C.K., Yu, F.: Segment anything in high quality. In: NeurIPS (2023)
3. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. arXiv preprint arXiv:2304.02643 (2023)
4. Li, L.H., Zhang, P., Zhang, H., Yang, J., Li, C., Zhong, Y., Wang, L., Yuan, L., Zhang, L., Hwang, J.N., et al.: Grounded language-image pre-training. In: CVPR (2022)
5. Liu, M., Zhu, Y., Cai, H., Han, S., Ling, Z., Porikli, F., Su, H.: PartSLIP: Low-shot part segmentation for 3d point clouds via pretrained image-language models. In: CVPR (2023)
6. Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Li, C., Yang, J., Su, H., Zhu, J., et al.: Grounding DINO: Marrying dino with grounded pre-training for open-set object detection. arXiv preprint arXiv:2303.05499 (2023)
7. Ren, T., Liu, S., Zeng, A., Lin, J., Li, K., Cao, H., Chen, J., Huang, X., Chen, Y., Yan, F., Zeng, Z., Zhang, H., Li, F., Yang, J., Li, H., Jiang, Q., Zhang, L.: Grounded SAM: Assembling open-world models for diverse visual tasks. arXiv preprint arXiv:2401.14159 (2024)
8. Wu, T., Zhang, J., Fu, X., Wang, Y., Ren, J., Pan, L., Wu, W., Yang, L., Wang, J., Qian, C., Lin, D., Liu, Z.: OmniObject3D: Large-vocabulary 3D object dataset for realistic perception, reconstruction and generation. In: CVPR (2023)
9. Xiang, F., Qin, Y., Mo, K., Xia, Y., Zhu, H., Liu, F., Liu, M., Jiang, H., Yuan, Y., Wang, H., Yi, L., Chang, A.X., Guibas, L.J., Su, H.: SAPIEN: A simulated part-based interactive environment. In: CVPR (2020)
10. Yi, K.M., Trulls, E., Ono, Y., Lepetit, V., Salzmann, M., Fua, P.: Learning to find good correspondences. In: CVPR (2018)
11. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: ICCV (2023)