Medical Segmentation Based on SAM and Implementation of Variant Problems

Zhi Han* 522030910215
Nan Jiang* 521030910386
Hanwen Liu* 522030910109
Jingyao Wu* 522030910199

Abstract

Segment Anything Model (SAM) is a useful foundation model to segment objects with prompts. Meanwhile, medical image segmentation is a critical component in clinical practice. However, using SAM for medical image segmentation with default prompt to automatically classify organs in the body do not perform as well as in other tasks. Therefore, we try various existing methods to adapt SAM for medical scenario by Parameter-Efficient Fine-Tuning (PEFT), compare the pros and cons of different methods, and select the best one to be incorporated to our own design . We implement a classifier on SAM and introduce a better loss function, which are specialized for classification tasks. Our trained model achieves relatively good results on FLARE22, which is closed to the state-of-the-art methods. We also conduct extensive experiments on downstream scenario to validate the effectiveness of our design.

1. Introduction

Segmentation is a fundamental task in medical image analysis, which involves identifying and delineating regions of interest (ROI) in various medical images, such asorgans, lesions, and tissues. Accurate segmentation is essential for many clinical applications, including disease diagnosis, treatment planning, and monitoring of disease progression. Medical image segmentation aims at indicating the anatomical or pathological structure of the corresponding tissues as required to facilitate computer-aided diagnosis and intelligent clinical surgery.

Fast and Low-resource semi-supervised Abdominal organ segmentation in CT (FLARE 2022) is a challenge fit into real medical situations, with a dataset containing many labeled and unlabeled cases of 13 organs. Our experiments are carried out mainly based on FLARE 2022 and other medical datasets.

SAM[7] is a promptable segmentation method that requires points or bounding boxes to specify the segmentation targets, and is better for generalization than other existing deep learning-based interactive segmentation methods. However, applying SAM for medical image segmentation directly does not get satisfying result. So how to finetune SAM is a crucial consideration. Since given SAM's substantial size, full fine-tuning directly on the model requires significant computational resources, we need a more parameter efficient finetuing method. LoRA and adapter are both useful existing PEFT methods, and we conduct experiments based on them. In the LoRA part, we apply LoRA layers to the Q and V projection layers of each of the transformer block in image encoder and mask decoder, and perform full fine-tuning on the segmentation head. In the Adapter part, we freeze all parameters and integrate adapters into transformer block in image encoder and mask decoder. To enable SAM to perform deterministic prediction, we implement a classifer on SAM by modifying SAM's segmentation head in mask decoder as base of various PEFT model. Our model achieved 0.5667 Dice Similarity Coefficient(DSC) on average of segmenting different organs, which has a 10 DSC advantage over the base SAM. In addition, we apply our model to various downstream tasks like GID5 and SIRST and also achieve good results, obviously overwhelming the base SAM and SAM2.

2. Related Works

2.1. Interactive Segmentation Model

SAM[7] is a state-of-the-art general image segmentation model that has been trained on a large natural image dataset. This model can segment objects using various interactive prompts, such as points, boxes, masks, and text, and it exhibits excellent generalized zero-shot learning and transfering capability. Based on SAM, SAM2[8] extends the promptable segmentation task to video. Although these models equip with astonishing zero-shot generalization ability, Some researches[2] reveal that like other foundational models, SAM is not effective in some vision tasks.

2.2. Parameter-Efficient Fine-Turning with adapter

PEFT has proven to be an efficient strategy for fine-tuning a fundamental model for a specific usage[10]. Compared to full fine-tuning, it keeps most of the parameters frozen

and learns fewer parameters. The concept of Adapters was first introduced in the NLP community[5] as a tool to fine-tune a large pre-trained model for each downstream task. Therefore, SAM-Adapter[2] is proposed, first introducing Adapter layer which is MLP layers in this research, to SAM. SAM-Adapter only tunes adapters with parameters of SAM transformer frozen. It is more time and resource-efficient and perform well in camouflaged objects and shadow detection.
Med-SA[9] also utilize adapters. Through leveraging parameter-efficient adaptation with simple yet effective SD-Trans and HyP-Adpt, it can process 3D medical images and incorporate visual prompt into the adapter, achieving substantial improvements over the original SAM model.

2.3. Parameter-Efficient Fine-Turning with LoRA

According to the research of LoRA[6], since adapter layers have to be processed sequentially, it will introduce inference latency. Therefore, LoRA, an efficient adaptation strategy that not introduces inference latency is introduced, based on the fact that the pre-trained language models have a low instrisic dimension. Researchers also find that adapt both W_q and W_v will yeild the best result.
The following researchers apply LoRA into SAM image encoder, which is SAMed[11], and achieved competitive results on Synapse dataset. However, SAMed do not use any prompt during inference, losing the important interactivity form multiple prompts of SAM.

3. Methodology

3.1. Classifier on SAM

Given an image $x \in \mathbb{R}^{H\times W\times C}$ together with a specific prompt, SAM leverages the provided prompt to generate a variable number of segmentation masks to address the ambiguity in segmentation prompts. However, this ambiguity prediction approach can also pose challenges in classification-related tasks where deterministic prediction is required. For instance, when directly applying SAM on a task aimed at segmenting only n objects belonging to k predefined classes from x and classifying them. SAM fails because it is designed to segment any potential objects, regardless of whether they belong to the target k classes, which introduces ambiguity in its prediction.
In order to perform deterministic prediction, we adopt the approach proposed in [11] and modify SAM's segmentation head in mask decoder by inserting several MLP layers. This modification allows SAM to predict x's corresponding segmentation map $\hat{S} \in \mathbb{R}^{H\times W}$ where each pixel belongs to an element in a predefined class list $Y = \{y_0, y_1, ..., y_k\}$ as close to the ground truth S as possible. Here we regard y_0 as the background class and $y_i, i \in \{1, ..., k\}$ as different classes of objects to be segmented out of x. Figure 1 shows the framework of our mask decoder.
Specifically, for k segmentation classes, including 1 background class and $k-1$ classes of objects, the mask decoder predicts k semantic masks $\hat{S}_l \in \mathbb{R}^{H\times W\times k}$ with each channel corresponding to a semantic label. The final predicted segmentation map $\hat{S}$ is generated as

$$\hat{S} = \text{argmax}(\text{Softmax}(\hat{S}_l, d = -1), d = -1). \quad (1)$$

where $d = -1$ means the Argmax and Softmax are performed across the channel dimension.

3.2. Loss Function

Since the modified SAM performs class-aware segmentation, we introduce cross entropy loss as a part of final loss, which is widely adopted in classification tasks. Following SAM's design, we use dice loss as the other part of final loss. The final loss can be described as

$$\mathcal{L} = \lambda_1 \text{CE}(\hat{S}_l, S) + \lambda_2 \text{Dice}(\hat{S}_l, S) \quad (2)$$

where $\text{CE}(\cdot)$ and $\text{Dice}(\cdot)$ refer to cross entropy loss and dice loss. S denotes the ground truth mask. λ_1, λ_2 are two weight parameters and we set them as 0.3, 0.7 respectively during our experiments.

3.3. SAM-LoRA

Given SAM's substantial size, full fine-tuning directly on the model requires significant computational resources. In contrast, LoRA fine-tuning updates only a small subset of parameters, while achieving notable performance gains with significantly lower resource consumption. We divide SAM into three components: image encoder, prompt encoder, mask decoder and fine tune them respectively. Figure 2 illustrates the overall architecture of LoRA fine-tuning in SAM.
SAM employs a Vision Transformer(ViT)[4] as its image encoder. We first freeze all parameters of the image encoder and then insert additional trainable LoRA layers to each of the transformer block. The details of these LoRA layers are shown in Figure 3. Following the strategy proposed in [11], we apply LoRA layers to the Q and V projection layers of each of the transformer block in image encoder.

Consider the input image token sequences $F \in \mathbb{R}^{B\times N\times C_{in}}$ and output token sequences $\hat{F} \in \mathbb{R}^{B\times N\times C_{out}}$, operated by a projection layer $W \in \mathbb{R}^{C_{out}\times C_{in}}$. LoRA fine-tuning freezes W, and adds a bypass layer consisting of two trainable linear layers: $A \in \mathbb{R}^{r\times C_{in}}$ and $B \in \mathbb{R}^{C_{out}\times r}$, where $r \ll min\{C_{in}, C_{out}\}$. The LoRA fine-tuning can be described as

$$\begin{aligned} \hat{F} &= \hat{W}F, \\ \hat{W} &= W + \Delta W = W + BA. \end{aligned} \quad (3)$$

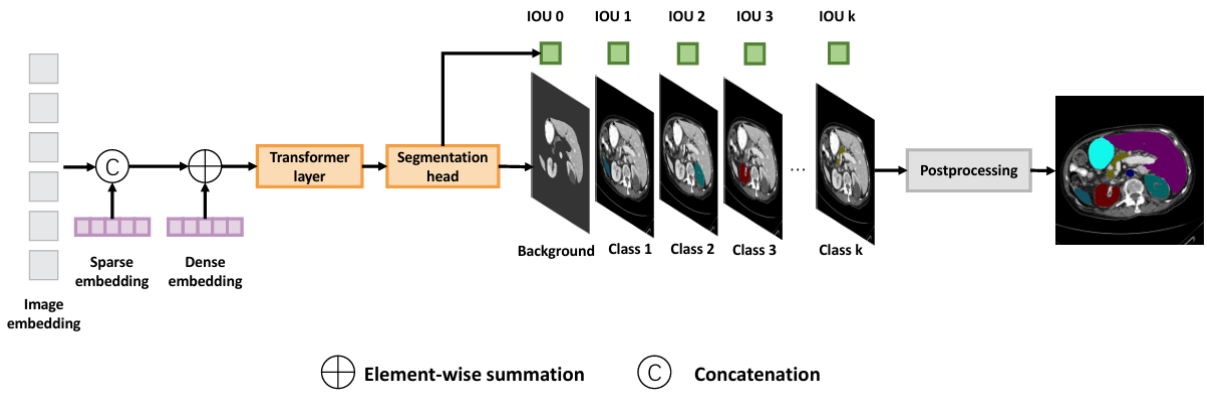

Figure 1. The detailed framework of mask decoder. The segmentation maps of each classes are generated individually.

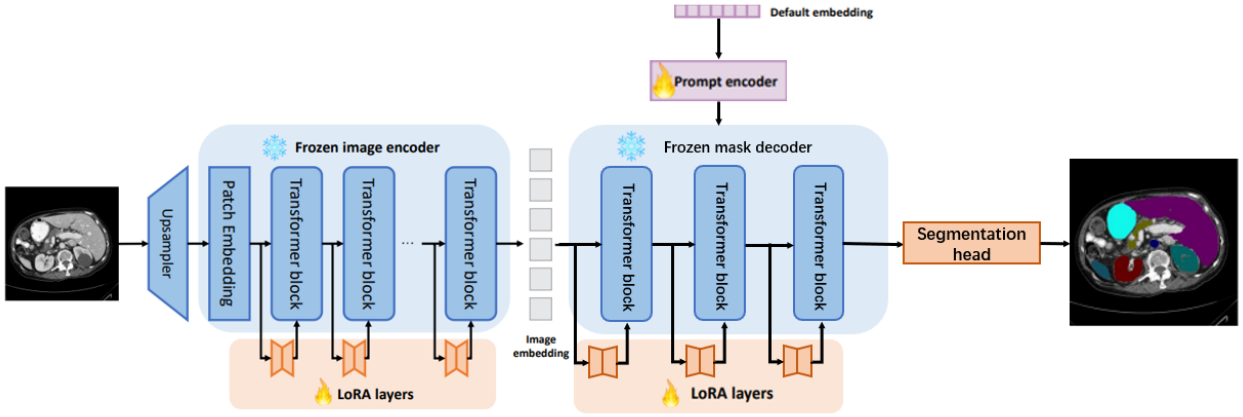

Figure 2. The overall architecture of SAM-LoRA.

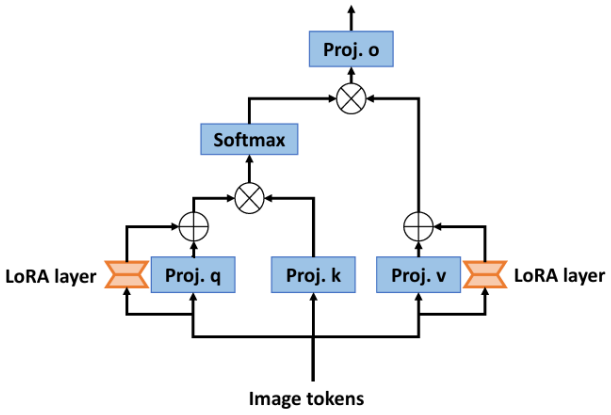

Figure 3. LoRA layers in image encoder.

In the multi-head self attention layer, LoRA fine-tuning can be expressed as

$$\begin{aligned} \mathrm{Att}(Q,K,V) &= \mathrm{Softmax}(\frac{QK^T}{\sqrt{C_{out}}}+B)V, \\ Q &= \hat{W}_q F = W_q F + B_q A_q F, \\ K &= W_k F, \\ V &= \hat{W}_v F = W_v F + B_v A_v F. \end{aligned} \tag{4}$$

Here, W_q, W_k, W_v are frozen projection layers and A_q, B_q, A_v, B_v are trainable LoRA layers.

For the prompt encoder, we perform full fine-tuning directly because of its lightweight structure. Additionally, we utilize a default prompt embedding throughout our experiments. Since our objective is task-specific, fine tuning all parameters of the prompt encoder on the default embedding makes the default prompt universally beneficial to segmentation in our target domain [11].

The mask decoder in SAM comprises a transformer layer and a segmentation head, as illustrated in Figure 1. We apply the same LoRA strategy as used for the image encoder

on the transformer layer and perform full fine-tuning on the segmentation head.

3.4. SAM-Adapter

In addition to LoRA, we also explore fine-tuning SAM using Adaption. Following the division of SAM components outlined in Section 3.3, we fine-tune each part respectively. Figure 4 illustrates the overall architecture of Adapter fine-tuning in SAM.

Following the methodology proposed in [9], we first freeze all parameters of the image encoder and then integrate two adapters into each transformer block. An adapter module consists of a down-projection MLP layer, ReLU activation and a up-projection MLP layer, as shown in Figure 4(b). The down-projection layer first compresses the embedding into a low dimension, while the up-projection layer expands the compressed embedding back to its original dimension. It can be described as

$$\hat{F} = \mathrm{ReLU}(FW_{down})W_{up} + F, \tag{5}$$

where W_{down} and W_{up} are two projection layer. F and $\hat{F}$ refer to input embedding and output embedding respectively. The first adapter is placed after the multi-head attention and the second adapter is placed in the residual path of the MLP layer that follows the multi-head attention. Additionally, the embedding is scaled by a scale factor $s = 0.5$ immediately after the second adapter [1].

For the prompt encoder, we adopt the same full fine-tuning strategy described in Section 3.3 with the default embedding.

In the mask decoder, two adapters are also introduced for each transformer block. The first adapter is positioned in the residual path of the MLP layer exactly the same way as in the image encoder. The second adapter is deployed after the residual connection of the image embedding-to-prompt cross-attention, as illustrated in Figure 4(c). However, we do not employ the Hyper-Prompting Adapter shown in Figure 4(c), as it is designed for incorporating different prompts, whereas our experiments use a single default prompt embedding.

4. Experimental Results

4.1. SAM on Medical Scenes

In this section, we focus on running segment anything model on medical scenarios, which is FLARE22 datasets, in order to classify the 13 organs in the specific images. The original code does not realize the recognition and segmentation of the fix categories, such as liver, kidney, etc. Therefore, for each class in the ground truth mask, we calculate the Dice Similarity Coefficient(DSC) with each of the predicted mask given by SAM, and select the largest one as the final prediction. The result is illustrated in Table 1.

Organs	SAM
Background	0.4678
Liver	0.5418
Right kidney	0.7789
Spleen	0.8002
Pancreas	0.2321
Aorta	0.7915
Inferior Vena Cava	0.4659
Right Adrenal Gland	0.0253
Left Adrenal Gland	0.1891
Gallbladder	0.4196
Esophagus	0.2759
Stomach	0.4558
Duodenum	0.2937
Left kidney	0.8049
Average	0.4673

Table 1. DSC of SAM on FLARE22

From the table, we find that the performance targeting at pancreas and adrenal gland is much less than the other, probably as a result of the inaccuracy of SAM classifying tiny objects. Nevertheless, as a whole, the average DSC of SAM to segment medical scenes is relatively good, up to about 0.467.

4.2. PEFT Algorithm

In this section, we conduct experiments among different models on the FLARE22 datasets and the results are shown in Table 2. Table 3 demonstrates the number of total parameters and trainable parameters of the two fine-tuning method. Similar to LoRA, a trainable segmentation head is added in the end to generate the final prediction. Compared to original model SAM, SAM Adapter and SAM Lora both experience great improvement in DSC, 0.095 and 0.099 respectively. Among the different rank, model with rank 4 shows advantage over the others, and its DSC comes to around 0.5667. As for this medical segmentation on FLARE22, we believe that matrix with $r = 1$ is too simple to capture the complicated mode and details in the fine tune tasks, while $r = 8$ might result in the over-fitting of the model, especially when the data set is not that large. Consequently, we find the model with rank 4 might be the best answer to this specific problem.

4.3. SAM vs SAM2

In this section, we primarily emphasize on the comparision between SAM and SAM2, and the experiment is still conduct on FLARE22 dataset, while the outcome is shown in Table 4.

From the data comparison, it is apparent that the perfor-

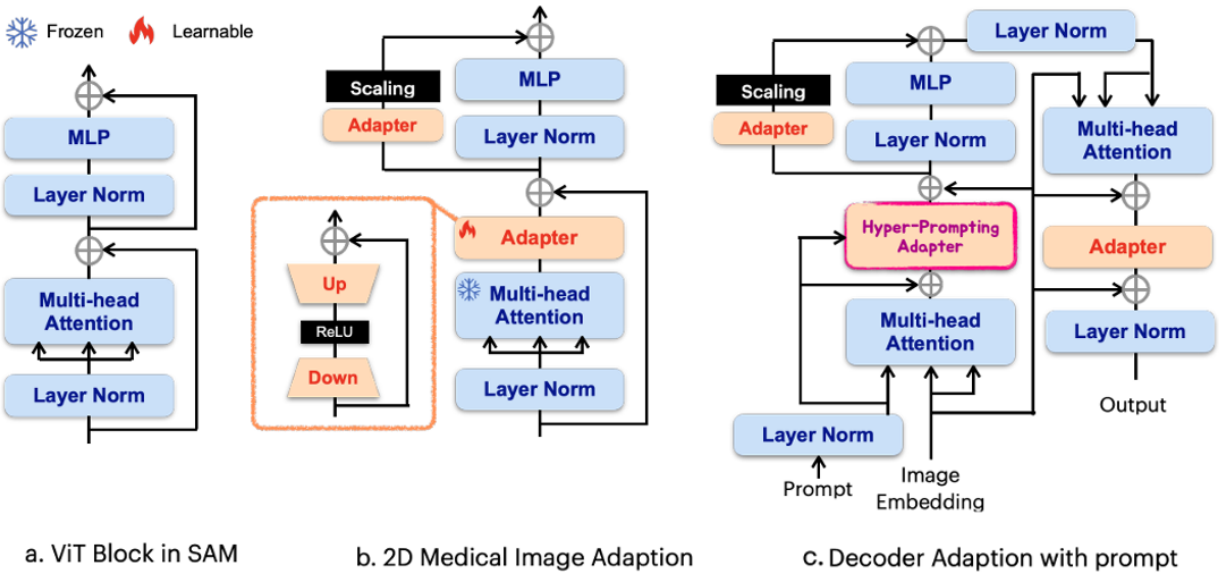

Figure 4. The overall architecture of SAM-Adapter.

Organs	SAM	SAM Adapter	SAM Lora(rank=1)	SAM Lora(rank=4)	SAM Lora(rank=8)
Background	0.4678	**0.9973**	0.9944	0.9961	0.9953
Liver	0.5418	**0.7328**	0.6758	0.7191	0.7093
Right kidney	**0.7789**	0.5536	0.5260	0.5224	0.4828
Spleen	**0.8002**	0.4130	0.3901	0.4406	0.4417
Pancreas	0.2321	**0.4307**	0.3826	0.4010	0.3985
Aorta	0.7915	**0.7985**	0.7759	0.7975	0.7872
Inferior Vena Cava	0.4659	**0.6611**	0.5818	0.6342	0.6309
Right Adrenal Gland	0.0253	0.1777	0.1778	**0.1914**	0.1850
Left Adrenal Gland	0.1891	**0.1993**	0.1761	0.1874	0.1538
Gallbladder	**0.4196**	0.2674	0.2546	0.2802	0.2578
Esophagus	0.2759	0.353	0.3368	**0.3650**	0.3654
Stomach	0.4558	0.6076	0.6270	**0.6897**	0.6554
Duodenum	0.2937	0.7392	0.6792	**0.7519**	0.7192
Left kidney	0.8049	0.9367	0.9265	**0.9580**	0.9333
Average	0.4673	0.5620	0.5360	**0.5667**	0.5511

Table 2. DSC of SAM, SAM Adapter, Sam Lora(rank=1, 4, 8) on FLARE22

Method	Total(M)	Trainable(M)
LoRA(r=1)	95.3	2.4
LoRA(r=4)	95.5	2.5
LoRA(r=8)	95.6	2.7
Adapter	106.1	10.8

Table 3. parameters of the two fine-tuning method.

mance of SAM2 is much poorer in all classification, and the average gap is roughly 0.163. According to the [8], SAM 2 struggles with accurately tracking objects with very thin or fine details and if there are nearby objects with similar appearance (e.g., multiple identical juggling balls). Besides, we speculate that the hyperparameter of SAM2 is much more complicated, resulting in the poor performance in generalization ability. In addition, we notice that only after fine-tuning SAM2 on the specific scenario, it shows strong segmentation capability, in the light of [3], which, to some extent, reflects the possibly low generalization ability exposed in our experiment.

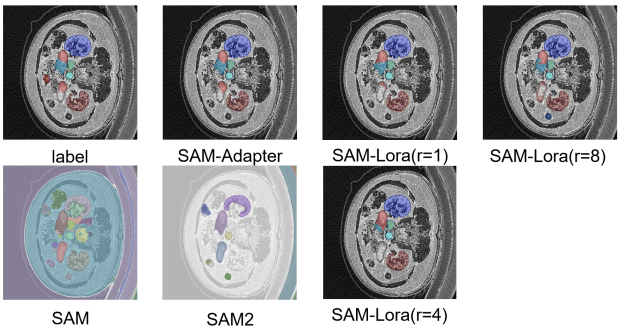

Figure 5. Visualization of Model results on FLARE22

Organs	SAM	SAM2
Background	**0.4678**	0.3357
Liver	**0.5418**	0.4255
Right kidney	**0.7789**	0.6290
Spleen	**0.8002**	0.6360
Pancreas	**0.2321**	0.0647
Aorta	**0.7915**	0.5825
Inferior Vena Cava	**0.4659**	0.2915
Right Adrenal Gland	**0.0253**	0.0081
Left Adrenal Gland	**0.1891**	0.0380
Gallbladder	**0.4196**	0.1520
Esophagus	**0.2759**	0.0609
Stomach	**0.4558**	0.2035
Duodenum	**0.2937**	0.1385
Left kidney	**0.8049**	0.6962
Average	**0.4673**	0.3044

Table 4. DSC of SAM and SAM2 on FLARE22

4.4. Downstream Scenario

In this section, on the basis of the previous work, we select the fine-tune model Lora with the rank of 4, which performs the best, to push on the Downstream Scenarios task. We also implement SAM, SAM2 on this specific work, expecting to find out more difference between those two models in details. And the findings are as follows.

In the first Scenario, which is the remote sensing image, the performance shown in Table 5 between SAM and SAM2 is similar, while SAM Lora sees a remarkable improvement, at about 0.142 evenly. It also shows a more powerful capacity of recognition of small and accurate target objects, buildings, water, farmland for instance. Moreover, we select a group of representative masks generated and visualize them in Figure 6 in order to reveal a much direct effect.

The second Scenario is the infrared image. As illustrated in Tabel 6, the results is basically in parallel with the first one. SAM Lora witness an about 0.3 improvement on SAM and a dramatic nearly 0.5 on SAM2, which again

Classification	SAM	SAM2	SAM Lora
Background	0.1059	0.0975	**0.8047**
Buildings	0.1742	0.1385	**0.6646**
Water	0.2771	0.3197	**0.4524**
Woods	**0.8107**	0.8027	0.3568
Lawn	**0.8367**	0.8249	0.3673
Farmland	0.2435	0.2325	**0.6411**
Average	0.4080	0.4026	**0.5478**

Table 5. DSC of SAM, SAM2 and SAM LORA on GID5

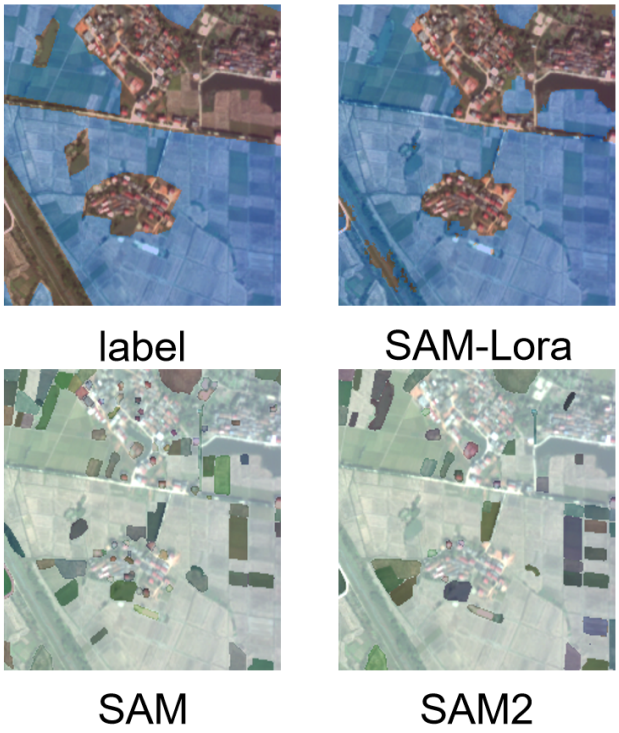

Figure 6. Visualization of Model results on GID5

Classification	SAM	SAM2	SAM Lora
Background	0.7614	0.4407	**0.9999**
Target	0.5203	0.4379	**0.8874**
Average	0.6409	0.4393	**0.9437**

Table 6. DSC of SAM, SAM2 and SAM LORA on SIRST

shows the strong ability of fine-tune SAM in image segmentation. Another difference, which is large gap between SAM and SAM2 deserves attention and might answer the previous expected question somehow. SAM2 should be the upgrade version of SAM, while it performs obviously worse on SIRST. More specific, from the visualized Figure 7, we

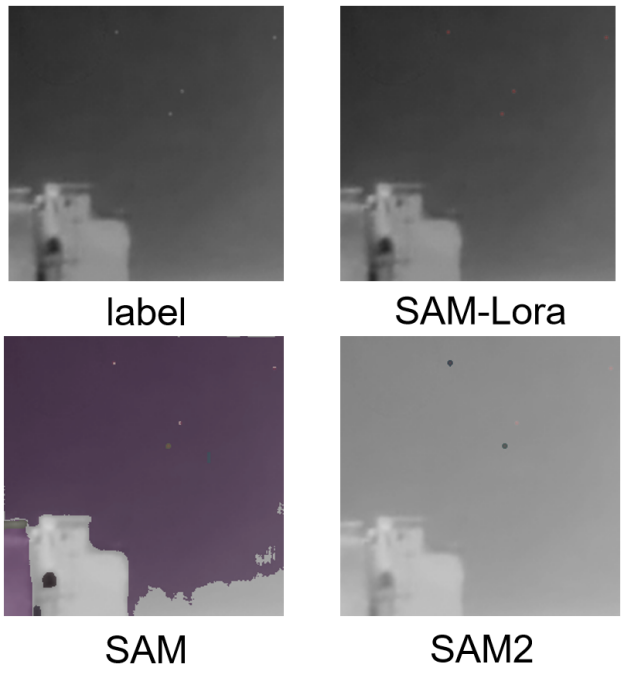

Figure 7. Visualization of Model results on SIRST

need to find four infrared points on the top left corner as shown in the label. SAM Lora locate them accurately, while SAM find five points and SAM2 only get two. From the previous work experience, we consider that SAM tend to segment everything it infer that might be a category, while SAM2 classify relatively small numbers of categories. We believe that this might to a large extent result from the lack of prior knowledge for both model have not pretrained on the dataset, which means that they do not understand the target they should find out and classify. Consequently, they can only segment the image aimlessly. SAM might add meaningless objects for its inclusive segmenting strategies, while the omission in SAM2 might be a trade-off for its compatibility for both image and video segmentation.

5. Conclusion

In this program, we lay an emphasis on the medical segmentation based on segment-anything model(SAM). Furthermore, we explore the difference between SAM and SAM2 and search for the potential reason. Meanwhile, we step into the realization of kinds of PEFT algorithms, Lora and Adapter to fine tune SAM. In addition, we implement out model in two downstream senarios, remote sensing image and infrared scenes and conduct the ablation study. Finally, we reflect on the limitation of segment-anything model and try to find a reasonable solution, which is supplemented in our future work.

5.1. Future Work

As for the question that how to cope with objects of irregular shape, we can first generate the original rough masks with the basic segmentation algorithm, and then extract certain quantities of outline points. After that, we can add these outline points as the special information of the irregular shaped objects to our prompt.

Through sufficient experiment and observing great amounts of visualized output images of SAM, we find that the the performance are often limited on the fine-grained segmentation and robustness to noise and artifacts. SAM depends on vision transformers(ViTs) to extract feature, effectively aggregating global features while lossing fine-grained, local details probably. Moreover, the ambiguity of boundaries might be serious in some specific scenarios, which should be optimized. We suggest using conditional random fields or level set methods to refine the boundaries and enhance edge precision through incorporating spatial and contextual information. As for the robustness to noise and artifacts, we can use pre-procession algorithms, such as Gaussian filtering, deep learning-based denoisers like DnCNN to clean the images before feeding them to SAM and train SAM on noisy data sets to improve its resilience to specific noise patterns and artifacts. We expect to conduct the experiments above and alleviate the limitations faced with SAM.

6. Contribution

Nan Jiang. Conduct the preliminary research, collect all the datasets(FLARE22, GID-MTL5/15, SIRST5k). Make the corresponding part of PPT.

Zhi Han. Implement the project code of SAM and SAM2, visualize of SAM and SAM2. Make the corresponding part of PPT, write the experimental results and conclusion of the report.

Hanwen Liu. Implement the project code of fine-tuning SAM, result calculation and visualization of LoRA-SAM and Adapter-SAM. Make the corresponding part of PPT, write the Methodology of the report.

Jingyao Wu. Conduct the preliminary research. Make the corresponding part of PPT and write the Abstract, Introduction and Related Works of the report.

References

[1] 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.

[2] Tianrun Chen, Lanyun Zhu, Chaotao Deng, Runlong Cao, Yan Wang, Shangzhan Zhang, Zejian Li, Lingyun Sun, Ying Zang, and Papa Mao. Sam-adapter: Adapting segment anything in underperformed scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3367–3375, 2023.

[3] Tianrun Chen, Ankang Lu, Lanyun Zhu, Chaotao Ding, Chunan Yu, Deyi Ji, Zejian Li, Lingyun Sun, Papa Mao, and Ying Zang. Sam2-adapter: Evaluating adapting segment anything 2 in downstream tasks: Camouflage, shadow, medical image segmentation, and more, 2024.

[4] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[5] 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*, pages 2790–2799. PMLR, 2019.

[6] 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.

[7] 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*, pages 4015–4026, 2023.

[8] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos, 2024.

[9] Junde Wu, Wei Ji, Yuanpei Liu, Huazhu Fu, Min Xu, Yanwu Xu, and Yueming Jin. Medical sam adapter: Adapting segment anything model for medical image segmentation. *arXiv preprint arXiv:2304.12620*, 2023.

[10] 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.

[11] Kaidong Zhang and Dong Liu. Customized segment anything model for medical image segmentation. *arXiv preprint arXiv:2304.13785*, 2023.