ROUGE: LEARNING GATED EXPERTS FOR SEGMENT ANYTHING IN THE WILD

Anonymous authors
Paper under double-blind review

ABSTRACT

Segment anything model (SAM) and its variants have recently shown promising performance as foundation models. However, existing SAM-based models can only handle scenarios seen during training, and usually suffer unstable performance when transferring to real-world unseen data, such as low-light, rainy or blurred images, which is crucial for applications such as autopilot. Therefore, adapting SAM-based models for real-world degradation while not impairing its original ability remains an open challenge. In this work, we propose a novel gated Mixture-of-Experts (MoE) structure, called RouGE, to improve the robustness of SAM-based models. Specifically, RouGE uses multiple lightweight probability gates to decompose complex real-world image conditions and judge whether the feature needs to be adjusted as well as to what extent the adjustment needs to be done, then handle them differently with a set of low-rank experts. During the inference stage, RouGE processes input images in a completely blind manner thus improving the model's performance in real-world scenarios. Extensive experiments demonstrate that RouGE consistently achieves state-of-the-art results on both degraded and clean images compared with other methods while tuning only 1.5% of parameters.

1 INTRODUCTION

Segment anything model (SAM) (Kirillov et al., 2023; Ravi et al., 2024) and its variants have recently shown impressive performance and have been widely applied in various downstream applications, *e.g.*, autopilot and medical image segmentation. However, existing SAM-based methods are usually trained on clean images without degradation. Given that degradations such as low light, rain and blur are almost unavoidable in real-world scenarios, existing models consequently suffer unstable performance when transferring to real-world unseen data. Therefore, how to improve the robustness of SAM-based models to deal with real-world diverse scenarios poses an open challenge.

To allow for a robust SAM-based model for real-world applications, several methods have been explored. For instance, one simple solution may use a two-step workflow with another image restoration model (Li et al., 2022; Wang et al., 2024a; Potlapalli et al., 2023) before segmentation to remove undesired degradations. Such methods rely heavily on the reconstruction results of pretrained image restoration models and cannot handle various types of degradation (*e.g.*, noise, blur or rain). More importantly, restored images may not benefit high-level visual tasks as they are initially designed for human eyes and may generate artifacts which has negative impacts on downstream tasks (Cui et al., 2021; Chen et al., 2024). To relieve such issues, other methods involve fine-tuning segmentation models that are tailored for specific degradation (Cui et al., 2021; Chen et al., 2023b). However, fine-tuning on specific degradation requires prior knowledge of the degradation type of input image, which is hard to achieve in real-world applications. Recently, Chen et al. (2024) proposed RobustSAM using a post-processing module to handle real degradation, but still suffers from heavy computational cost by increasing the model's parameters by about 32%.

Despite attempts have been made to obtain robust SAM models, current methods still face the following challenges. First, fine-tuning the foundation model can inherently degrade its original performance and lead to catastrophic forgetting problems. Second, the diversity of real-world degradation leads to significant variations in degradation types and a robust model needs to handle various degradation and clean images in a completely blind manner. Third, manually labeled real-world

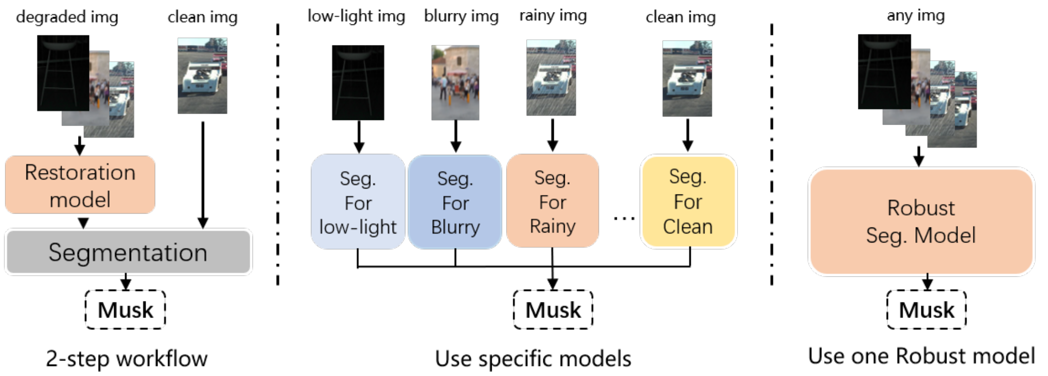

Figure 1: Comparison of Three Methods. Both the 2-step approach and using specialized models approach require obtaining prior knowledge of image categories for corresponding processing, and their workflow is complex. A robust model that can directly handle all types of images is evidently the optimal solution. Therefore, we propose a novel approach to empower a less-robust SAM-based model to become a robust model.

degradation data is scarce, and practical applications often require specific scene and degradation types, complicating model training further.

To maintain the original performance of the model, a good way is to minimize changes applied to the pretrained model weights with the Parameter Efficient Fine-Tuning (PEFT) techniques. By fine-tuning with almost all model parameters frozen, PEFT methods can adapt models to new domains while preserving the model's generalization ability at marginal cost and are widely used in both vision and natural language fields (Houlsby et al., 2019; Jie & Deng, 2022; Chen et al., 2022; Pfeiffer et al., 2020a; Wang et al., 2022; Yu et al., 2024). To tackle diverse data types, a natural approach is to break down complex tasks into multiple simpler tasks. By using mixture-of-expert (MoE)-like methods, we can decompose the complexity of real-world environments into multiple conditions, enabling us to use a set of smaller modules to solve the complex problem. Considering the third point, training with unlabeled images can better reduce the challenges of industrial applications.

Based on the above observations, we introduce a novel module **RouGE**, a plug-in **Ro**bustness-**U**plift module using **G**ated **E**xperts to perform differentiated processing on degraded and non-degraded inputs within a pretrained network. RouGE module comprises lightweight multiple probability gates and their corresponding low-rank experts (including lazy and trainable experts), to efficiently select suitable experts for input data and perform effective combined processing. The probability gates provide the model with interpretable classification capabilities to handle blind input, while the design of lazy and trainable experts endows the module with the ability to not disturb the distribution of the model's original parameters. Meanwhile, we propose an unsupervised imitation learning method designed for RouGE. We use unlabeled clean images to synthesize degraded images and let the model learn to narrow the gap between them. Through imitation learning, RouGE can be trained using a small amount of unlabeled images (approximately 2k images for each type of degradation), making it more suitable for industrial applications.

The main contributions of this work are summarized as follows: **(i)** We propose RouGE, a PEFT module designed for making pretrained less-robust SAM-based model a robust all-in-one model with marginal cost. The design of RouGE ensures the capability to maintain the model's original output features unchanged and only conduct selective feature modifications, thus avoiding the catastrophic forgetting issue associated with fine-tuning. **(ii)** We propose an unsupervised imitation learning approach, utilizing unlabeled images and synthesized degraded images for training, thereby circumventing the problem of missing labeled data and facilitating easier training of robust models for industrial applications. **(iii)** Our comprehensive experiments demonstrate that the RouGE method significantly enhances model robustness. Compared to the original model, RouGE can improve segmentation accuracy for degraded inputs by 4-13% in mAP with hardly any negative effects on results for non-degraded inputs. RouGE also outperforms other fine-tuning methods by a significant margin, even with a low trainable parameter ratio (about 1.56%).

2 RELATED WORKS

2.1 SAM AND ITS VARIANTS

Since the introduction of SAM, there has been a continuous emergence of derivative works (Zhang et al., 2023b; Zhang & Jiao, 2023). Researchers in the field of medical image segmentation are focused on fine-tuning SAM for high-quality medical image segmentation tasks (Zhang et al., 2023d; Mohapatra et al., 2023; de Oliveira et al., 2023; Li et al., 2024; Wu et al., 2023; Hu et al., 2023; Gao et al., 2023). In addition, there is a wealth of work fine-tuning SAM to adapt to other types of segmentation tasks such as satellite image segmentation (Ren et al., 2024), shadow Detection (Jie & Zhang, 2023; Chen et al., 2023c), marine animal segmentation (Zhang et al., 2024), and so on (He et al., 2024; Williams et al., 2023; Cao et al., 2023). Considering the vast parameter count of SAM, in real-world applications, compressed SAM models capable of real-time segmentation hold a higher value. Through knowledge distillation and model pruning, (Zhang et al., 2023a; Zhao et al., 2023; Xiong et al., 2023) have successfully compressed SAM models to a fraction of their original size. The EfficientSAM (Xiong et al., 2023) uses the MEA (He et al., 2022) method to distillate the pretrained SAM model and retains the performance of the SAM model most comprehensively. Recently, RobustSAM (Chen et al., 2024) has similarly noted the sensitivity of SAM to real-world degradation and used a post-processing module to handle this problem. However, their model still has a high learnable parameter count, which demands significant computational resources for both inference and training. By employing PEFT methods, we can effectively leverage the performance of the base model to achieve efficient model adjustments, thereby reducing computational costs.

2.2 PARAMETER EFFICIENT FINE-TUNING

The PEFT method has been widely applied to SAM-based models (Sahay & Savakis, 2024), primarily fall into prompt tuning (Wang et al., 2024b; Jia et al., 2022), adapter-like tuning (Houlsby et al., 2019; Jie & Deng, 2022; Chen et al., 2022; Wang et al., 2020; Pfeiffer et al., 2020a; Wang et al., 2022; Yu et al., 2024), partial tuning (Basu et al., 2024; Zaken et al., 2021) and reparameterization fine-tuning (Jie & Deng, 2023; Lian et al., 2022; Hu et al., 2021) categories. However, the primary application scenario of PEFT methods is fine-tuning models for downstream tasks and the original model's feature distribution would undergo significant disruption. To mitigate this issue, knowledge injection (Zhang et al., 2023e;c; Wang et al., 2020) and MoE-based (Shazeer et al., 2017; Kim et al., 2020; Yu et al., 2021) PEFT methods have been proposed. However, the former (Wang et al., 2020) requires explicit task annotations, while the latter (Wang et al., 2022; Chen et al., 2022) yields sub-optimal results due to the lack of clear classification methods. These methods all fail to effectively enhance robustness.

3 METHODOLOGY

3.1 MOTIVATION

Segmentation models are sensitive to image degradation. When facing various types of degradation, SAM and SAM-based models may experience varying degrees of performance loss (Ji et al., 2024; Huang et al., 2023; Qiao et al., 2023; Wang et al., 2023). This is because degradation alters the overall or local distribution of image features, leading the model to erroneous perceptions. For instance, the texture of rainwater may cause black objects and shadows to be perceived as one entity, low-light environments may render object edges difficult to distinguish, and motion blur may lead to the cohesion of different objects. The SAM model exhibits relative robustness, whereas EfficientSAM performs comparatively worse. In Figure 2, we depict the segmentation performance degradation of SAM and the SAM-based EfficientSAM model when encountering three types of real-world degradation: low-light, motion

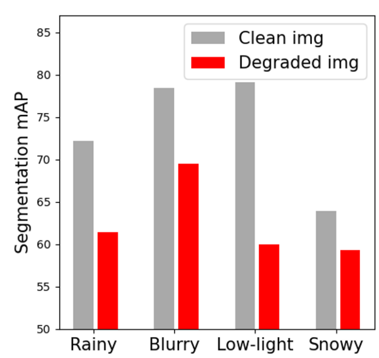

Figure 3: Impact of different type of real-world degradation on EfficientSAM

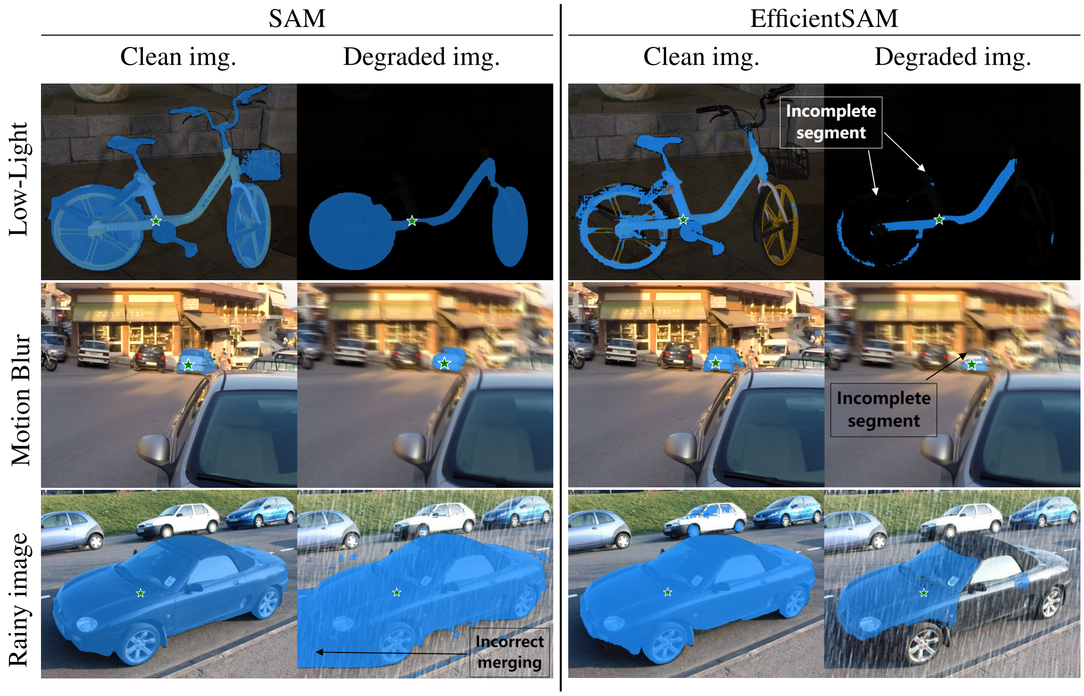

Figure 2: Presentation of the impact of various degradation on SAM and SAM-based Model

blur, and rainy conditions. How to mitigate the impact of real-world degradation on SAM-based models becomes an open challenge.

3.2 Robustness-Uplift Gated Experts

Fine-tuning pretrained models often leads to parameter drift, potentially resulting in models that only achieve domain adaptation rather than robustness improvement. Adapting the model to multiple domains simultaneously is the key point to achieving targeted robustness enhancement. Designs like that of Adapter-Hub (Poth et al., 2023; Pfeiffer et al., 2020b) can provide manual domain switching for models. From this, we conceive integrating multiple adapters into a single module, enabling the module to learn automatic switching, thereby achieving performance improvements across multiple domains. Therefore, we propose RouGE model and its unsupervised training process. RouGE utilizes lightweight probability gates within the module to control the weights of various experts, achieving diverse processing for different images by assigning corresponding expert proportions.

3.2.1 Overall Framework

The design of the RouGE model follows three key principles: (i) Automatically differentiate different inputs without type labels; (ii) The module should have the ability to "do nothing" to ensure that the original performance of the base model remains undisturbed; (iii) Utilize a minimal number of trainable parameters to ensure the module's parameter efficiency.

To achieve the above three objectives, we present a module with multiple independent probability gates, lazy expert (Expert 0), and trainable experts (Expert i), as shown in Figure 4. During both training and inference stages, RouGE takes image features $\mathcal{F}_t$ and input features x_t as inputs. $\mathcal{F}_t$ are fed into the probability gates to obtain the proportions of each expert. x_t are passed into each expert, and the predicted results generated by experts are multiplied by their respective proportions before being summed up and outputted. During the training process, synthetic image pairs of degraded and clean images are used, aiming to enhance model robustness by aligning the model predictions of degraded images with those of clean images. During the inference phase, arbitrary types of images can be used without the need to distinguish whether they are degraded images or not. Next, we will proceed to introduce each module separately.

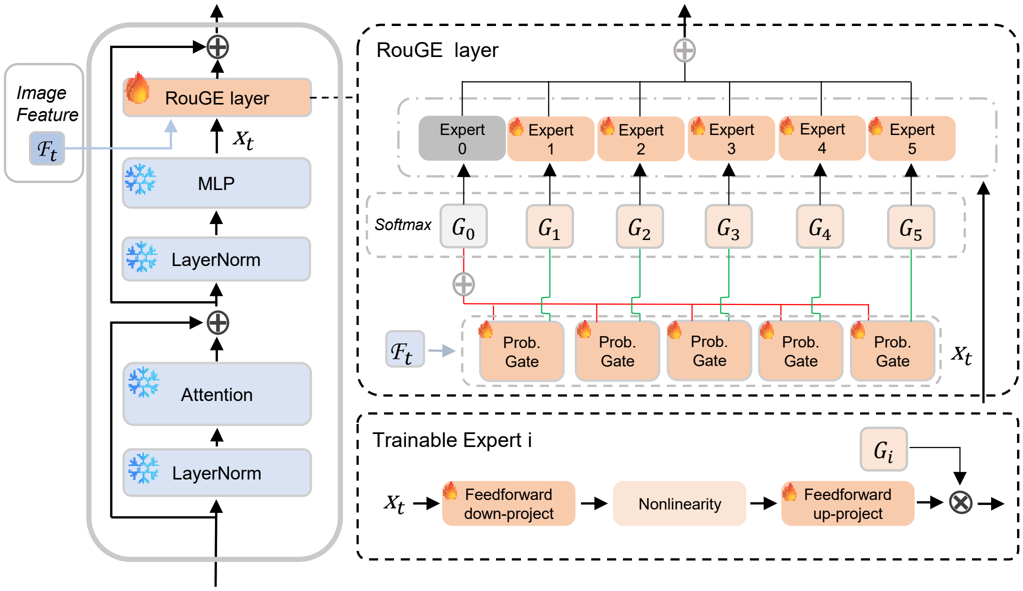

Figure 4: Structure of RouGE. RouGE module is inserted after the MLP layer of the transformer block, taking the MLP output features as input. The output result of the module is merged with the previous layer's features to obtain the output of the transformer block. This figure shows the structure of RouGE with 6 experts. Expert 0 is the lazy expert which directly multiplies the input data by G_0 and outputs the result. The other experts are trainable experts, composed of a dimension-reducing linear layer, a non-linear layer, and a dimension-restoring linear layer.

3.2.2 Probability Gates

The role of the probability gates is to distinguish between different types of input data and select the appropriate experts for processing. To reduce the noise in gate decisions, probability gates only take the image features $\mathcal{F}_t$ as input. In practical applications, we use global $\mathcal{F}_t$ extracted from the input image and the same $\mathcal{F}_t$ are used throughout the inference process for a single image. The probabilities generated by the probability gates are processed to serve as the output proportion parameters for each expert. The gate employs a lightweight structure of dual-layer fully connected layers, outputting a floating-point number between zero and one representing the acceptance probability.

To maintain the stability of the output features, the outputs of gates are concatenated and undergo a softmax function. In the case of having n experts, the number of gates is $n-1$. The acceptance probability of gate i is G_i. The total rejection probability G_0 is obtained by summing up the rejection probabilities of all gates and dividing by the total number of gates. After undergoing softmax processing, the sum of all probabilities equals one, meaning the proportions of each expert sum up to one. So we define the output of the probability gates as probability vector $\mathbf{G}$.

$$\mathbf{G} = \text{Softmax}([\frac{1}{n-1}\sum_{i}^{n-1} 1 - G_i, G_1, G_2, ..., G_{n-1}]). \tag{1}$$

3.2.3 Bottleneck Structure Experts

The probability vector $\mathbf{G}$ generated by the probability gates controls the weights of the experts and makes experts specializing in the current type of degradation exert their maximum impact. Each expert possesses its processing expertise after training and experts are divided into a lazy expert and multiple trainable experts. Lazy expert directly forwards input features to avoid introducing any bias and trainable experts introduce trainable parameters to fix different types of degradation.

To limit the number of trainable parameters in the model, we adopt the adapter (Houlsby et al., 2019)-like bottleneck structure, which includes a down-projection layer with parameters $\boldsymbol{W}_D \in \mathbb{R}^{m \times n}$ and an up-projection layer with parameters $\boldsymbol{W}_U \in \mathbb{R}^{n \times m}$. m is the input dimension and $n$is the bottleneck middle dimension, with $n \ll m$. The experts take x_t as input and output $E_i(x_t)$ of the same size. Experts can be formulated as

$$E_i(x_t) = \begin{cases} \text{GeLU}(x_t \cdot \boldsymbol{W}_D^i) \cdot \boldsymbol{W}_U^i, & i \in [1, n-1] \\ x_t, \quad i = 0 \end{cases} \tag{2}$$

Experts utilize a very small number of trainable parameters to ensure the efficiency of the module's parameters. We also compared the effectiveness of using other low-rank expert structures, as described in section 4.4. After being weighted by $\mathbf{G}$, the output y_t of the module can be defined as

$$\mathbf{E} = [x_t, E_1(x_t), E_2(x_t), ..., E_{n-1}(x_t)], \tag{3}$$

$$y_t = \mathbf{G}\mathbf{E}^T. \tag{4}$$

Parameters of all experts and probability gates are updated during the entire training stage. We do not manually specify experts for each type of input data. Instead, we allow the model to finely decompose task types, and use combinations of multiple experts to achieve better processing results through fully end-to-end training. By analyzing $\mathbf{G}$, we show it in section 4.3.2 that the trained RouGE model can differentiate between different types of input data and process them in a targeted manner. Additionally, we experimentally validated the effectiveness of the lazy expert in section 4.5.

Through ablation studies, we conclude that RouGE does not need to be added to every transformer block. Instead, adding them only to the final few blocks of the model can achieve better results and more details can be found in appendix A.1.

3.3 Loss Function for Unsupervised Imitation Learning

To better train the lightweight probability gate and experts and mitigate the absence of labeled degraded image data, we employ a method based on imitation learning to minimize noise during training. In the field of image restoration, a significant amount of artificially synthesized degraded datasets (such as rainy or foggy images) are proposed for restoration models. These datasets include pairs of clean images and degraded images generated by adding specific types of degradation. Our training method precisely leverages these datasets to teach the model how to "ignore" these degradations. Since the content in the images is consistent, the segmentation of the same object should yield identical ground truth results.

During training, we utilize models that do not include RouGE ($\mathcal{M}_{ori}$) and use clean images I_{clean} as input to obtain clean outputs $\mathcal{M}_{ori}(I_{clean})$ as targets T.

$$T = Mask(\mathcal{M}_{ori}(I_{clean})). \tag{5}$$

Next, we feed both clean images I_{clean} and degraded images I_{deg} into models containing RouGE ($\mathcal{M}_{RouGE}$), respectively. We then compute losses by comparing the results with T separately and perform backpropagation. The training objective is to ensure that models containing RouGE produce consistently high-quality results when faced with a set of clean images and degraded images with the same content. In the segment anything task, we employed the combination of Dice Loss and Sigmoid Focal Loss as the loss function $Loss_{dice\&focal}$. The training loss can be described as

$$L_{clean} = Loss_{dice\&focal}(T, \mathcal{M}_{RouGE}(I_{clean})), \tag{6}$$

$$L_{deg} = Loss_{dice\&focal}(T, \mathcal{M}_{RouGE}(I_{deg})). \tag{7}$$

4 Experiments

We evaluated the effectiveness of RouGE on image segmentation tasks. First, we introduce the experimental settings in section 4.1, covering the use of datasets, backbone selection, and the settings of other baseline methods. In section 4.2, we compare RouGE with other baseline models and provide a comprehensive analysis of the results. Next, in section 4.3, we empirically validate the automatic classification capability and out-of-domain performance of RouGE, and also compare the performance of restore-then-segment with RouGE. Finally, in section 4.4, we conduct other ablation experiments to explain its superiority.

4.1 Experimental Settings

Datasets. We obtained the Rain200L (Yang et al., 2017), DDN (Fu et al., 2017), GoPro (Nah et al., 2017), LIS (Chen et al., 2023a), and Snow100k (Liu et al., 2018) datasets from the image restoration domain and split them to serve as training and testing data for the model. Additionally, we used CityRain and CityFoggy (Cordts et al., 2015; 2016), as additional test data for experimentation. Among these, Rain200L and DDN consist of rainy weathered image pairs created using different methods, GoPro comprises dynamic blurry image pairs, LIS includes low-light image pairs, and Snow100k consists of snowy weathered image pairs.

All these datasets include artificially synthesized degraded images of varying degrees as well as their original clean images. Since the segment anything task requires a point prompt or bounding box as input, we selected the clean images from all image pairs and obtained bounding boxes in the images using a state-of-the-art object detection model. Subsequently, we inputted the bounding boxes and clean images into the Segment Anything model (Kirillov et al., 2023) to obtain the ground truth masks for the quantitative test. To avoid selecting points that are off-center from the object, we performed an erosion operation on the ground truth mask and then randomly selected a point as the point prompt. A set of image pairs uses the same point prompt and ground truth mask because the non-noise information on the image pairs is identical. We utilized CLIP's image encoder (Radford et al., 2021) as the image feature extractor in the experiment and pre-extracted image features for each in. The ablation study on image feature selection is in Appendix 10.

Pretrained backbone. We adopt the EfficientSAM-Ti (Xiong et al., 2023) as the backbone model and we utilized the pretrained parameters provided by the authors of EfficientSAM. The model comprises a transformer-based image encoder and a mask decoder. It takes the input image and point prompt and outputs the mask of the object pointed to by that point on the image.

baseline models. We selected 8 baseline models from 4 categories of methods for comparative experiments and the replicated RobustSAM on the EfficientSAM-Ti. Categorized by type, we selected (i) full fine-tuning. (ii) adapter-based: Adapter (Houlsby et al., 2019), Convpass (Jie & Deng, 2022), Adaptformer (Chen et al., 2022). (iii) partial fine-tuning: LN (Basu et al., 2024), Bitfit (Zaken et al., 2021). (iv) mixture-of-adapter: Adamix (Wang et al., 2022), AdapterFusion (Pfeiffer et al., 2020a) (v) RobustSAM (Chen et al., 2024): employing AMFG-F, AOTG, and ROT on base model, as our baseline methods. Additionally, while the RobustSAM paper employed supervised learning, we opted for unsupervised learning to ensure fairness.

Hyperparameter settings. In all experiments, we use AdamW as the optimizer with $lr = 1e-4, weightdecay = 5e-2$. We use the combination of dice loss and sigmoid focal loss as the loss function and each accounts for 50%. In the absence of specific instructions, we set the number of experts in the RouGE model to 6. Additionally, the RouGE model is only added to the last two layers of the transformer block in the image encoder. During training, we utilized a NVIDIA GeForce RTX 3090 GPU and the five datasets were alternated sequentially to train a robust model capable of handling five types of degradation simultaneously.

4.2 Comparison with SOTA methods

4.2.1 Quantitative experiment

We performed unsupervised training on the Rain200L, DDN, GoPro, LIS, and Snow100k datasets simultaneously and compared different fine-tuning methods, as shown in Table 1. As can be observed, our proposed RouGE method demonstrates strong robustness improvement capabilities. RouGE can significantly enhance the model's mean average precision on both degraded and non-degraded data. Compared to other methods, the RouGE model exhibits the greatest improvement in mAP and reduces the performance degradation between degraded and non-degraded data. From the table, it can be observed that other methods fail to balance the segmentation performance between degraded and non-degraded inputs, thus validating our argument. Moreover, the full fine-tuning method performs poorly when the available dataset size is small. The trainable parameters of the RouGE model account for only 1.56% of the total fine-tuning parameters. In summary, the RouGE model surpasses other methods, achieving state-of-the-art performance in enhancing robustness.

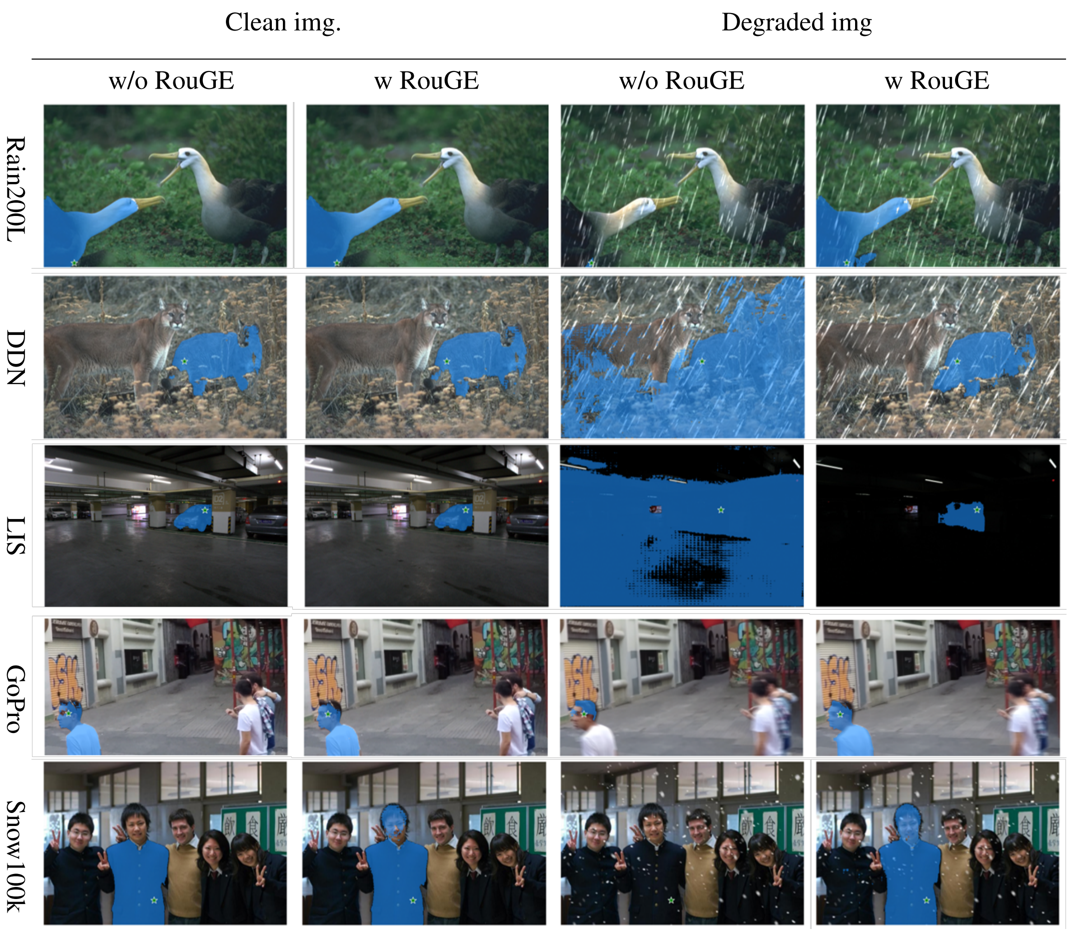

Figure 5: Presentation of the Segmentation performance

Table 1: **Comparison with other methods on Robustness-uplift benchmark.** We report the mean average precision (mAP) of various methods on the test set. **Bold** number indicates the best value for that data type.

		Rain200L		DDN		GoPro		LIS		Snow100k	
Method	Params	Clean	Rainy	Clean	Rainy	Clean	Blurry	Clean	Low-light	Clean	Snowy
Base model	-	74.13	72.82	72.23	61.40	78.48	69.48	**79.12**	60.02	63.92	59.37
Full fine-tune	100%	69.57	67.85	65.09	62.52	61.05	59.55	64.60	59.45	58.85	59.98
LN	0.18%	75.44	74.48	72.9	69.05	78.62	71.73	78.71	65.47	64.02	62.51
Bitfit	0.51%	75.95	74.15	73.43	65.99	73.84	69.03	78.29	63.17	63.17	61.74
Adapter	1.47%	74.19	74.96	75.94	69.78	76.93	71.51	77.72	64.09	63.75	62.08
Convpass	2.00%	74.50	73.52	72.06	60.11	73.85	70.34	69.21	62.01	63.80	61.43
Adaptformer	8.61%	74.58	75.01	75.58	70.76	76.52	71.66	77.19	64.93	63.98	62.55
Adamix	3.67%	72.38	71.02	75.73	63.96	79.85	75.22	75.96	63.85	63.72	60.91
AdapterFusion	7.47%	73.92	72.67	73.71	71.61	77.72	70.61	73.19	64.45	63.62	60.09
RobustSAM	32.11%	75.66	75.82	77.47	73.33	79.75	**75.74**	77.06	65.20	65.00	**63.19**
RouGE (Ours)	1.56%	**76.61**	**76.99**	**77.75**	**74.85**	**80.01**	75.17	78.77	**65.62**	**65.34**	63.14

4.2.2 QUALITATIVE EXPERIMENT

In Figure 5, we show the robustness enhancement ability of RouGE. For clean image inputs, the insertion of RouGE only brings minimal changes in segmentation outcomes. Conversely, for inputs with various types of degradation, the insertion of RouGE significantly improved model segmentation results. This aligns with our previously conducted quantitative experiments. More segmentation results can be found in appendix A.4

Table 2: Restore-then-Segment experiments. We report the mean average precision (mAP). **Bold** indicates better result.

Method	Rain200L-Rainy	DDN-Rainy	GoPro-Blurry	LIS-Low-light	Snow100k-Snowy
Base model	72.82	61.40	69.48	60.02	59.37
AirNet+Base model	73.56	66.31	69.94	53.53	60.18
RouGE+Base model	**76.99**	**74.85**	**75.17**	**65.62**	**63.14**

4.3 Discussion

4.3.1 Restore then Segment?

The goal of image restoration tasks is to restore images to a form that is more friendly to the human eye instead of downstream visual tasks. We used AirNet (Li et al., 2022) as the image restoration model and conducted segmentation experiments on degraded images after restoration. From Table 2, we can see that the improvement of image restoration on downstream segmentation tasks is limited.

4.3.2 Task discrimination capability of RouGE

RouGE model utilizes probability gates to distinguish input data, thereby achieving automatic classification and processing of unlabeled data. The outputs of these gates demonstrate strong interpretability and can be utilized for feature analysis. By using the output of the probability gates in the RouGE model as features for t-SNE clustering, we obtained a visualization that demonstrates how the RouGE model classifies and processes the input. From the clustering results, it is evident that there is a clear distinction between degraded and non-degraded data. Moreover, due to varying degrees of degradation, some degraded data may bear similarities to non-degraded data, while others exhibit significant differences.

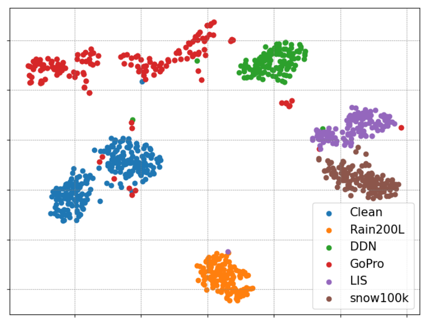

Figure 6: t-SNE visualization of probability vector G.

4.3.3 Out-of-domain experiments

To further demonstrate that the RouGE model does not disrupt the original model's parameter distribution, we conducted out-of-domain data experiments. We conducted tests using the CityRain and CityFoggy datasets, which contain road images under normal weather conditions as well as rainy and foggy weather conditions. We compared the segmentation performance of the RouGE model with that of the base model.

Table 3: Out-of-domain experiments. We report the mean average precision (mAP). **Bold** indicates better result.

	Cityrain		Cityfoggy	
Method	Clean	Rainy	Clean	Foggy
Base model	69.49	61.23	71.21	56.14
RouGE	**70.27**	**66.60**	**75.38**	**71.29**

From Table 3, the RouGE model exhibits equally outstanding performance in out-of-domain scenarios, preserving the model's original performance intact. Moreover, it demonstrates performance improvements for similar degradation types like rainy images(Cityrain) and exhibits commendable zero-shot performance for unseen degradation types like foggy images(Cityfoggy).

4.4 Ablation Studies

4.4.1 Expert Design

In addition to Adapter-like experts, we also explored the effectiveness of other types of experts. In previous experiments, we found that the LN method, which fine-tunes the affine transformation parameters of the LN layers, could achieve relatively good results. Therefore, we considered testing the use of affine-based experts. The affine expert we designed contains a set of affine transform

Table 4: Test the performance of Adapter-like expert and Affine-based expert.

	Rain200L		DDN		GoPro		LIS		Snow100k	
Method	Clean	Rainy	Clean	Rainy	Clean	Blurry	Clean	Low-light	Clean	Snowy
Affine expert	74.86	75.57	75.28	70.68	**80.26**	73.75	**78.89**	65.57	**65.49**	**63.26**
Adapter expert	**76.61**	**76.99**	**77.75**	**74.85**	80.01	**75.17**	78.77	**65.62**	65.34	63.14

Table 5: Ablation study on lazy expert

	Rain200L		DDN		GoPro		LIS		Snow100k	
Method	Clean	Rainy	Clean	Rainy	Clean	Blurry	Clean	Low-light	Clean	Snowy
w/o lazy expert	75.76	**77.27**	76.04	71.96	79.06	**75.24**	77.64	64.46	63.98	**63.43**
w lazy expert	**76.61**	76.99	**77.75**	**74.85**	**80.01**	75.17	**78.77**	**65.62**	**65.34**	63.14

Table 6: Comparison between supervised and unsupervised RouGE model

	Rain200L		DDN		GoPro		LIS		Snow100k	
Method	Clean	Rainy	Clean	Rainy	Clean	Blurry	Clean	Low-light	Clean	Snowy
Base model	74.13	72.82	72.23	61.40	78.48	69.48	79.12	60.02	63.92	59.37
Unsupervised	75.44	74.48	72.9	69.05	78.62	71.73	78.91	65.47	64.02	62.51
Supervised	**77.08**	**84.10**	**77.60**	**78.62**	**84.39**	**84.41**	**80.16**	**76.21**	**67.39**	**67.56**

parameters, γ and β, similar to the trainable parameters of the layer normalization layer. The comparative experimental results are presented in Table 4. Based on the results, the performance of the Adapter-like expert is better than that of the Affine-based expert.

4.5 LAZY EXPERT

The presence of lazy experts facilitates the model's ability to handle clean images and reduces training complexity. In the experiments, we maintained the same number of trainable experts. The results in Table 5 reflect the positive effects brought by the lazy expert.

4.5.1 SUPERVISED VS. UNSUPERVISED LEARNING

The unsupervised learning training of the RouGE model greatly reduces the cost of data acquisition. However, at the same time, unsupervised learning also makes RouGE relatively ineffective when faced with severely degraded types of data such as the LIS dataset. Using more accurate data labels for supervised learning can further enhance the model's capabilities. Based on unsupervised imitation learning and switching to using ground truth labels to train degraded images instead, We conducted a comparative experiment with supervised learning using labels generated by SAM, as shown in Table 6.

As observed, the model's detection accuracy on degraded images has significantly improved. Therefore, under circumstances where obtaining data labels is feasible, you can weigh the cost and benefit of obtaining labels to choose a more suitable training method.

5 CONCLUSION

In this paper, we propose a plugin robustness enhancement module, RouGE, which can enhance the robustness of pretrained SAM-based models at marginal cost. Experiments conducted both within and outside the domain demonstrate RouGE's capability to selectively modify degraded images while preserving the original performance of the model for clean images. Compared with existing PEFT methods and reproduction of RobustSAM, RouGE demonstrates superiority in both robustness enhancement capability and efficiency in terms of trainable parameters. RouGE model exhibits high versatility, as it can be seamlessly integrated into any transformer block. This renders it with the potential to be applied across various types of visual models. In the future, we will continue to explore the application of RouGE in a broader range of visual models and tasks.

REFERENCES

Samyadeep Basu, Shell Hu, Daniela Massiceti, and Soheil Feizi. Strong baselines for parameter-efficient few-shot fine-tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 11024–11031, 2024.

Yunkang Cao, Xiaohao Xu, Chen Sun, Yuqi Cheng, Zongwei Du, Liang Gao, and Weiming Shen. Segment any anomaly without training via hybrid prompt regularization. *arXiv preprint arXiv:2305.10724*, 2023.

Linwei Chen, Ying Fu, Kaixuan Wei, Dezhi Zheng, and Felix Heide. Instance segmentation in the dark. *International Journal of Computer Vision*, 131(8):2198–2218, 2023a.

Linwei Chen, Ying Fu, Kaixuan Wei, Dezhi Zheng, and Felix Heide. Instance segmentation in the dark. *International Journal of Computer Vision*, 131(8):2198–2218, 2023b.

Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems*, 35:16664–16678, 2022.

Tianrun Chen, Lanyun Zhu, Chaotao Ding, Runlong Cao, Shangzhan Zhang, Yan Wang, Zejian Li, Lingyun Sun, Papa Mao, and Ying Zang. Sam fails to segment anything?–sam-adapter: Adapting sam in underperformed scenes: Camouflage, shadow, and more. *arXiv preprint arXiv:2304.09148*, 1(2):5, 2023c.

Wei-Ting Chen, Yu-Jiet Vong, Sy-Yen Kuo, Sizhou Ma, and Jian Wang. Robustsam: Segment anything robustly on degraded images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4081–4091, 2024.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Scharwächter, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset. In *CVPR Workshop on the Future of Datasets in Vision*, volume 2, pp. 1, 2015.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.

Ziteng Cui, Guo-Jun Qi, Lin Gu, Shaodi You, Zenghui Zhang, and Tatsuya Harada. Multitask aet with orthogonal tangent regularity for dark object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2553–2562, 2021.

Christian Mattjie de Oliveira, Luis Vinícius de Moura, Rafaela Cappelari Ravazio, Lucas Silveira Kupssinskü, Otávio Parraga, Marcelo Mussi Delucis, and Rodrigo Coelho Barros. Zero-shot performance of the segment anything model (sam) in 2d medical imaging: A comprehensive evaluation and practical guidelines. *CoRR*, 2023.

Xueyang Fu, Jiabin Huang, Delu Zeng, Yue Huang, Xinghao Ding, and John Paisley. Removing rain from single images via a deep detail network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3855–3863, 2017.

Yifan Gao, Wei Xia, Dingdu Hu, and Xin Gao. Desam: Decoupling segment anything model for generalizable medical image segmentation. *arXiv preprint arXiv:2306.00499*, 2023.

Chunming He, Kai Li, Yachao Zhang, Guoxia Xu, Longxiang Tang, Yulun Zhang, Zhenhua Guo, and Xiu Li. Weakly-supervised concealed object segmentation with sam-based pseudo labeling and multi-scale feature grouping. *Advances in Neural Information Processing Systems*, 36, 2024.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pp. 2790–2799. PMLR, 2019.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Mingzhe Hu, Yuheng Li, and Xiaofeng Yang. Skinsam: Empowering skin cancer segmentation with segment anything model. *arXiv preprint arXiv:2304.13973*, 2023.

Yihao Huang, Yue Cao, Tianlin Li, Felix Juefei-Xu, Di Lin, Ivor W Tsang, Yang Liu, and Qing Guo. On the robustness of segment anything. *arXiv preprint arXiv:2305.16220*, 2023.

Wei Ji, Jingjing Li, Qi Bi, Tingwei Liu, Wenbo Li, and Li Cheng. Segment anything is not always perfect: An investigation of sam on different real-world applications, 2024.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European Conference on Computer Vision*, pp. 709–727. Springer, 2022.

Leiping Jie and Hui Zhang. When sam meets shadow detection. *arXiv preprint arXiv:2305.11513*, 2023.

Shibo Jie and Zhi-Hong Deng. Convolutional bypasses are better vision transformer adapters. *arXiv preprint arXiv:2207.07039*, 2022.

Shibo Jie and Zhi-Hong Deng. Fact: Factor-tuning for lightweight adaptation on vision transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 1060–1068, 2023.

Sijin Kim, Namhyuk Ahn, and Kyung-Ah Sohn. Restoring spatially-heterogeneous distortions using mixture of experts network. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4015–4026, 2023.

Boyun Li, Xiao Liu, Peng Hu, Zhongqin Wu, Jiancheng Lv, and Xi Peng. All-in-one image restoration for unknown corruption. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17452–17462, 2022.

Yuheng Li, Mingzhe Hu, and Xiaofeng Yang. Polyp-sam: Transfer sam for polyp segmentation. In *Medical Imaging 2024: Computer-Aided Diagnosis*, volume 12927, pp. 759–765. SPIE, 2024.

Dongze Lian, Daquan Zhou, Jiashi Feng, and Xinchao Wang. Scaling & shifting your features: A new baseline for efficient model tuning. *Advances in Neural Information Processing Systems*, 35: 109–123, 2022.

Yun-Fu Liu, Da-Wei Jaw, Shih-Chia Huang, and Jenq-Neng Hwang. Desnownet: Context-aware deep network for snow removal. *IEEE Transactions on Image Processing*, 27(6):3064–3073, 2018.

Sovesh Mohapatra, Advait Gosai, and Gottfried Schlaug. Brain extraction comparing segment anything model (sam) and fsl brain extraction tool. *arXiv preprint arXiv:2304.04738*, 2023.

Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3883–3891, 2017.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. Adapterfusion: Non-destructive task composition for transfer learning. *arXiv preprint arXiv:2005.00247*, 2020a.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. Adapterhub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 46–54, 2020b.

Clifton Poth, Hannah Sterz, Indraneil Paul, Sukannya Purkayastha, Leon Engländer, Timo Imhof, Ivan Vulić, Sebastian Ruder, Iryna Gurevych, and Jonas Pfeiffer. Adapters: A unified library for parameter-efficient and modular transfer learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 149–160, Singapore, December 2023. Association for Computational Linguistics. URL `https://aclanthology.org/2023.emnlp-demo.13`.

Vaishnav Potlapalli, Syed Waqas Zamir, Salman Khan, and Fahad Shahbaz Khan. Promptir: Prompting for all-in-one blind image restoration. *arXiv preprint arXiv:2306.13090*, 2023.

Yu Qiao, Chaoning Zhang, Taegoo Kang, Donghun Kim, Shehbaz Tariq, Chenshuang Zhang, and Choong Seon Hong. Robustness of sam: Segment anything under corruptions and beyond. *arXiv preprint arXiv:2306.07713*, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

Simiao Ren, Francesco Luzi, Saad Lahrichi, Kaleb Kassaw, Leslie M Collins, Kyle Bradbury, and Jordan M Malof. Segment anything, from space? In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 8355–8365, 2024.

Rajat Sahay and Andreas Savakis. Mopeft: A mixture-of-pefts for the segment anything model. *arXiv preprint arXiv:2405.00293*, 2024.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

Cong Wang, Jinshan Pan, Wei Wang, Jiangxin Dong, Mengzhu Wang, Yakun Ju, and Junyang Chen. Promptrestorer: A prompting image restoration method with degradation perception. *Advances in Neural Information Processing Systems*, 36, 2024a.

Haixin Wang, Jianlong Chang, Yihang Zhai, Xiao Luo, Jinan Sun, Zhouchen Lin, and Qi Tian. Lion: Implicit vision prompt tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 5372–5380, 2024b.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Guihong Cao, Daxin Jiang, Ming Zhou, et al. K-adapter: Infusing knowledge into pre-trained models with adapters. *arXiv preprint arXiv:2002.01808*, 2020.

Yaqing Wang, Sahaj Agarwal, Subhabrata Mukherjee, Xiaodong Liu, Jing Gao, Ahmed Hassan Awadallah, and Jianfeng Gao. Adamix: Mixture-of-adaptations for parameter-efficient model tuning. *arXiv preprint arXiv:2205.12410*, 2022.

Yuqing Wang, Yun Zhao, and Linda Petzold. An empirical study on the robustness of the segment anything model (sam). *arXiv preprint arXiv:2305.06422*, 2023.

Dominic Williams, Fraser MacFarlane, and Avril Britten. Leaf only sam: a segment anything pipeline for zero-shot automated leaf segmentation. *arXiv preprint arXiv:2305.09418*, 2023.

Junde Wu, Rao Fu, Huihui Fang, Yuanpei Liu, Zhaowei Wang, Yanwu Xu, Yueming Jin, and Tal Arbel. Medical sam adapter: Adapting segment anything model for medical image segmentation. *arXiv preprint arXiv:2304.12620*, 2023.

Yunyang Xiong, Bala Varadarajan, Lemeng Wu, Xiaoyu Xiang, Fanyi Xiao, Chenchen Zhu, Xiaoliang Dai, Dilin Wang, Fei Sun, Forrest Iandola, et al. Efficientsam: Leveraged masked image pretraining for efficient segment anything. *arXiv preprint arXiv:2312.00863*, 2023.

Wenhan Yang, Robby T Tan, Jiashi Feng, Jiaying Liu, Zongming Guo, and Shuicheng Yan. Deep joint rain detection and removal from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1357–1366, 2017.

Jiazuo Yu, Yunzhi Zhuge, Lu Zhang, Dong Wang, Huchuan Lu, and You He. Boosting continual learning of vision-language models via mixture-of-experts adapters. *arXiv preprint arXiv:2403.11549*, 2024.

Ke Yu, Xintao Wang, Chao Dong, Xiaoou Tang, and Chen Change Loy. Path-restore: Learning network path selection for image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):7078–7092, 2021.

Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *arXiv preprint arXiv:2106.10199*, 2021.

C Zhang, D Han, Y Qiao, JU Kim, SH Bae, S Lee, and CS Hong. Faster segment anything: Towards lightweight sam for mobile applications. *arXiv preprint arXiv:2306.14289*, 2023a.

Chaoning Zhang, Fachrina Dewi Puspitasari, Sheng Zheng, Chenghao Li, Yu Qiao, Taegoo Kang, Xinru Shan, Chenshuang Zhang, Caiyan Qin, Francois Rameau, et al. A survey on segment anything model (sam): Vision foundation model meets prompt engineering. *arXiv preprint arXiv:2306.06211*, 2023b.

Pingping Zhang, Tianyu Yan, Yang Liu, and Huchuan Lu. Fantastic animals and where to find them: Segment any marine animal with dual sam. *arXiv preprint arXiv:2404.04996*, 2024.

Qinggang Zhang, Junnan Dong, Hao Chen, Xiao Huang, Daochen Zha, and Zailiang Yu. Knowgpt: Black-box knowledge injection for large language models. *arXiv preprint arXiv:2312.06185*, 2023c.

Shan Zhang, Bohui Liang, Xuejun Zhang, Bin Li, and Liying Zhang. An investigation of segment anything model (sam) on uterus segmentation. In *Proceedings of the 2023 8th International Conference on Biomedical Signal and Image Processing*, pp. 39–41, 2023d.

Yichi Zhang and Rushi Jiao. Towards segment anything model (sam) for medical image segmentation: a survey. *arXiv [Preprint]*, 2023.

Zhengyan Zhang, Zhiyuan Zeng, Yankai Lin, Huadong Wang, Deming Ye, Chaojun Xiao, Xu Han, Zhiyuan Liu, Peng Li, Maosong Sun, et al. Plug-and-play knowledge injection for pre-trained language models. *arXiv preprint arXiv:2305.17691*, 2023e.

Xu Zhao, Wenchao Ding, Yongqi An, Yinglong Du, Tao Yu, Min Li, Ming Tang, and Jinqiao Wang. Fast segment anything. *arXiv preprint arXiv:2306.12156*, 2023.

A Appendix

A.1 How many experts and RouGEs are needed?

The number of experts and the number of RouGE directly affect the trainable parameter count. Usually, fine-tuned models of like adapter are inserted into each transformer block. However, the RouGE model is more of a widening rather than a deepening model, so inserting it into all transformer blocks may not necessarily be advantageous. The base model's has 11 transformer layers, and we experimented with inserting the RouGE model starting from ith layer, as shown in 7. Training was conducted for 10 epochs in all cases. The experimental results indicate that the RouGE model is not suitable for being added to all blocks but rather for being added to the final few layers. In our previous experiments, we used the optimal configuration, adding RouGE to the last two layers.

For the number of experts, we conducted a comparative experiment as shown in 8.When the number of experts degrades to 2, the RouGE model becomes an adapter that can adjust fusion coefficients. As the number of experts increases, the effectiveness of RouGE improves. After balancing the parameter count and effectiveness, we chose $N = 6$ as the experimental hyperparameters setting.

Table 7: Inserting RouGE from ith layer.

	Rain200L		DDN		GoPro		LIS		Snow100k	
i	Clean	Rainy	Clean	Rainy	Clean	Blurry	Clean	Low-light	Clean	Snowy
$i = 0$	76.04	73.73	73.21	71.02	79.09	**75.31**	78.28	62.71	65.05	63.46
$i = 5$	75.65	76.46	76.50	72.55	79.40	73.95	78.34	69.07	**65.82**	**63.83**
$i = 9$	75.91	75.03	77.34	74.02	79.41	75.13	78.23	64.98	65.41	63.17
$i = 10$	**76.61**	**76.99**	**77.75**	**74.85**	**80.01**	75.17	**78.77**	**65.62**	65.34	63.14
$i = 11$	75.49	77.30	73.51	69.93	78.99	73.01	78.35	65.08	65.11	63.38

Table 8: RouGE with N experts

	Rain200L		DDN		GoPro		LIS		Snow100k	
N	Clean	Rainy	Clean	Rainy	Clean	Blurry	Clean	Low-light	Clean	Snowy
$N = 2$	74.99	74.97	73.12	68.01	78.13	71.71	78.52	61.39	63.75	60.11
$N = 4$	75.93	76.01	77.63	74.81	78.96	73.12	78.14	64.21	65.02	63.11
$N = 6$	76.61	**76.99**	**77.75**	**74.85**	**80.01**	75.17	**78.77**	**65.62**	**65.34**	**63.14**
$N = 8$	**76.73**	76.51	77.32	73.92	79.69	**75.41**	78.72	65.59	65.31	62.98

A.2 RouGE with LN

Considering the compatibility between the LN method and Rouge, we conducted additional experiments by fine-tuning the LayerNorm layer within the blocks while adding RouGE, as shown in 9. The experimental results indicate that adding a small number of parameters, the LN-RouGE model can bring about a slight improvement in accuracy, but it cannot surpass the RouGE model itself entirely. Moreover, adding LN trainable parameters does not reduce the accuracy difference between degraded and non-degraded data groups.

Table 9: Test the combination of LN and the RouGE method.

	Rain200L		DDN		GoPro		LIS		Snow100k	
Method	Clean	Rainy	Clean	Rainy	Clean	Blurry	Clean	Low-light	Clean	Snowy
LN	75.44	74.48	72.9	69.05	78.62	71.73	78.91	65.47	64.02	62.51
RouGE	76.61	76.99	**77.75**	**74.85**	80.01	75.17	78.77	65.62	65.34	63.14
LN-RouGE	**76.86**	**77.54**	77.23	71.55	**81.11**	**75.18**	**79.65**	**66.98**	**66.45**	**63.98**

A.3 Comparison of using global feature F_t and local feature x_t

In experiment, we utilize the global feature F_t as a signal to control the gating weights in the probability gate input. The role of Ft is to provide the probability gate with more distinguishable features, thereby reducing the interference of intermediate variables in the model. As a comparison, we conducted additional experiments to compare the effectiveness of using x_t and F_t.

Table 10: Performance of RouGE with x_t and F_t

	Rain200L		DDN		GoPro		LIS		Snow100k	
feature	Clean	Rainy	Clean	Rainy	Clean	Blurry	Clean	Low-light	Clean	Snowy
x_t	**76.57**	76.94	77.60	73.40	79.44	74.75	**79.72**	64.53	64.19	63.06
F_t	76.61	**76.99**	**77.75**	**74.85**	**80.01**	**75.17**	78.77	**65.62**	**65.34**	**63.14**

A.4 More segmentation result

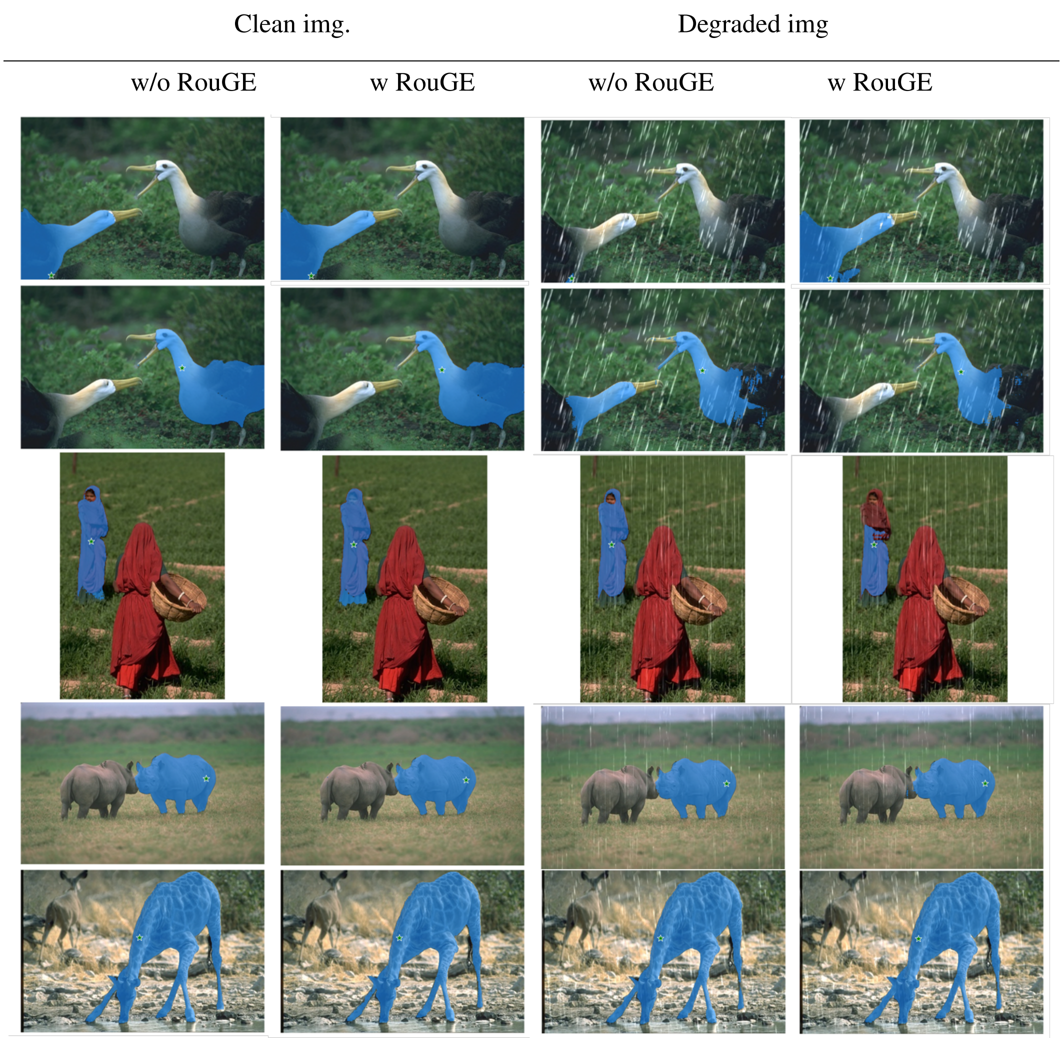

Figure 7: Presentation of the Segmentation performance on Rain200L dataset

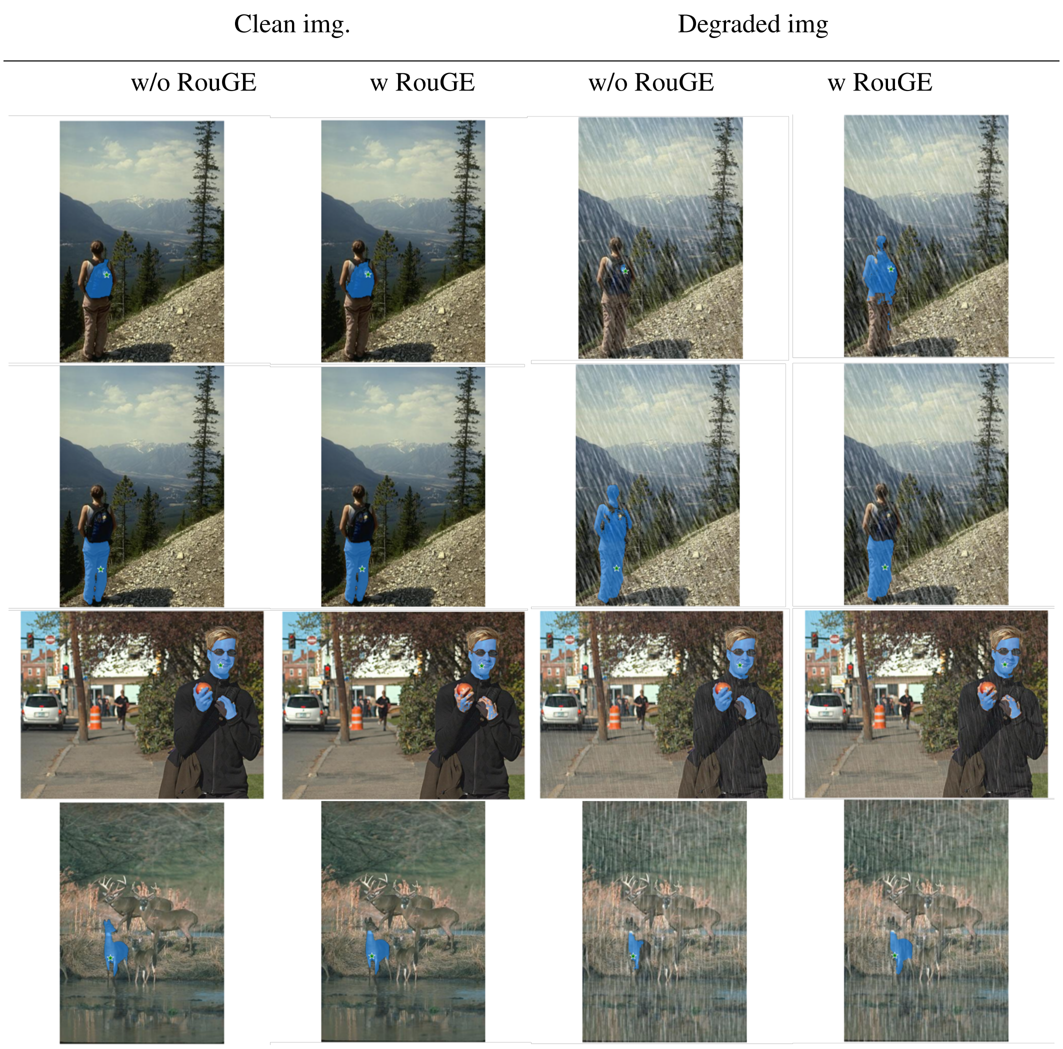

Figure 8: Presentation of the Segmentation performance on DDN dataset

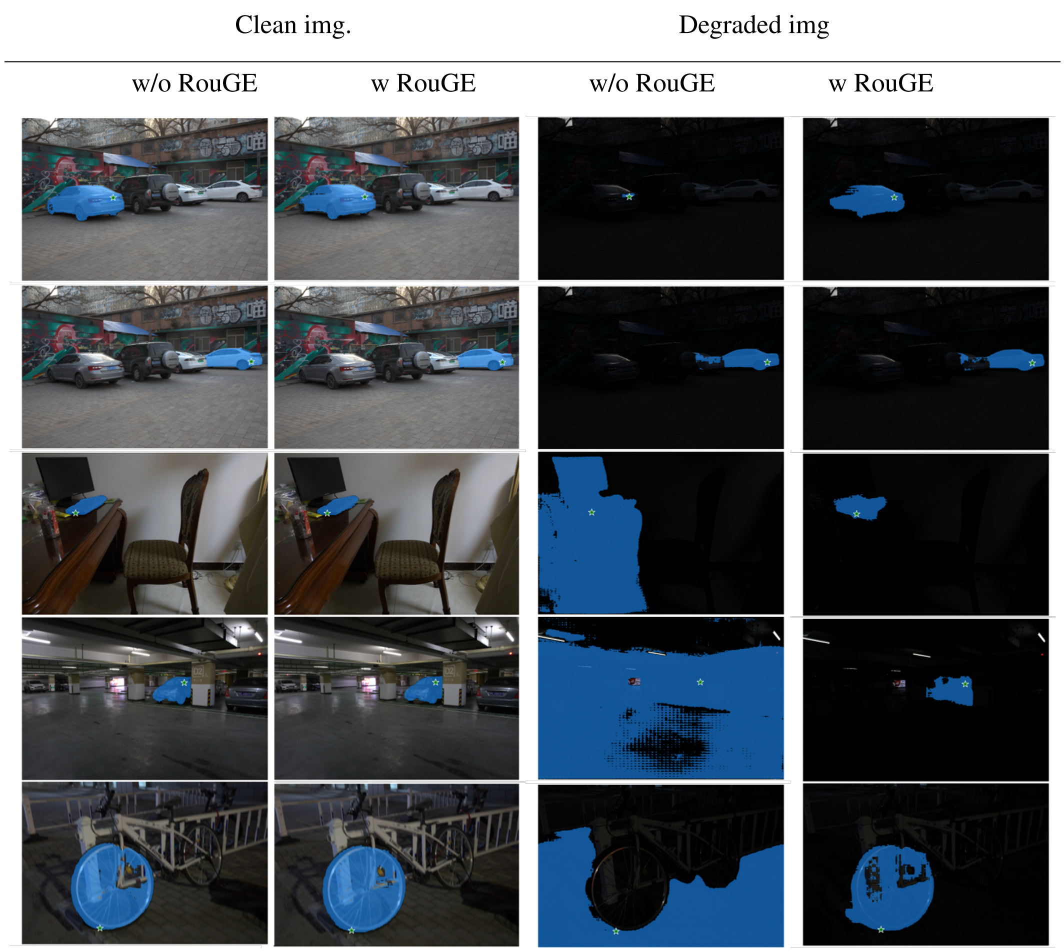

Figure 9: Presentation of the Segmentation performance on LIS dataset

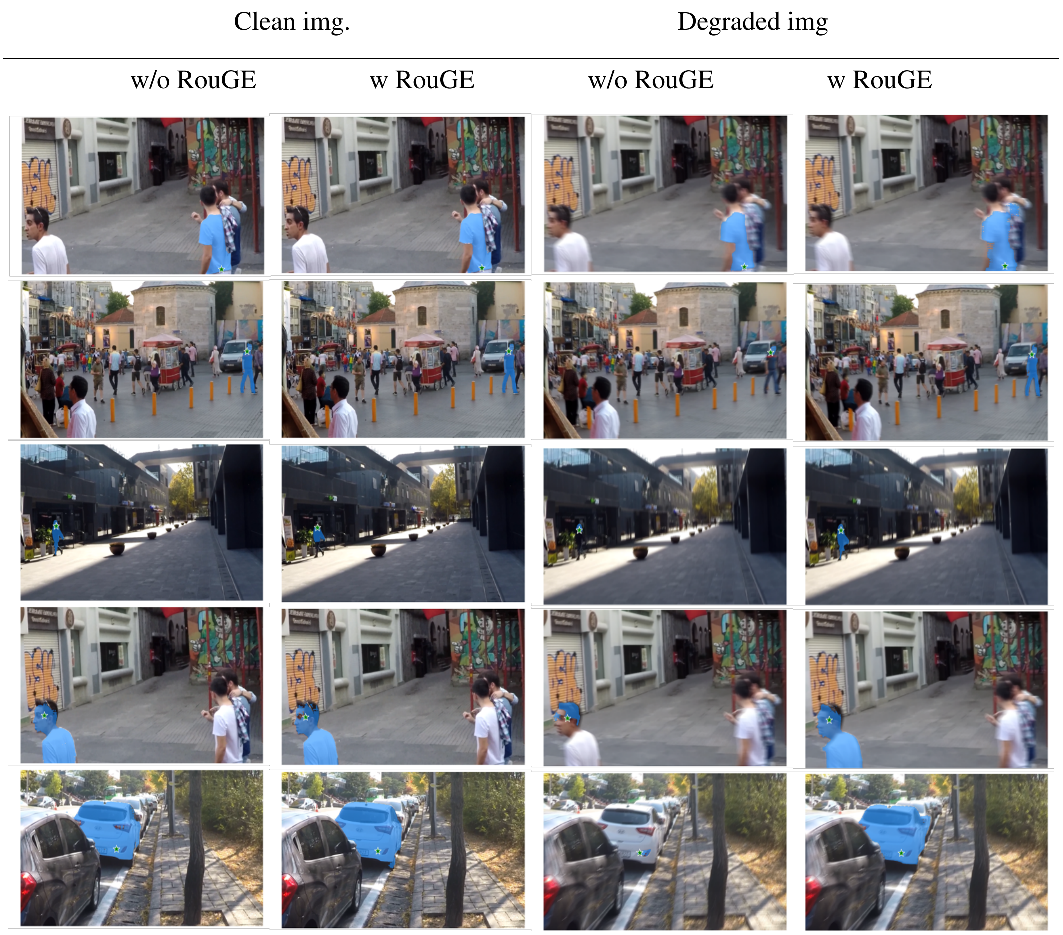

Figure 10: Presentation of the Segmentation performance on GoPro dataset

Clean img.		Degraded img	
w/o RouGE	w RouGE	w/o RouGE	w RouGE

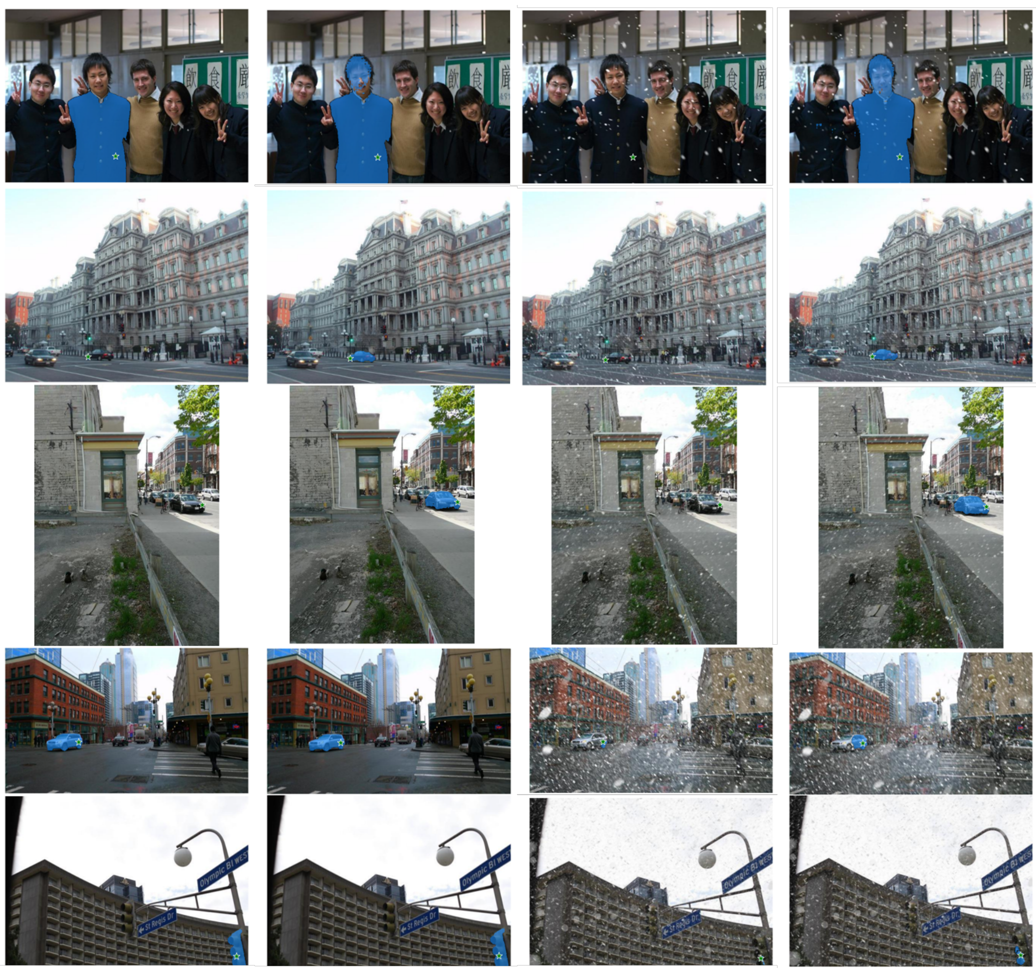

Figure 11: Presentation of the Segmentation performance on Snow100k dataset