UNICL-SAM: Uncertainty-Driven In-Context Segmentation with Part Prototype Discovery

Supplementary Material

UGGN Hyperparameters	
Hidden size	256
Hidden dropout prob	0.5
Variance gamma	1
Number of samples	1
Graph sparse ratio r	0.5
Uncertainty gate	1e-4

(a) Hyperparameters of UGGN.

QPG Hyperparameters	
Number of query tokens	50
Hidden size	256
Encoder hidden size	256
Attention heads	16
Hidden dropout prob	0.1
Attention dropout prob	0.1
Layer norm epsilon	1e-12
Hidden activation	gelu

(b) Hyperparameters of QPG.

Loss Hyperparameters	
λ_{bce}	2
λ_{dice}	1
$\lambda_{KL}^{\mathcal{G}}$	1
λ_{KL}^{reg}	5e-4
λ_{2}^{reg}	5e-4
λ_{P}	0.5

(c) Hyperparameters of loss.

Table A1. Hyperparameters for our UNICL-SAM.

A. Additional Implementation Details

In this section, we provide additional details of some proposed modules/operations and training settings.

A.1. Multi-scale feature adapter.

With a VitDet [1] style simple feature pyramid (SFP) architecture, we generate feature maps at scales $S = \{\frac{1}{2}, 1, 2, 4\}$ by applying convolutions with strides $\{\frac{1}{2}, 1, 2, 4\}$:

$$\{\mathbf{F}^{\frac{1}{2}}, \mathbf{F}^{1}, \mathbf{F}^{2}, \mathbf{F}^{4}\} = SFP(\mathbf{F}),\, \mathbf{F} \in \{\mathbf{F}_r, \mathbf{F}_t\} \tag{1}$$

Then with the multi-scale features, we apply upsampling and adaptive max pooling transformations to map them to the vanilla feature map size H, W. Subsequently, we aggregate the features in the spatial domain through summation followed by average pooling. A non-local mapping ($Nonlocal$) is then employed to produce the final feature map $\mathbf{F}$:

$$\begin{aligned} \mathbf{F}^{'} &= \frac{1}{S}\sum_{s=1}^{S} \mathbf{F}^{s},\, \mathbf{F}^{s} \in \{\mathbf{F}^{\frac{1}{2}}, \mathbf{F}^{1}, \mathbf{F}^{2}, \mathbf{F}^{4}\} \\ \mathbf{F} &= Nonlocal(\mathbf{F}^{'}) \end{aligned} \tag{2}$$

A.2. Correspondence extraction.

With the masked reference feature $\mathbf{F}_{rm}$ and target image feature $\mathbf{F}_t$, the patch-level similarity $\mathbf{Corr} \in \mathbb{R}^{HW \times HW}$ is calculated as follows:

$$\mathbf{Corr} = \frac{\mathbf{F}_{rm} \cdot \mathbf{F}_t}{\|\mathbf{F}_{rm}\| \cdot \|\mathbf{F}_t\|} \tag{3}$$

Then, we calculate the pseudo mask $\mathbf{y}_t^{pseudo}$ by retaining the maximum similarity at each patch and normalizing it to the $[0, 1]$ range using min-max normalization. Benefiting from the great generalization capabilities of DINOv2 and our multi-scale feature adapter module, as illustrated in Figure 1, the correspondence map shows a high correlation with the ideal segmentation label.

A.3. Query-based prompt generator.

The query-based prompt generator (QPG) is a simple decoder-style transformer architecture. For each layer, we first apply self-attention and then follow with cross-attention to interact with in-context instructions.

A.4. Training pipeline.

We simply employ uniform dataset-level sampling without adjusting the dataset ratios. Data augmentations are applied to reference and target images, including random resizing and cropping. More details on the model hyperparameters are shown in Table A1.

B. Robustness Simulation Tests Details

In this section, we provide a detailed description of the robustness simulation tests we constructed. To comprehensively evaluate the model on the degradation and domain shifts that may be encountered in various real-world scenarios, we designed 6 categories comprising a total of 18 image transformations, accompanied by 3 label transformations and varying degradation levels for each transformation. The specific settings are outlined in Table B2. For transformations with multiple parameters, each combination will be systematically explored. To mitigate evaluation costs, we apply various transformations while fixing the seed on 1,500 images randomly sampled from the COCO test split for all transformations.

We specifically designate the “Generation” type because this process does not involve transforming the original reference examples. Instead, we utilize the category names from COCO to construct input prompts such as ”A good photo of [c].”, where [c] denotes a specific category. Utilizing the SDv2.0 model [6], we generate 64 images for each category based on the constructed prompts. Subsequently, we

Type	Transformation	Argument	Value
Support Image			
Color	Color jitter	Brightness	0.5, 1.5
		Contrast	0.5, 1.5
		Saturation	0.5, 1.5
		Hue	0.1, 0.3, 0.5
	Gray	-	-
	Light	Lightness	0.3, 0.7, 1.3, 1.7
Blurness	Gaussian blur	Kernel size	7, 11, 15
	Motion blur	Blur limit	5, 9, 15
	Mean shift blur	Color radius	1, 10, 50
	Sharpness	Factor	5, 10, 15
Compression	Jpeg compression	Quality	5, 10, 20
	Posterize	Bit	3, 2, 1
	Solarize	Threshold	0, 64, 128, 192, 256
	Gaussian noise	Var limit	5e3, 1e4, 5e4
Space	Horizontal flip	-	-
	Vertical flip	-	-
	Rotate	Degree	45, 90, 180
Domain shift	Cartoon	-	-
	Sobel	Kernel size	3
	Generation*	-	-
Support Label			
Deformation	Bbox	-	-
	Erode	Kernel size	5
	Dilate	Kernel size	5

Table B2. A detailed list of the transformation arguments and parameters for the robustness simulation tests is presented. The values further to the right indicate a greater impact.

apply the label generation method of X-paste [11] to create binary segmentation labels corresponding to each generated image. We then sample examples from the sets as usual.

C. Additional Ablations

In this section, we conduct more thorough ablation experiments to illustrate the effectiveness of our method.

C.1. Ablations on training data.

We further investigate the effectiveness of combining diverse datasets under our joint in-context training framework. As illustrated in line 1 of Table C3, the model demonstrates strong performance even when trained exclusively on a subset of COCO comprising approximately 35k images. We attribute this to effective architectural design. Further training on the full COCO can provide further improvement. We validate that incorporating more diverse semantic segmentation data, such as ADE20K, helps improve the model on out-of-domain FSS1000 and LVIS-92^i tests.

Training Data	COCO-20^i	FSS-1000	LVIS-92^i
COCO-s	74.6	81.7	29.9
COCO	77.3	82.4	32.0
+ ADE20k	77.8	84.0	34.1

Table C3. Ablation of training data.

C.2. Ablations on model trainable parameters.

We construct ablation experiments to analyze the impact of the proposed modules on trainable parameters and performance, with results presented in Table C4. In this ablation experiment, we utilized the entire COCO dataset to train the model and observe the resulting changes in performance. While multi-scale features enhance model performance, they also introduce significant computational overhead (approximately 45M trainable parameters when comparing lines 2 and 4). It is noteworthy that the difference in parameters between lines 1 and 3 is slightly greater than that between lines 2 and 4. This discrepancy can be attributed to changes in the mapping layer parameters resulting from the introduction of local prototypes. The model continues to achieve commendable results even without the multi-scale structure. We attribute this sustained performance to the refined visual features generated by the optimization strategies afforded by uncertainty modeling and effective part prototype guidance. It is worth mentioning that UGGN only requires about **2M** parameters, demonstrating its lightweight and efficient design. Therefore, we recommend utilizing the full version for scenarios that demand superior performance; however, the lightweight UGGN is sufficient for less demanding applications.

MFA	UGGN	Trainable Params	COCO-20^i	FSS-1000	LVIS-92^i
✓	✓	55.4M	77.3	82.4	32.0
✓	✗	52.5M	77.4	81.2	29.8
✗	✓	9.8M	77.0	81.3	30.2
✗	✗	7.8M	75.4	80.7	28.9

Table C4. Ablations of model trainable parameters.

D. Additional Quantitative Analysis

D.1. Comparisons on robustness simulation testing.

Here we provide a detailed comparison of robustness testing. As illustrated in Table D5, our UNICL-SAM significantly outperforms previous approaches across all transformations. This demonstrates the effectiveness and robustness of our proposed framework, which is well-equipped to handle the diverse examples that may arise in practical applications.

In addition, we made several intriguing observations. All approaches exhibited insensitivity to color variations; however, they demonstrated significant performance degradation in response to common noise types, such as Gaussian

Type	Transformation	Matcher	SegGPT	SINE	UNICL-SAM
Clean		41.6	62.1	70.0	**79.8**
		Support Image			
Color	Color jitter	40.3$_{(-1.3)}$	54.2$_{(-7.9)}$	69.2$_{\underline{(-0.8)}}$	**78.7**$_{(-1.1)}$
	Gray	42.1$_{\underline{(+0.5)}}$	59.5$_{(-2.6)}$	69.1$_{(-0.9)}$	**78.8**$_{(-1.0)}$
	Light	40.7$_{(-0.9)}$	64.6$_{\underline{(+2.5)}}$	69.7$_{(-0.3)}$	**78.9**$_{(-0.9)}$
Blurness	Gaussian blur	39.1$_{(-2.5)}$	54.5$_{(-7.6)}$	67.2$_{(-2.8)}$	**77.6**$_{\underline{(-2.2)}}$
	Motion blur	40.1$_{(-1.5)}$	61.8$_{\underline{(-0.3)}}$	69.2$_{(-0.8)}$	**77.5**$_{(-2.3)}$
	Mean shift blur	38.5$_{(-3.1)}$	61.7$_{\underline{(-0.4)}}$	68.3$_{(-1.7)}$	**78.4**$_{(-1.4)}$
	Sharpness	38.7$_{(-2.9)}$	58.6$_{(-3.5)}$	68.5$_{(-1.5)}$	**78.5**$_{\underline{(-1.3)}}$
Compression	Jpeg compression	38.6$_{(-3.0)}$	61.0$_{\underline{(-2.1)}}$	67.3$_{(-2.7)}$	**77.7**$_{\underline{(-2.1)}}$
	Posterize	37.4$_{(-4.2)}$	61.1$_{\underline{(-1.0)}}$	65.9$_{(-4.1)}$	**76.4**$_{(-3.4)}$
	Solarize	39.7$_{(-1.9)}$	61.7$_{\underline{(-0.4)}}$	69.1$_{(-0.9)}$	**78.7**$_{(-1.1)}$
	Gaussian noise	20.7$_{(-20.9)}$	48.5$_{(-13.6)}$	45.7$_{(-24.3)}$	**71.0**$_{\underline{(-8.8)}}$
Space	Horizontal flip	40.5$_{(-1.1)}$	64.9$_{\underline{(+2.8)}}$	70.6$_{(+0.6)}$	**79.0**$_{(-0.8)}$
	Vertical flip	35.2$_{(-6.4)}$	43.8$_{(-18.3)}$	63.8$_{(-6.2)}$	**74.0**$_{\underline{(-5.8)}}$
	Rotate	38.4$_{(-3.2)}$	60.0$_{(-2.1)}$	66.4$_{(-3.6)}$	**78.2**$_{\underline{(-1.6)}}$
Domain shift	Cartoon	31.8$_{(-9.8)}$	59.0$_{\underline{(-3.1)}}$	60.3$_{(-9.7)}$	**72.5**$_{(-7.3)}$
	Sobel	35.2$_{(-6.4)}$	46.9$_{(-15.2)}$	63.1$_{(-6.9)}$	**74.9**$_{\underline{(-4.9)}}$
	Generation*	42.1$_{\underline{(+0.5)}}$	62.0$_{(-0.1)}$	67.3$_{(-2.7)}$	**77.6**$_{(-2.2)}$
		Support Label			
Deformation	Bbox	26.4$_{(-15.2)}$	46.2$_{(-15.9)}$	45.9$_{(-24.1)}$	**69.0**$_{\underline{(-10.8)}}$
	Erode	39.4$_{\underline{(-2.2)}}$	53.5$_{(-8.6)}$	59.2$_{(-10.8)}$	**76.0**$_{(-3.8)}$
	Dilate	40.5$_{(-1.1)}$	60.2$_{(-1.9)}$	67.8$_{(-2.2)}$	**79.0**$_{\underline{(-0.8)}}$

Table D5. A comprehensive comparison of the robustness of existing advanced in-context segmentation generalists across various transformations. We highlight the best performance in **bold**, while indicating minimal performance degradation in $\underline{\text{underline}}$.

Methods	DAVIS 2017			DAVIS 2016		
	$J\&F\uparrow$	$J\uparrow$	$F\uparrow$	$J\&F\uparrow$	$J\uparrow$	$F\uparrow$
			generalist model			
Painter [9]	34.6	28.5	40.8	70.3	69.6	70.9
SegGPT [10]	75.6	72.5	78.6	$\underline{83.7}$	$\underline{83.6}$	$\underline{83.8}$
VRP-SAM [8]	64.8	62.1	67.4	-	-	-
SINE [3]	$\underline{77.0}$	$\underline{72.6}$	$\underline{81.3}$	82.3	81.4	83.2
UNICL-SAM	74.0	69.3	78.6	81.7	76.5	**86.8**

Table D6. Comparison with state-of-the-art generalist models on video object segmentation benchmarks. Previous state-of-the-art results are $\underline{\text{underlined}}$.

noise. A similar phenomenon was observed with the spatial transformation of vertical flipping, where SegGPT experienced a substantial decline in performance, potentially indicating underlying issues associated with the MAE framework. Furthermore, domain shift notably affected the performance of all approaches. However, examples generated based on SD still achieved good performance compared with real-world ones. This suggests that current state-of-the-art generative methods can provide high-quality examples that assist in context segmentation, thereby offering a user-friendly approach to example provision. Regarding the impact of example labeling, we found that a simple transformation involving the bounding rectangle of example masks significantly influenced performance. This transformation is commonly encountered in practical applications involving interactive labeling methods (*e.g.*, bbox is one of SAM's prompting types). This finding indicates that existing methods require further enhancement to effectively adapt to user interaction scenarios. We hope these analyses will aid future research in further evaluating model performance and implementing targeted improvements.

D.2. Comparisons on video object segmentation benchmarks.

We further conduct experiments on DAVIS 2016 [4] and 2017 [5] video object segmentation benchmarks. The results in Table D6 demonstrate UNICL-SAM's competitive DAVIS benchmark performance. Note that UNICL-

SAM uses fewer training data than SegGPT (trained on 12 datasets) and SINE (trained with large-scale instance segmentation dataset Object365 [7]), highlighting its versatility and generalization capability.

E. Additional Qualitative Results

This section provides more visualizations of the intermediate outputs and predictions generated by UNICL-SAM. We also introduce a comparison with existing solutions to better analyze.

E.1. Qualitative Results on COCO-20^i.

To illustrate the great in-context segmentation ability of our UNICL-SAM, we provide qualitative results on the COCO-20^i benchmark. As illustrated in Figure E1, the model exhibits a robust semantic understanding across diverse scenarios. For the examples, it is evident that uncertainty is often significantly activated at edges and small targets, while the clustered masks reveal specific local semantics.

We would like to emphasize the capabilities of UNICL-SAM with respect to target images. The pseudo masks generated based on the correspondence map often contain significant noise, particularly for small targets, as observed in lines (5) to (11). Following our uncertainty quantization stage, the estimated uncertainty maps indicate high activation in the noisy irrelevant areas. Notably, after implementing our feature optimization strategy, the pseudo masks effectively reduce significant noise, resulting in a more precise delineation of regions of interest.

This enhancement allows the model to generate precise segmentation predictions despite substantial size discrepancies between examples and targets (*e.g.*, lines (1), (2), and (8) to (11)). It effectively addresses challenges posed by occluded objects (*e.g.*, lines (1) and (4)) and varying lighting conditions, as illustrated in row (7). Furthermore, the model demonstrates robustness in handling objects exhibiting different states, as seen in rows (3) and (9). Even for targets exhibiting similar interferences, such as those in line (2), our model successfully distinguishes and segments the correct target. This analysis underscores both the superiority of the proposed UNICL-SAM framework and the effectiveness of the optimization strategies employed.

E.2. Visualization comparisons.

We also perform a visual comparison with existing advanced context segmentation models, as shown in Figure E2. Our baseline exhibits notable positioning errors and fragmented masks, as illustrated in lines (1), (3), (6), and (7). The introduction of the uncertainty modeling and correspondence feature optimization strategy and part prototypes enables UNICL-SAM to effectively manage these complex segmentation scenarios. Among all the evaluated models, SINE [3] demonstrates the highest performance, successfully segmenting nearly all the desired targets. However, it is still susceptible to issues of over-segmentation and inadequate edge refinement. Conversely, SegGPT [10] frequently fails to produce segmentation results, while Matcher [2] often misidentifies the target areas. Visual comparisons further underscore the superiority of our model, which adeptly addresses segmentation challenges that previous methods struggle to resolve.

E.3. Qualitative results on robustness tests.

We present qualitative results from the robustness tests we conducted. As illustrated in Figure E3, we randomly selected a variety of transformations that encompass six major categories we designed. The results indicate that UNICL-SAM performs segmentation effectively, even in the presence of significant size discrepancies and complex lighting conditions. This capability extends to managing extreme transformations, including solarization, vertical flipping, Sobel filtering, and bounding box labeling. Only the semantic ambiguity arising from bounding box annotations may affect the model performance, as indicated in the final line. We attribute this success to the effective architecture and feature optimization strategies proposed in this study. These findings unequivocally demonstrate the robustness of UNICL-SAM and highlight its potential for application in real-world scenarios.

F. Discussions

In this paper, we introduce UNICL-SAM, which exhibits superior performance and robustness, with its core component, UGGN, requiring only a minimal number of parameters (approximately 2M). Notably, we have fixed the parameters of the pre-trained SAM, enabling the integration of existing SAM optimization strategies to support a wider range of applications. Although the performances on video object segmentation benchmarks still show room for improvement, we anticipate the potential of our UNICL-SAM to be applied in real-world applications due to its strong robustness.

References

[1] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. In *European conference on computer vision*, pages 280–296. Springer, 2022. 1

[2] Yang Liu, Muzhi Zhu, Hengtao Li, Hao Chen, Xinlong Wang, and Chunhua Shen. Matcher: Segment anything with one shot using all-purpose feature matching. *arXiv preprint arXiv:2305.13310*, 2023. 4

[3] Yang Liu, Chenchen Jing, Hengtao Li, Muzhi Zhu, Hao Chen, Xinlong Wang, and Chunhua Shen. A simple image

segmentation framework via in-context examples. *Advances in Neural Information Processing Systems*, 2024. 3, 4

[4] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 724–732, 2016. 3

[5] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017. 3

[6] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1

[7] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8430–8439, 2019. 4

[8] Yanpeng Sun, Jiahui Chen, Shan Zhang, Xinyu Zhang, Qiang Chen, Gang Zhang, Errui Ding, Jingdong Wang, and Zechao Li. Vrp-sam: Sam with visual reference prompt. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23565–23574, 2024. 3

[9] Xinlong Wang, Wen Wang, Yue Cao, Chunhua Shen, and Tiejun Huang. Images speak in images: A generalist painter for in-context visual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6830–6839, 2023. 3

[10] Xinlong Wang, Xiaosong Zhang, Yue Cao, Wen Wang, Chunhua Shen, and Tiejun Huang. Seggpt: Segmenting everything in context. *arXiv preprint arXiv:2304.03284*, 2023. 3, 4

[11] Hanqing Zhao, Dianmo Sheng, Jianmin Bao, Dongdong Chen, Dong Chen, Fang Wen, Lu Yuan, Ce Liu, Wenbo Zhou, Qi Chu, et al. X-paste: Revisiting scalable copy-paste for instance segmentation using clip and stablediffusion. In *International Conference on Machine Learning*, pages 42098–42109. PMLR, 2023. 2

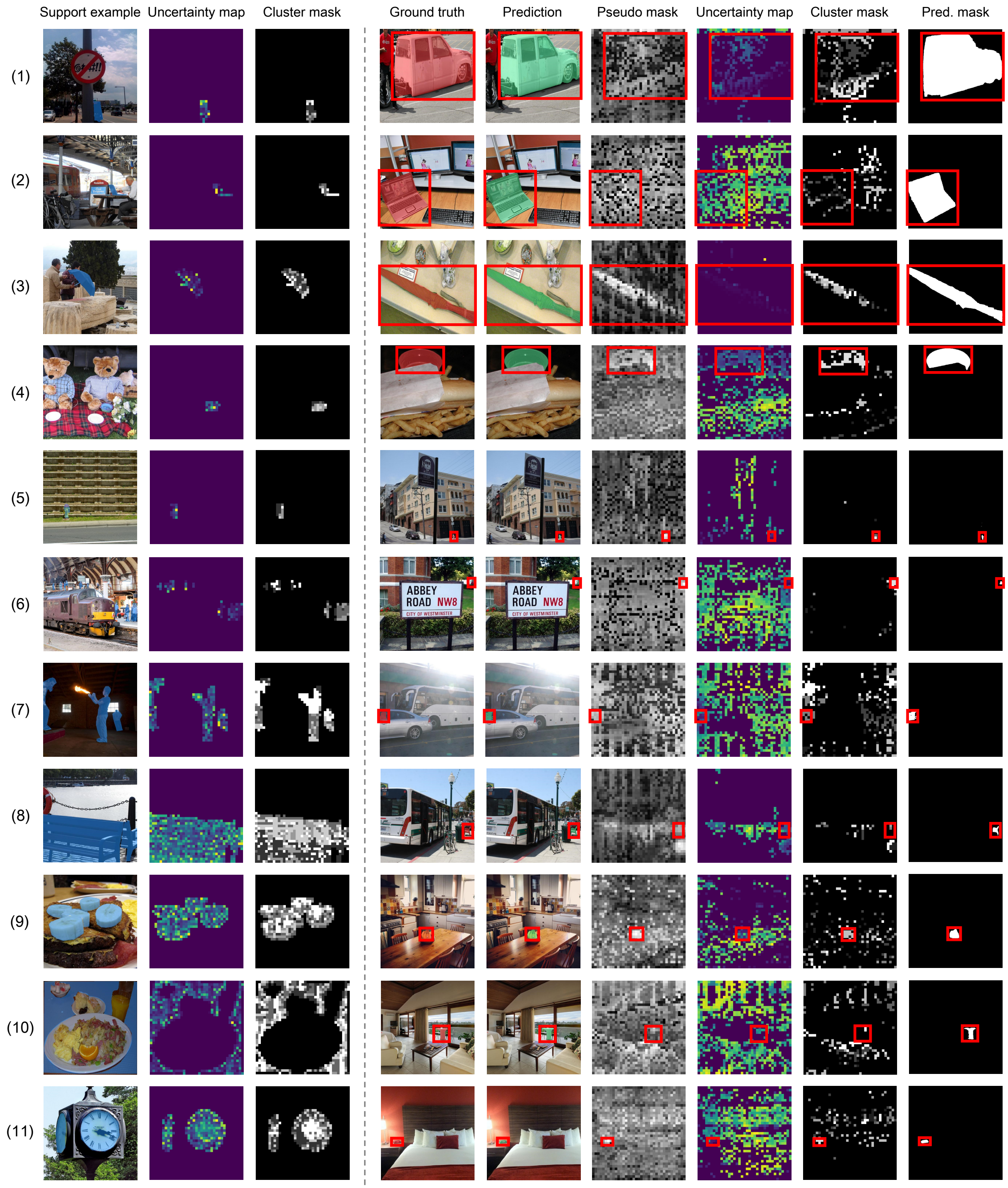

Figure E1. Visualization of one-shot semantic segmentation. The blue area denotes the mask for in-context prompts, the red area represents the ground-truth mask, and the green area indicates the model's predictions. (If there is no additional explanation, the subsequent visualizations will follow this setting.) We also performed a detailed visualization of the network's intermediate outputs, including the pseudo masks, uncertainty maps, and clustered masks. To better observe the area of interest, we use red boxes □ to mark the corresponding position. Please zoom in for a better view.

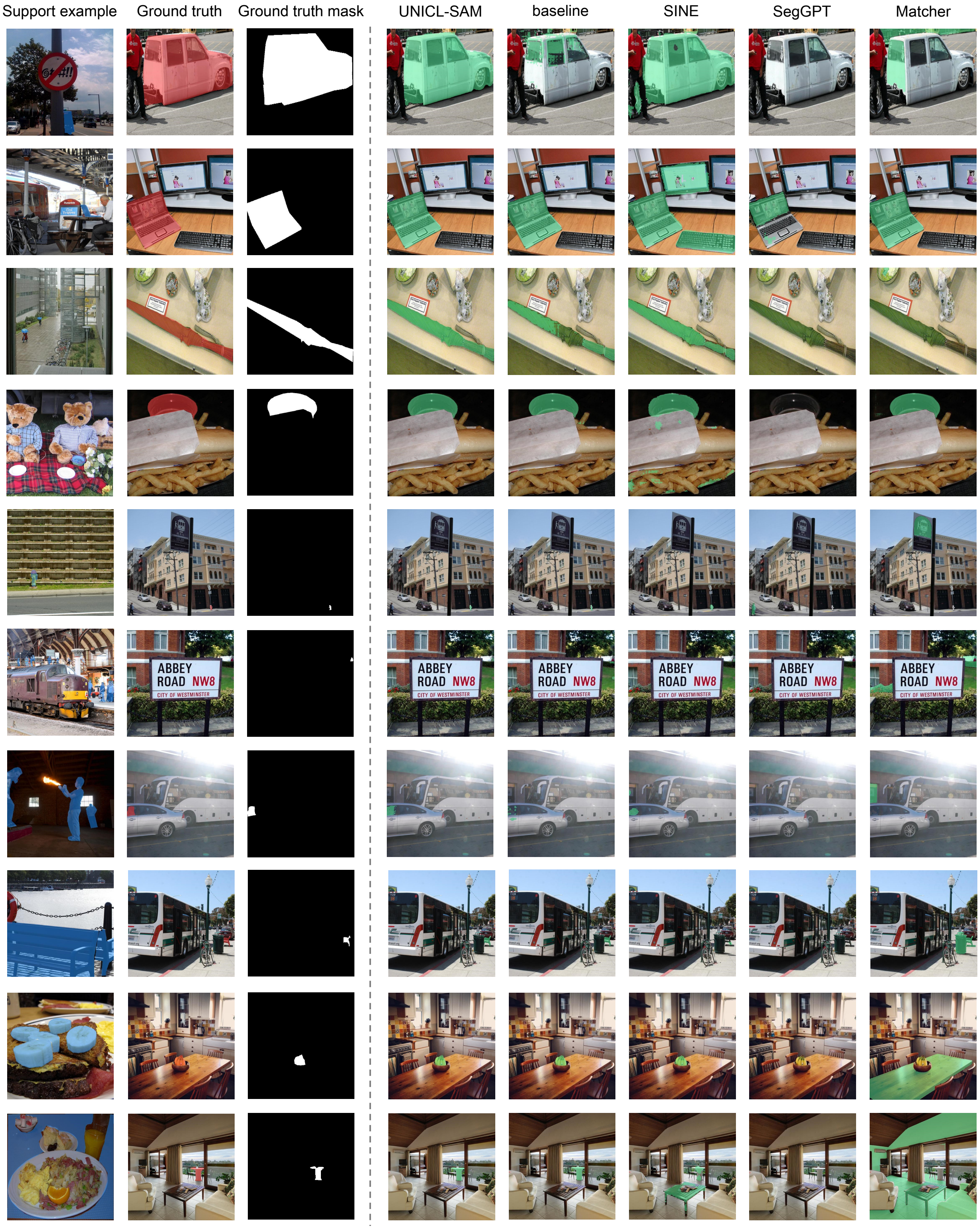

Figure E2. Visualization comparisons between advanced in-context segmentation generalists. Our UNICL-SAM effectively addresses complex and challenging scenarios that previous approaches have struggled to manage, demonstrating robust in-context semantic segmentation capabilities. Please zoom in for a better view.

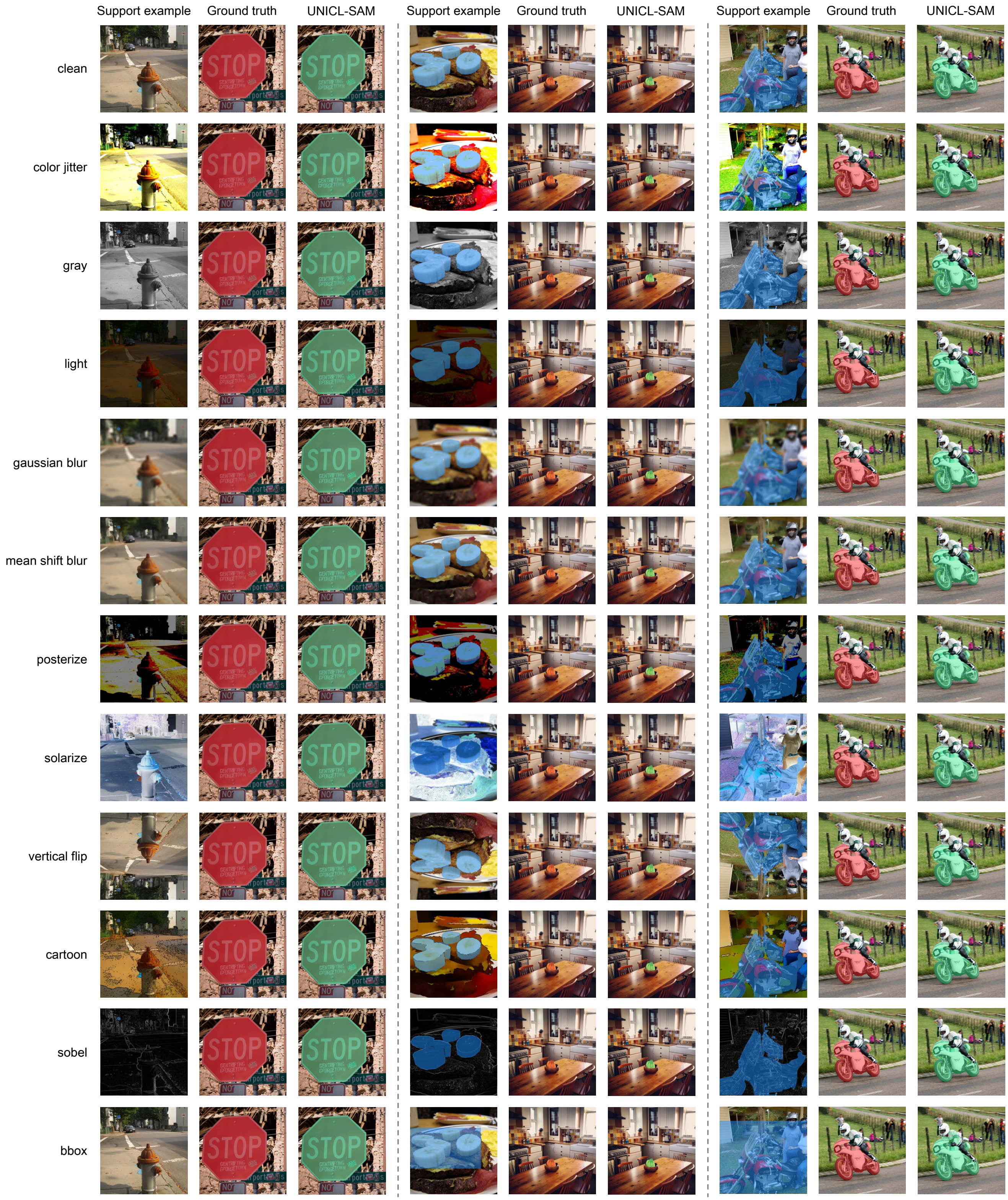

Figure E3. Qualitative results on robustness tests. The corresponding transformation type is indicated on the left. UNICL-SAM consistently achieves accurate segmentation results across various types of degradation and domain shift variations that we designed for testing, demonstrating a notable degree of robustness. Please zoom in for a better view.