Supplementary Material of RobustSAM: Segment Anything Robustly on Degraded Images

Wei-Ting Chen[1,2†] Yu-Jiet Vong[1] Sy-Yen Kuo[1] Sizhou Ma[2*] Jian Wang[2*]

[1]National Taiwan University [2]Snap Inc.

1. More Experiments

1.1. Ablation Study

We conducted comprehensive ablation studies on the BDD-100k [23] and LIS [2, 22] datasets, along with the subset of the COCO [17] dataset incorporated within the Robust-Seg dataset collection. It is important to highlight that these datasets are unseen (zero-shot) and with real-world degradations in our evaluations. The results, as detailed in Table 1, underscore that every component integrated into RobustSAM contributes positively to its overall performance. This consistent enhancement across various datasets demonstrates the robustness and adaptability of RobustSAM, particularly in zero-shot settings.

1.2. Different Backbones in RobustSAM

In Table 2, we showcase a thorough comparison of SAM and RobustSAM across various Vision Transformer (ViT) [4] backbones, including ViT-B, ViT-L, and ViT-H. This comparison encompasses the numerical results on the combined BDD-100k [23] and LIS [2, 22] datasets and extends to the COCO [17] dataset. The data clearly illustrate that RobustSAM consistently outperforms SAM across these diverse backbones. Furthermore, in Figure 1, we provide a comprehensive comparison of performance, inference speed, and model size among various SAM and RobustSAM variants, offering additional insights into the efficiency and scalability of these models. Notably, our models effectively enhance performance in degraded scenarios with only a marginal increase in computational burden.

1.3. Comparison of Varying Number of Point Prompts

To examine the interactive segmentation performance of RobustSAM using point prompts, we have conducted a comprehensive comparison in Figure 2. This comparison assesses RobustSAM against SAM with a range of input point numbers on the BDD-100k [23] and LIS [2, 22] datasets in a zero-shot learning context. RobustSAM consistently achieves superior performance throughout these datasets compared to SAM, irrespective of the number of input points used.

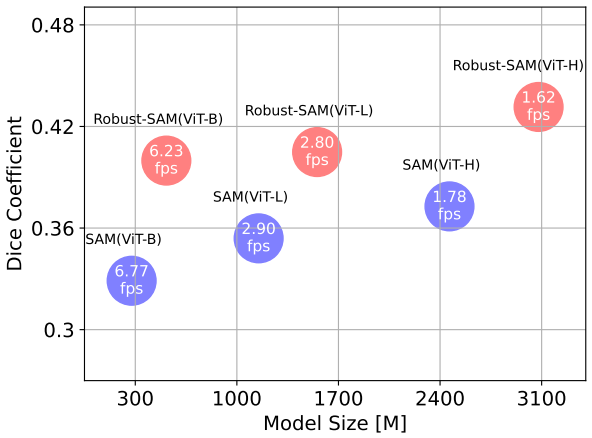
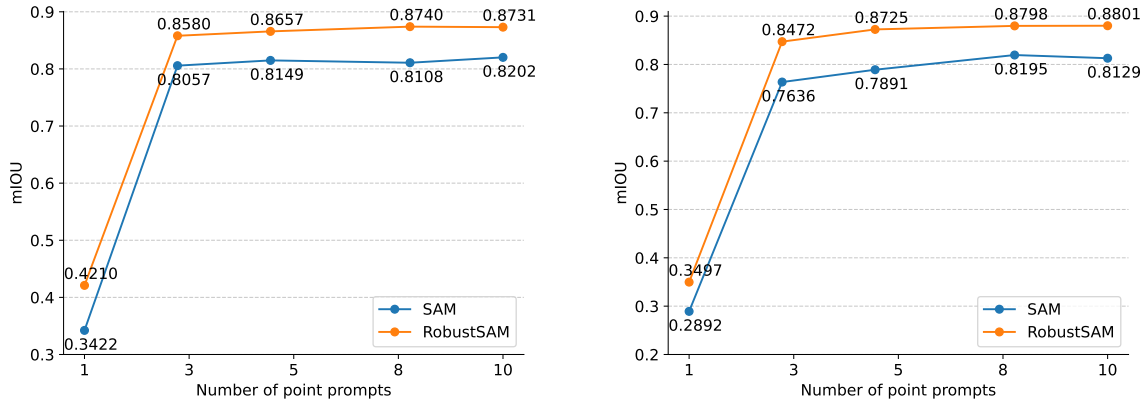


Figure 1. **Comparison of performance, speed, and model size among various SAM and RobustSAM variants.** The suffixes -B, -L, and -H correspond to ViT-B (Base), ViT-L (Large), and ViT-H (Huge) versions, respectively, representing different scales and complexities of the Vision Transformer architecture.

1.4. Token Visualization

In Figure 3, we provide an illustrative comparison of cross-attention in the last token-to-image layer of the mask decoder between SAM's original output token and the enhanced Robust Output Token of RobustSAM. This comparison underscores the ability of the Robust Output Token to achieve more focused and precise attention. It excels in accurately identifying object boundaries and contents, an aspect often overlooked by SAM's original output token in

† Part of the work done during internship at Snap Research.
* Co-corresponding authors



mIoU on BDD-100k dataset

mIoU on LIS dataset

Figure 2. **Comparative analysis of interactive segmentation performance using different numbers of input points on the BDD-100k [23] and LIS [2, 22] datasets in a zero-shot setting.** RobustSAM consistently surpasses SAM across diverse point quantities, exhibiting a more pronounced enhancement, especially in scenarios with reduced prompt ambiguity on the BDD-100k dataset.

Module	BDD-100k+LIS		COCO			
	IoU	PA	AP	AP_S	AP_M	AP_L
Baseline						
SAM	0.3056	0.8911	0.5002	0.3168	0.4292	0.5243
SAM-Finetune	0.1871	0.7691	0.1321	0.0384	0.1211	0.1731
SAM-Finetune Decoder	0.2476	0.8691	0.1457	0.0391	0.1322	0.1868
SAM-Finetune Output Token	0.3194	0.9036	0.4853	0.3011	0.4312	0.5266
RobustSAM						
w AMFG	0.3455	0.9059	0.5021	0.3147	0.4295	0.5273
w AMFG-F	0.3535	0.9120	0.5045	0.3150	0.4336	0.5370
w AMFG-F+AOTG	0.3651	0.9193	0.5075	0.3161	0.4349	0.5381
w AMFG-F+AOTG+ROT (**ALL**)	**0.3717**	**0.9226**	**0.5130**	**0.3192**	**0.4416**	**0.5518**

Table 1. **Efficacy of Proposed Modules:** An evaluation of the BDD-100k [23], LIS [2, 22], and COCO [17] datasets reveals that each of the proposed modules enhances the performance of RobustSAM. (We use point prompts for BDD-100k+LIS and bounding box prompts for COCO in this comparison.)

Model	BDD-100k+LIS		COCO			
	IoU	PA	AP	AP_S	AP_M	AP_L
SAM-B	0.3003	0.8826	0.4589	0.2958	0.3840	0.4752
RobustSAM-B	**0.3317**	**0.8972**	**0.4710**	**0.2961**	**0.4175**	**0.5268**
SAM-L	0.3056	0.8911	0.5002	0.3168	0.4292	0.5243
RobustSAM-L	**0.3717**	**0.9226**	**0.5130**	**0.3192**	**0.4416**	**0.5518**
SAM-H	0.3384	0.9305	0.5087	0.3184	0.4430	0.5255
RobustSAM-H	**0.3813**	**0.9367**	**0.5167**	**0.3188**	**0.4455**	**0.5697**

Table 2. **Performance comparison between SAM and RobustSAM across different Vision Transformer (ViT) backbones.**

degraded scenes. This focused attention is particularly evident in its handling of the object's boundaries, demonstrating RobustSAM's enhanced capability to discern and highlight details and structures, crucial for effective segmentation in challenging imaging conditions.

1.5. Visualization of Feature Representation

We conducted an experiment by randomly sampling 50 images for each of the six degradations from our dataset. We ran SAM, RobustSAM w/o consistency loss and RobustSAM to extract the mask features and performed t-

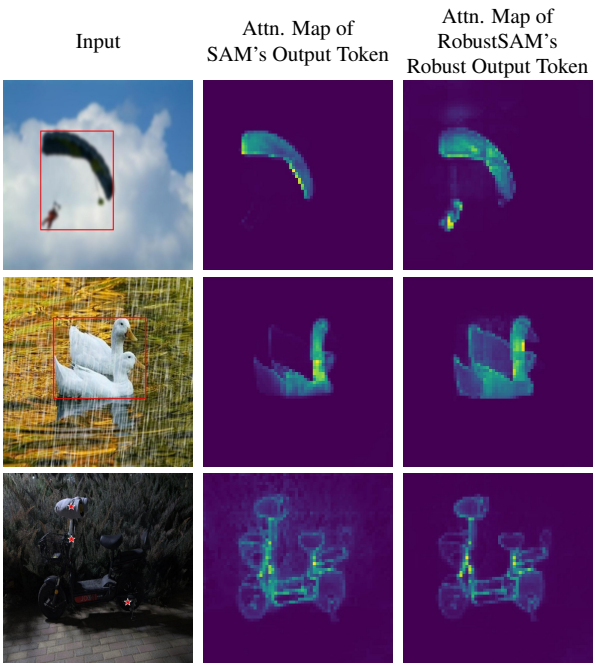


Figure 3. **Comparison of cross-attention in the final decoder layer between SAM's original output token and the enhanced Robust Output Token.** The Robust Output Token distinctly exhibits a precise focus on accurately identifying object boundaries and contents. It further demonstrates attention to the boundary and thin structural regions, which are typically overlooked by the original output token.

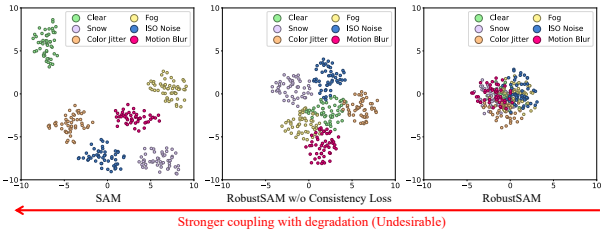


Figure 4. **Visualization of feature representation based on different baselines.**

SNE analysis. The results shown in Figure 4 indicate that in the original SAM, features from the same degradation type tended to cluster together. However, with RobustSAM's feature suppression mechanism, features from different degradations significantly overlap each other, suggesting the minimal influence of degradation on feature extraction. Moreover, when consistency loss is not used, the clustering due to same degradation is more evident compared to RobustSAM, demonstrating the effectiveness of the consistency loss.

SA-1B	SAM	RobustSAM
IoU/PA	0.6917/0.8719	**0.7065/0.8726**

Table 3. **Zero-shot segmentation comparison on a subset of SA-1B dataset [10].**

1.6. Qualitative Evaluation

In our qualitative evaluation, showcased in Figure 5, we present a comprehensive set of results illustrating the efficacy of our approach in both degraded and clear (Row 3 and 8) scenarios. We adopt original SAM [10], HQ-SAM [9], Air-Net [11] + SAM, and URIE [20] + SAM in this comparison. These visual comparisons clearly demonstrate the superior segmentation capabilities of our method under various conditions. Notably, our approach maintains high accuracy and detail in degraded scenes, where existing methods often struggle, effectively segmenting intricate patterns and structures. Moreover, our method can maintain the performance in clear conditions.

1.7. Quantitative Evaluation

In our quantitative evaluation, we expanded the baseline comparisons, detailed in Tables 4, 5, and 6. Our focus was on enhancing segmentation accuracy by preprocessing images with restoration methods like MW-Net [18], SwinIR [14], and MPR-Net [24] before applying the SAM technique. This approach was rigorously tested on the BDD-100k [23] and LIS [2, 22] datasets, as well as on subsets of unseen datasets with synthetic degradations, namely COCO [17], NDD20 [21], STREETS [19], and FSS-1000 [13], all part of the Robust-Seg dataset collection.

Furthermore, we extended our experiments to include fine-tuning of SwinIR, MW-Net, and Air-Net using our degraded-clear image pairs, followed by SAM application (referred to as SwinIR-F, MW-Net-F, and Air-Net-F). We also fine-tuned HQ-SAM with our training data, denoted as HQ-SAM-F. The findings indicate a marginal performance improvement through fine-tuning, but these adaptations still do not match the effectiveness of our proposed method. This underscores the robustness and superiority of our approach, particularly in achieving high segmentation accuracy under various degrees of image degradation.

Moreover, we evaluated the performance of RobustSAM on a specific subset of the SA-1B dataset [10], comprising 11,186 images. This evaluation was conducted in comparison with the standard SAM [10] method. The outcomes, presented in Table 3, clearly indicate that RobustSAM outperforms SAM, demonstrating the efficacy of the proposed RobustSAM approach.

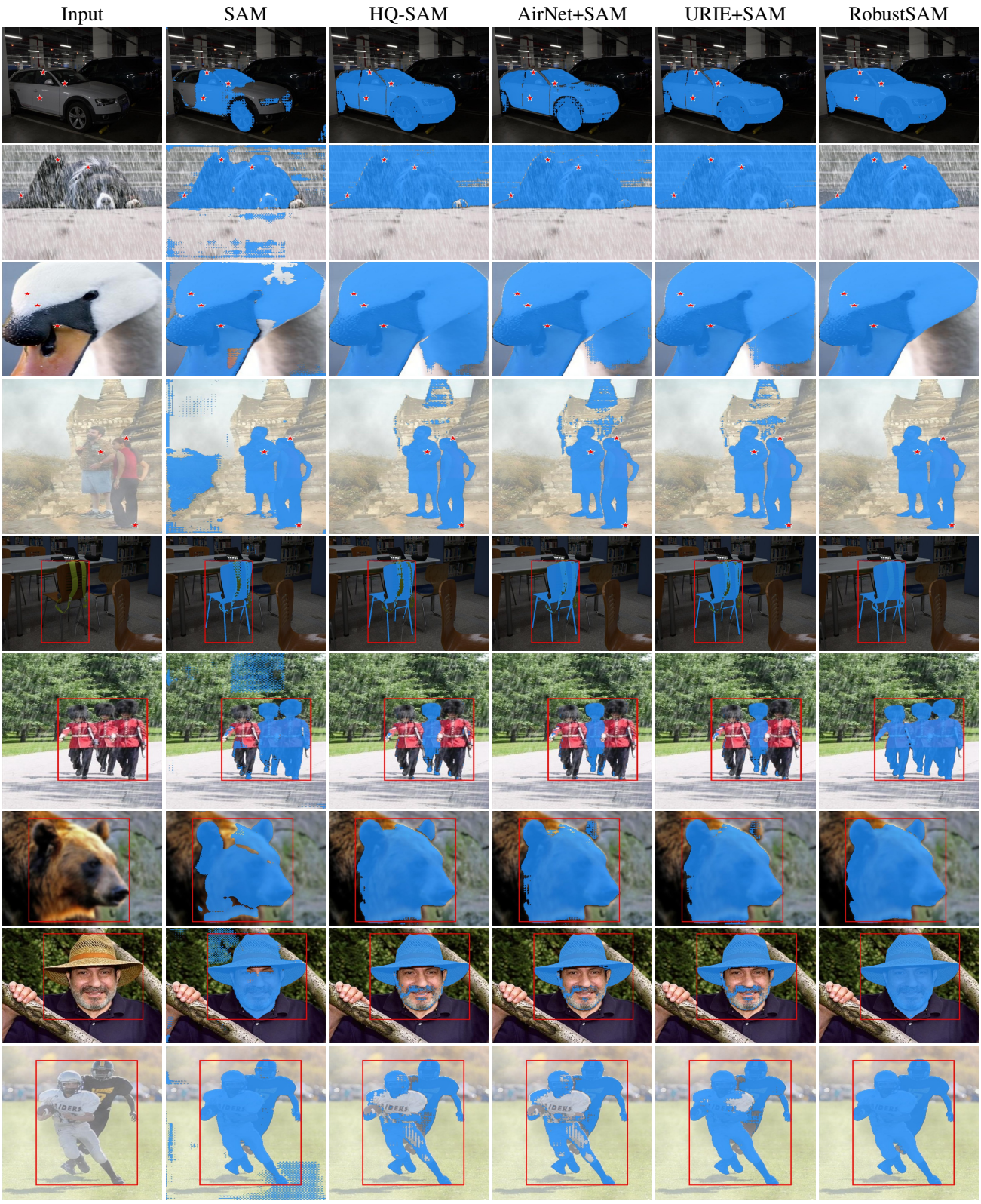


Figure 5. **Qualitative Analysis of Segmentation:** This figure offers a visual comparison to illustrate the enhanced performance of Robust-SAM compared to current methods. Notably, Rows 3 and 8 depict scenes without degradations..

Method	Point		Bounding Box	
	IoU	Dice	IoU	Dice
SAM	0.3056	0.3837	0.8808	0.9171
HQ-SAM	0.2943	0.3712	0.8877	0.9245
HQ-SAM-F	0.2951	0.3720	0.8906	0.9406
AirNet+SAM	0.3245	0.4550	0.8776	0.9129
AirNet-F+SAM	0.3251	0.4583	0.8781	0.9133
MPR-Net+SAM	0.3271	0.4079	0.8758	0.9174
SwinIR+SAM	0.3294	0.4117	0.8677	0.9241
SwinIR-F+SAM	0.3313	0.4120	0.8683	0.9253
MW-Net+SAM	0.3195	0.4000	0.8898	0.9379
MW-Net-F+SAM	0.3199	0.4006	0.8908	0.9387
URIE+SAM	0.3042	0.3828	0.8799	0.9165
RobustSAM	**0.3717**	**0.8926**	**0.8958**	**0.9416**

Table 4. **Zero-Shot Segmentation Comparison:** This figure presents a comparison of segmentation performance on the entire BDD-100k [23] and LIS [2, 22] datasets, which are unseen datasets with real-world degradations. We utilized point and bounding box prompts for segmentation. The notation '+SAM" indicates the process of first restoring the image, followed by applying SAM for segmentation, consistent with the methodology described in Tables 5 and 6.

Method	Degrade		Clear		Average	
	IoU	PA	IoU	PA	IoU	PA
SAM	0.7981	0.9555	0.8295	0.9707	0.8000	0.9565
HQ-SAM	0.8079	0.9617	0.8448	0.9756	0.8102	0.9625
HQ-SAM-F	0.8082	0.9620	0.8452	0.9760	0.8106	0.9630
AirNet+SAM	0.7988	0.9629	0.8312	0.9752	0.8008	0.9637
AirNet-F+SAM	0.7992	0.9635	0.8316	0.9760	0.8083	0.9640
MPR-Net+SAM	0.7969	0.9585	0.8227	0.9712	0.7985	0.9593
SwinIR+SAM	0.7951	0.9543	0.8210	0.9701	0.7971	0.9580
SwinIR-F+SAM	0.7956	0.9553	0.8255	0.9713	0.7983	0.9592
MW-Net+SAM	0.7713	0.9432	0.8183	0.9692	0.7813	0.9491
MW-Net-F+SAM	0.7722	0.9440	0.8192	0.9701	0.7830	0.9501
URIE+SAM	0.7904	0.9593	0.8288	0.9740	0.7928	0.9602
RobustSAM	**0.8195**	**0.9778**	**0.8529**	**0.9817**	**0.8216**	**0.9780**

Table 5. **Zero-shot segmentation comparison on the whole NDD20 [21], STREETS [19], and FSS-1000 [13] (unseen datasets with synthetic degradations) in Robust-Seg dataset using point prompts.**

1.8. Improving SAM-prior Tasks

We validate the effectiveness of RobustSAM in enhancing downstream tasks (*i.e.,* dehazing [7] and deblurring [12]) that use SAM as a prior. Our results, showcased in Figure 6 and Figure 7, include both the segmentation masks and the reconstructed outcomes for dehazing and deblurring tasks. These results clearly demonstrate that employing RobustSAM as a prior significantly boosts the performance of these tasks. This enhancement is particularly evident in degraded scenarios where RobustSAM maintains its strong segmentation capabilities. By reliably segmenting images even in challenging conditions, RobustSAM provides a robust foundation for subsequent image restoration tasks, leading to improved overall outcomes in both clarity and detail.

Method	Performance Metrics			
	AP	AP_S	AP_M	AP_L
SAM	0.5002	0.3168	0.4292	0.5243
HQ-SAM	0.5052	0.2920	0.4267	0.5517
HQ-SAM-F	0.5063	0.2925	0.4272	**0.5518**
AirNet+SAM	0.4926	0.3068	0.4263	0.5187
AirNet-F+SAM	0.4933	0.3075	0.4272	0.5203
MPR-Net+SAM	0.4986	0.3133	0.4301	0.5227
SwinIR+SAM	0.4911	0.3027	0.4211	0.5195
SwinIR-F+SAM	0.4923	0.3038	0.4219	0.5201
MW-Net+SAM	0.5027	0.3161	0.4354	0.5290
MW-Net-F+SAM	0.5033	0.3165	0.4362	0.5294
URIE+SAM	0.4980	0.3186	0.4319	0.5215
RobustSAM	**0.5130**	**0.3192**	**0.4416**	**0.5518**

Table 6. **Zero-shot segmentation comparison on the whole COCO [17] (unseen datasets with synthetic degradations) in Robust-Seg dataset using Bounding Box prompts.**

2. Robust-Seg Dataset

The meticulously curated Robust-Seg dataset, designed to train and evaluate the RobustSAM model, encapsulates a rich repository of 43,000 images with corresponding annotated masks. These images are sourced from a suite of renowned datasets, namely LVIS [5], ThinObject-5k [15], MSRA10K [3], NDD20 [21], STREETS [19], FSS-1000 [13], and COCO [17]. We have augmented this collection with 15 types of synthetic alterations using the albumentations [1] and imgaug [8] libraries, introducing a diverse range of visual degradations to the dataset. The degradations include snow, fog, rain, Gaussian noise, ISO noise, impulse noise, re-sampling blur, motion blur, zoom blur, color jitter, compression artifacts, elastic transformation, frosted glass blur, low light, and contrast adjustments. Alongside these, one augmentation category is designated for images without any modifications, preserving their original clarity.

These synthetic degradations are meant to simulate a breadth of challenging visual scenarios, thereby extending the robustness of the model against a spectrum of image qualities. This strategic augmentation process yields 688,000 image-mask pairs, significantly expanding the dataset's volume and variety. The specific quantities of images drawn from each contributing dataset are detailed in Table 7.

For a visual demonstration of the augmented images and their varied degradations, please refer to Figure 8, where we showcase the synthetic effects introduced to the dataset.

	LVIS [5]	ThinObject-5k [15]	MSRA10K [3]	NDD20 [21]	STREETS [19]	FSS-1000 [13]	COCO [17]	**Total**
Image Number	20252	4748	10000	1000	1000	1000	5000	43000

Table 7. **Data composition of our constructed Robust-Seg dataset.**

Figure 6. **Enhancing SAM-based Dehazing Method:** A qualitative demonstration of RobustSAM's superiority in refining the SAM-based single image dehazing.

Figure 7. **Enhancing SAM-based Deblurring Method:** A qualitative demonstration of RobustSAM's superiority in refining the SAM-based single image deblurring.

3. Implementation Details

3.1. Network Architecture

During training RobustSAM on the composed Robust-Seg dataset, we fix the model parameters of the pre-trained SAM model (gray blocks in Fig. 2 of the main paper) while only making the proposed RobustSAM learnable, including Robust Output Token (ROT), Anti-Degradation Output Token Generation (AOTG) module, Anti-Degradation Mask Feature Generation (AMFG) module and a three-layer MLP which is used to generate the final robust mask. The code will be made publicly available.

AMFG module. The AMFG module first passes the input feature through the Instance Normalization (IN) and Batch Normalization (BN) layers, respectively. The ReLU activation function is applied to the normalized features. Then,

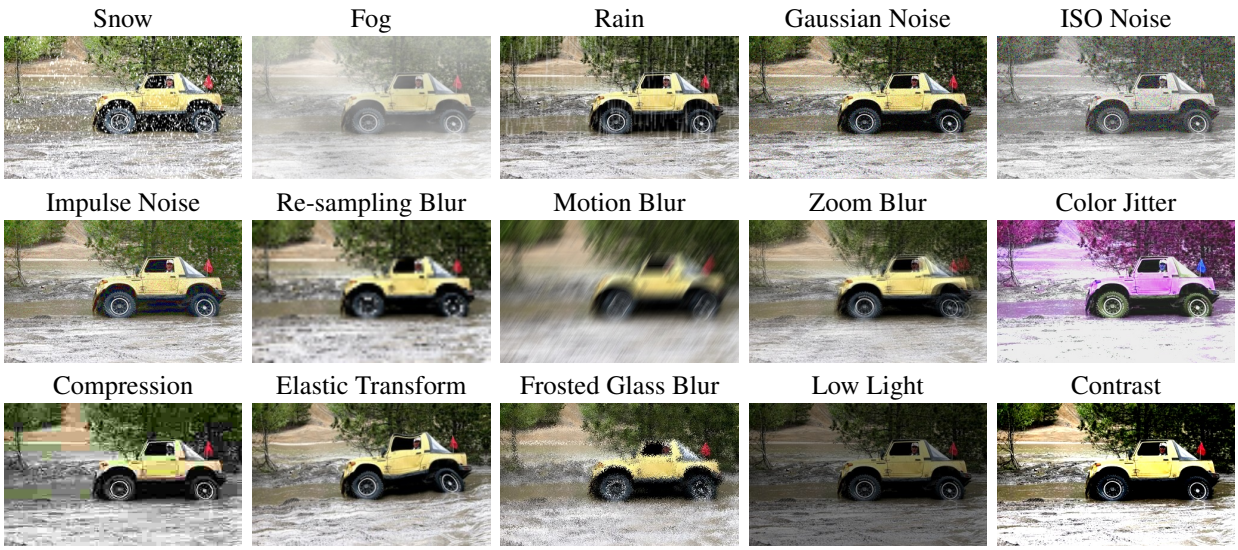


Figure 8. **Illustrative samples of images with synthetic degradations from the Robust-Seg dataset.**

we convolve both output features using a 3×3 convolution layer with padding one and sum them up together. Next, a selector network will be used to generate two attention maps (which have the same shape) based on the summed feature.

There are several steps in the selector network to generate attention maps. First of all, we utilize an adaptive average pooling layer to reshape the input feature. The reshaped feature is then processed using a fully connected layer, followed by the ReLU activation function. Next, two fully connected layers adjust the dimensions of the input features and generate two unnormalized attention maps. Both attention maps are stacked, and the Softmax function is applied to normalize the attention maps between 0 and 1. Lastly, the normalized attention maps will perform element-wise multiplication with the input features of the selector network.

Following the aforementioned process, we can obtain an enhanced feature. To compensate for any semantic information that may have been lost, this enhanced feature is concatenated with the original input features along the channel dimension. Then, we choose the squeeze-and-excitation [6] approach to refine the concatenated feature.

The output feature of the squeeze-and-excitation module will then be transformed using the Fourier transform to obtain phase components and amplitude components, respectively. After that, we apply a 1×1 convolution with zero padding and stride one on the amplitude components to remove degradation elements. Next, an inverse Fourier transform is performed to restore the refined features to their original spatial representation.

Finally, a combination of two transposed convolution layers with 2×2 kernels and stride two is applied to align the dimension and generate the final output feature of the AMFG module.

AOTG module. On the other hand, the AOTG module consists of two IN layers and an MLP network. The original robust output token will first pass through the IN layers to filter out information sensitive to degradation-related details. After that, an MLP is applied to adjust the dimension of the robust output token. There are two fully connected layers inside the MLP, with a ReLU activation function between them.

3.2. Prompt Generation

We follow the same inference pipeline of SAM but use the mask prediction from robust output token as the final mask prediction. For box-prompting-based evaluation, we utilize the ground truth mask to generate four corner coordinates of the bounding box. Then, the coordinates are used as the box prompt to feed into our RobustSAM model. For point-prompting-based evaluation, we randomly sample the points from the ground truth masks and use them as the input point prompts.

3.3. Evaluation Protocol

We employ several metrics to assess our model's performance:

Intersection over Union (IoU) is a common metric used to measure segmentation accuracy on a particular dataset. It is defined as the area of overlap between the predicted and ground truth segmentation divided by the area of union between the predicted and ground truth segmentation. The IoU metric is given by:

$$\text{IoU} = \frac{|P \cap G|}{|P \cup G|} \tag{1}$$

where P represents the set of pixels in the predicted segmentation, and G is the set of pixels in the ground truth segmentation. Higher IoU values indicate better segmentation accuracy.

Dice Coefficient (Dice) [16], often referred to as the Dice Similarity Coefficient (DSC), is a statistical tool that measures the similarity between two sets of data. In the context of image segmentation, it quantifies the similarity between the predicted segmentation and the ground truth. The Dice Coefficient is calculated as follows:

$$\text{Dice} = \frac{2|P \cap G|}{|P| + |G|} \tag{2}$$

where $|P \cap G|$ represents the common elements (overlapping pixels) between the predicted and ground truth sets, and $|P|$ and $|G|$ are the total elements in each set, respectively. Like IoU, a higher Dice score indicates greater similarity between the predicted and actual segmentations.

Pixel Accuracy (PA) quantifies the proportion of correctly classified pixels in an image. It is calculated as follows:

$$\text{PA} = \frac{\sum_{i=1}^{N} \mathbb{1}(p_i = g_i)}{N} \tag{3}$$

where N is the total number of pixels in the image, p_i represents the predicted class of pixel i, and g_i is the ground truth class of pixel i. The function $\mathbb{1}(p_i = g_i)$ is an indicator function that equals 1 when the predicted class of a pixel matches its ground truth class and 0 otherwise. The numerator sums up all instances where the predicted and actual classes of pixels are the same, and the denominator is the total number of pixels. A higher PA indicates better accuracy of the model in classifying each pixel.

Average Precision (AP) measures the average of precision scores at different thresholds. It is a way to summarize the precision-recall curve into a single value representing overall segmentation accuracy. AP is calculated as follows:

$$\text{AP} = \frac{1}{N_{\text{thres}}} \sum_{t=1}^{N_{\text{thres}}} \text{Precision}(t) \tag{4}$$

where N_{thres} is the total number of threshold levels used, and $\text{Precision}(t)$ is the precision of the segmentation at a specific threshold t. In practice, AP is computed by taking the average of precision values calculated at several predetermined threshold levels. These levels typically range from 0 to 1, indicating the probability threshold at which a pixel is classified as part of the segmented object. A higher AP value indicates a better model performance across various threshold levels.

References

[1] Alexander Buslaev, Vladimir I. Iglovikov, Eugene Khvedchenya, Alex Parinov, Mikhail Druzhinin, and Alexandr A. Kalinin. Albumentations: Fast and flexible image augmentations. *Information*, 2020. 5

[2] Linwei Chen, Ying Fu, Kaixuan Wei, Dezhi Zheng, and Felix Heide. Instance segmentation in the dark. *IJCV*, 2023. 1, 2, 3, 5

[3] Ming-Ming Cheng, Niloy J. Mitra, Xiaolei Huang, Philip H. S. Torr, and Shi-Min Hu. Global contrast based salient region detection. *IEEE TPAMI*, 2015. 5, 6

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1

[5] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019. 5, 6

[6] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018. 7

[7] Zheyan Jin, Shiqi Chen, Yueting Chen, Zhihai Xu, and Huajun Feng. Let segment anything help image dehaze. *arXiv preprint arXiv:2306.15870*, 2023. 5

[8] Alexander B. Jung, Kentaro Wada, Jon Crall, Satoshi Tanaka, Jake Graving, Christoph Reinders, Sarthak Yadav, Joy Banerjee, Gábor Vecsei, Adam Kraft, Zheng Rui, Jirka Borovec, Christian Vallentin, Semen Zhydenko, Kilian Pfeiffer, Ben Cook, Ismael Fernández, François-Michel De Rainville, Chi-Hung Weng, Abner Ayala-Acevedo, Raphael Meudec, Matias Laporte, et al. imgaug. https://github.com/aleju/imgaug, 2020. Online; accessed 01-Feb-2020. 5

[9] Lei Ke, Mingqiao Ye, Martin Danelljan, Yifan Liu, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. Segment anything in high quality. *arXiv:2306.01567*, 2023. 3

[10] 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

[11] Boyun Li, Xiao Liu, Peng Hu, Zhongqin Wu, Jiancheng Lv, and Xi Peng. All-in-one image restoration for unknown corruption. In *CVPR*, 2022. 3

[12] Siwei Li, Mingxuan Liu, Yating Zhang, Shu Chen, Haoxiang Li, Hong Chen, and Zifei Dou. Sam-deblur: Let segment anything boost image deblurring. *arXiv preprint arXiv:2309.02270*, 2023. 5

[13] Xiang Li, Tianhan Wei, Yau Pun Chen, Yu-Wing Tai, and Chi-Keung Tang. Fss-1000: A 1000-class dataset for few-shot segmentation. *CVPR*, 2020. 3, 5, 6

[14] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *ICCV*, 2021. 3

[15] Jun Hao Liew, Scott Cohen, Brian Price, Long Mai, and Jiashi Feng. Deep interactive thin object selection. In *WACV*, 2021. 5, 6

[16] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 8

[17] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 1, 2, 3, 5, 6

[18] Prashant W. Patil, Sunil Gupta, Santu Rana, Svetha Venkatesh, and Subrahmanyam Murala. Multi-weather image restoration via domain translation. In *ICCV*, 2023. 3

[19] Corey Snyder and Minh Do. Streets: A novel camera network dataset for traffic flow. *NIPS*, 2019. 3, 5, 6

[20] Taeyoung Son, Juwon Kang, Namyup Kim, Sunghyun Cho, and Suha Kwak. Urie: Universal image enhancement for visual recognition in the wild. In *ECCV*, 2020. 3

[21] Cameron Trotter, Georgia Atkinson, Matt Sharpe, Kirsten Richardson, A Stephen McGough, Nick Wright, Ben Burville, and Per Berggren. Ndd20: A large-scale few-shot dolphin dataset for coarse and fine-grained categorisation. *arXiv preprint arXiv:2005.13359*, 2020. 3, 5, 6

[22] Hong Yang, Wei Kaixuan, Chen Linwei, and Fu Ying. Crafting object detection in very low light. In *BMVC*, 2021. 1, 2, 3, 5

[23] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, Trevor Darrell, et al. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv preprint arXiv:1805.04687*, 2(5):6, 2018. 1, 2, 3, 5

[24] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *CVPR*, 2021. 3