BLO-SAM: Bi-Level Optimization Based Finetuning of the Segment Anything Model for Overfitting-Preventing Semantic Segmentation

Li Zhang [1] **Youwei Liang** [1] **Ruiyi Zhang** [1] **Amirhosein Javadi** [1] **Pengtao Xie** [1]

Abstract

The Segment Anything Model (SAM), a foundation model pretrained on millions of images and segmentation masks, has significantly advanced semantic segmentation, a fundamental task in computer vision. Despite its strengths, SAM encounters two major challenges. Firstly, it struggles with segmenting specific objects autonomously, as it relies on users to manually input prompts like points or bounding boxes to identify targeted objects. Secondly, SAM faces challenges in excelling at specific downstream tasks, like medical imaging, due to a disparity between the distribution of its pretraining data, which predominantly consists of general-domain images, and the data used in downstream tasks. Current solutions to these problems, which involve finetuning SAM, often lead to overfitting, a notable issue in scenarios with very limited data, like in medical imaging. To overcome these limitations, we introduce BLO-SAM, which finetunes SAM based on bi-level optimization (BLO). Our approach allows for automatic image segmentation without the need for manual prompts, by optimizing a learnable prompt embedding. Furthermore, it significantly reduces the risk of overfitting by training the model's weight parameters and the prompt embedding on two separate subsets of the training dataset, each at a different level of optimization. We apply BLO-SAM to diverse semantic segmentation tasks in general and medical domains. The results demonstrate BLO-SAM's superior performance over various state-of-the-art image semantic segmentation methods. The code of BLO-SAM is available at https://github.com/importZL/BLO-SAM.

[1]Department of Electrical and Computer Engineering, University of California, San Diego, La Jolla, CA 92093, USA. Correspondence to: Pengtao Xie <p1xie@ucsd.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

1. Introduction

Semantic segmentation is a critical task in computer vision, which aims to assign each pixel with a semantic class (object classes such as dog and cat, or scene categories such as sky and ocean) (Mo et al., 2022). Deep learning has demonstrated great success in advancing the performance of semantic segmentation (Lateef & Ruichek, 2019). This progress has been further propelled by the emergence of foundation models (FMs) (Bommasani et al., 2021), a.k.a. large pretrained models, which have demonstrated unprecedented prevalence across diverse tasks, including vision (Radford et al., 2021; Moor et al., 2023), language (Brown et al., 2020; Touvron et al., 2023a;b), and multi-modality (Alayrac et al., 2022; Wang et al., 2022). Building on a data engine using 11 million image-mask pairs, the Segment Anything Model (SAM) (Kirillov et al., 2023) emerges as a noteworthy segmentation foundation model, and demonstrates strong capabilities in segmenting diverse natural images. As a novel promptable segmentation model, SAM distinguishes itself by yielding desired segmentation masks when provided with appropriate points or bounding boxes as prompts (detailed descriptions about SAM can be found in Appendix B).

Despite its strong performance in producing accurate segmentation masks based on prompts, SAM faces a notable limitation - it cannot autonomously segment specific objects, as it requires points or bounding boxes as manual prompts[1]. For example, if we would like to segment lungs from a chest X-ray image, we need to provide at least one point on the lung region or bounding boxes enclosing the lungs to indicate that the objects aimed to segment are the lungs. Another significant challenge of applying SAM for downstream segmentation tasks arises from the distribution discrepancy between the pretraining data of SAM and the data in downstream tasks. On tasks where the data distribution deviates from the pretraining data, SAM struggles to segment the desired objects accurately even with proper prompts (Mazurowski et al., 2023; He et al., 2023).

To address the distribution discrepancy issue, Med-SA (Wu

[1]Although the SAM paper mentions that SAM can accept textual prompts, only points, bounding boxes, and coarse masks are supported as prompts in its public codebase.

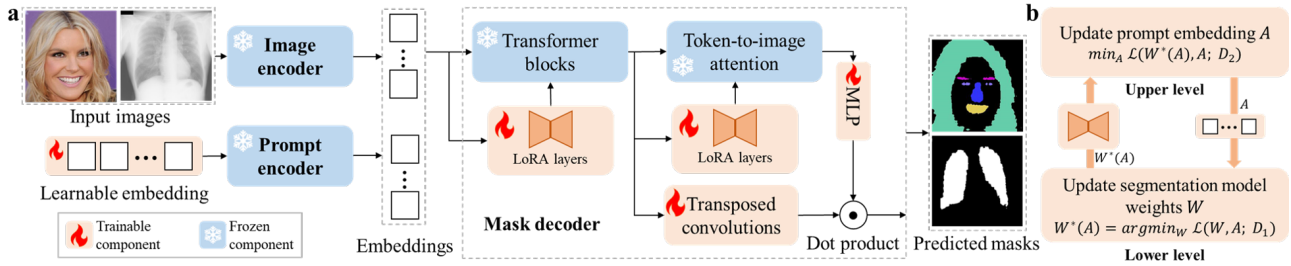

Figure 1. **(a)** BLO-SAM Overview: The majority of original SAM parameters are frozen. LoRA layers enable parameter-efficient finetuning of the mask decoder's Transformer layers. Learnable parameters are categorized into two groups: learnable model parameters in the mask decoder and learnable embeddings as input to the prompt encoder. **(b)** Two parameter groups are updated through two interdependent optimization problems.

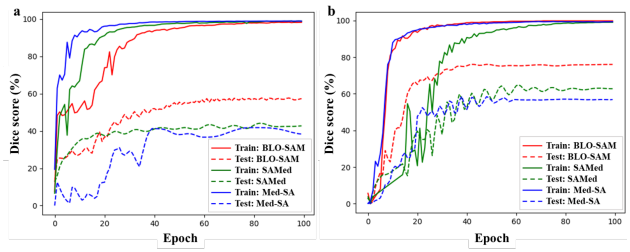

Figure 2. Dice scores on the train and test sets versus training epochs, on **(a)** gastrointestinal disease and **(b)** human body segmentation tasks. The train and test curves for SAMed exhibit a large gap between epochs 60 and 100, indicating that severe overfitting to the training data occurs. A similar pattern is observed in the case of Med-SA. Conversely, the train-test gap for BLO-SAM is much smaller compared to SAMed and Med-SA, underscoring our method's effectiveness in mitigating overfitting.

et al., 2023) utilizes Adapter (Houlsby et al., 2019) to finetune SAM's image encoder and mask decoder in downstream tasks, resulting in superior performance over several SOTA segmentation methods on various medical datasets. Nevertheless, Med-SA still needs prompts manually provided by humans or generated from an initial segmentation mask, during both training and inference phases, making it less practical in real-world applications. To address both challenges (i.e., distribution discrepancy and the need for manual prompts), SAMed (Zhang & Liu, 2023) finetunes SAM with a default prompt (i.e., a learnable vector) instead of manual prompts, directly outputting multiple segmentation masks for all the classes. However, the above methods still have limitations. They bear a risk of overfitting when labeled data are very limited in downstream tasks. For example, in medical domains, images with labeled segmentation masks are often scarce, either due to the limited number of input images (e.g., for privacy concerns) or the difficulty of obtaining segmentation masks (e.g., requiring domain experts to annotate, which is time-consuming and expensive). Finetuning large foundation models like SAM on these label-scarce settings often leads to overfitting to training data and poor generalization to test images. As demonstrated in our experiments (see Fig. 2), SAMed and Med-SA suffer from severe overfitting and low test performance.

In response to the above challenges, we introduce BLO-SAM, a finetuning method for SAM that addresses the overfitting issue based on bi-level optimization (BLO). BLO-SAM combats overfitting by updating two separate sets of learnable parameters on two splits of the training data. Illustrated in Fig. 1(a), BLO-SAM involves two sets of parameters: i) segmentation model's weight parameters, including LoRA layers (Hu et al., 2021), transposed convolutions, and MLP heads, and ii) the prompt embedding. These parameters undergo optimization through two levels of nested optimization problems, as shown in Fig. 1(b). In the lower level, we iteratively train segmentation model weights by minimizing a finetuning loss on a designated subset of the training dataset while keeping the learnable prompt embedding fixed. Following this finetuning, we transition to the upper level to validate the model's effectiveness on the remaining subset of the training data. The prompt embedding is then updated by minimizing the validation loss. By segregating the learning processes for different learnable parameters onto distinct data subsets within two optimization problems, our method effectively mitigates the risk of overfitting to a single dataset, enhancing the model's generalization on test examples.

BLO-SAM's mechanism of combating overfitting is inspired by the well-established practice of hyper-parameter tuning (Franceschi et al., 2018). Typically, a model's weight parameters (such as weights and biases in a neural network) are trained on a training dataset, while the hyper-parameters (like the number of layers in a neural network) are tuned on a separate validation set, which prevents overfitting the training data. If hyper-parameters were also tuned on the training set, it would lead to significant overfitting. Similarly, in the context of SAM, prompt embeddings can be regarded as a form of 'hyper-parameters'. Overfitting arises when these embeddings, along with the segmentation model weights, are optimized together by minimizing a loss function on a single dataset, as SAMed do. Our BLO-SAM follows the correct way of 'hyper-parameter tuning': it optimizes the 'hyper-parameters' - the prompt embedding - on a 'validation set' - a subset of the training data, while training the

segmentation model weights on the other subset.

Our work makes two key contributions:

- We propose BLO-SAM, an overfitting-resilient approach for finetuning SAM with only a few training examples. We propose to learn the prompt embedding and model parameters of SAM on two distinct subdatasets in a BLO framework, effectively combating overfitting in ultra-low data regimes. Furthermore, our method enables fully automated segmentation without the need for manual prompts during inference and training. This substantially enhances the practical applicability of SAM in real-world scenarios.
- Our extensive experiments on six datasets from general domains and medical domains demonstrate the strong effectiveness of BLO-SAM with less than 10 training examples. Our method significantly outperforms vanilla SAM, SAM-based models, few-shot semantic segmentation methods, and popular segmentation models, without requiring manual prompts.

2. Related Work

2.1. Semantic Segmentation

In the realm of semantic segmentation, substantial progress has been made recently, particularly within the context of deep learning-based methods. Many works focus on the architectural design of segmentation neural networks, such as fully convolutional networks (FCNs) (Long et al., 2015), U-Net (Ronneberger et al., 2015), DeepLab (Chen et al., 2017a), and SegFormer (Xie et al., 2021). The incorporation of attention mechanisms, exemplified in Atrous Spatial Pyramid Pooling (ASPP) (Chen et al., 2017a) and non-local neural networks (Wang et al., 2018), has further improved contextual understanding in semantic segmentation. Recently, the Segment Anything Model (SAM) (Kirillov et al., 2023) emerged as an FM for image segmentation. SAM uses an MAE-pretrained ViT (He et al., 2022) to encode the input image, positional embedding to encode prompts, and a lightweight Transformer (Vaswani et al., 2017) based mask decoder to produce high-quality segmentation masks. Please see Appendix B for details.

As discussed in Section 1, multiple works (Mazurowski et al., 2023; He et al., 2023) have shown SAM demonstrates reduced effectiveness in downstream tasks when there is a noticeable difference between the data it was pretrained on and the task-specific data. For example, He et al. (2023) studied the performance of SAM on 12 medical segmentation datasets, showing SAM performs significantly worse than five SOTA algorithms on some datasets. The findings underscore the necessity of finetuning SAM when it is applied to domains other than natural images. To address this, MedSAM (Ma & Wang, 2023), Med-SA (Wu et al., 2023), and SAMed (Zhang & Liu, 2023) finetune SAM on medical images, outperforming several SOTA methods. To enable prompting SAM using texts, LISA (Lai et al., 2023) integrates a multi-modal large language model, LLaVA (Liu et al., 2023a), with SAM. In addition, Yang et al. (2023) and Cheng et al. (2023) further extend SAM to video segmentation and object tracking in videos, demonstrating SAM's versatility across a range of visual tasks

2.2. Model Finetuning

Finetuning techniques have demonstrated state-of-the-art effectiveness in adapting large-scale foundation models for specialized downstream tasks (Radford et al., 2018; Devlin et al., 2018). However, as foundation models grow in size, finetuning all parameters becomes increasingly expensive (Brown et al., 2020; Touvron et al., 2023a; Kirillov et al., 2023) in computation. Therefore, various parameter-efficient finetuning methods have been proposed to alleviate this issue. For instance, Adapter (Houlsby et al., 2019) injects trainable adapter layers into pretrained Transformers (Vaswani et al., 2017). Low-Rank Adaptation (LoRA) (Hu et al., 2021) adds learnable low-rank matrices to the pretrained weight matrices of foundation models and only optimizes the low-rank matrices in the finetuning stage with the pretrained parameters frozen. Zhang et al. (2023a) proposes to adaptively allocate budgets for updating different LoRA layers based on their importance scores when adapting to a specific downstream task. Prompt-tuning (Lester et al., 2021) proposes optimizable 'soft prompts' for a specific downstream task and only optimizes the trainable prompts during finetuning. P-tuning (Liu et al., 2023b) finetunes the pretrained foundation model by training a neural network to generate prompt embeddings while keeping the pretrained parameters frozen. IA3 (Liu et al., 2022) proposes to multiply the output of activation layers in the pretrained foundation models with trainable vectors and optimize these vectors in the finetuning stage. Contrary to these methods that emphasize parameter efficiency, our approach is centered on mitigating overfitting. It is designed to complement these methods rather than replace them.

2.3. Bi-Level Optimization

Bi-level optimization (BLO) refers to a class of optimization problems in which one optimization problem (lower level) is nested within another optimization problem (upper level) (Sinha et al., 2017). Various tasks can be formualted in a BLO framework, including meta-learning (Finn et al., 2017; Killamsetty et al., 2022; Qin et al., 2023), neural architecture search (Liu et al., 2018; Hosseini & Xie, 2022), and data reweighting (Shu et al., 2019; Garg et al., 2022; Hosseini et al., 2023). In these tasks, model weights are usually learnt during lower-level optimization on the training split of the dataset, while meta-variables like hyper-parameters

or architectures are learnt in the upper-level optimization on a separate validation split to alleviate the issue of overfitting.

Numerous gradient-based optimization algorithms and software have been developed for solving BLO problems. For example, Liu et al. (2018) develop a finite difference approximation method to efficiently compute the gradients with regard to the upper level variables in BLO problems. Finn et al. (2017) propose to compute the gradient updates of meta variables directly with iterative differentiation (Grazzi et al., 2020). Choe et al. (2022) develop a software for users to easily and efficiently compute gradients within BLO problems with different approximation methods.

3. Method

3.1. Overview of BLO-SAM

Our approach, BLO-SAM, finetunes SAM for downstream semantic segmentation tasks using bi-level optimization (BLO). In the pretrained SAM model, certain parameters undergo finetuning, whereas others remain static throughout the finetuning process, as shown in Fig. 1. Furthermore, to eliminate the need for manual prompts, we learn a prompt embedding vector. To combat overfitting, we split the training set into two halves (denoted by D_1 and D_2), which are used for learning SAM's finetunable parameters and the prompt embedding, respectively. In the lower-level of our BLO framework, SAM's finetunable parameters (denoted by W) are optimized on the sub-dataset D_1, with the prompt embedding (denoted by A) tentatively fixed. The optimal solution $W^*(A)$, dependent on A, is then passed to the upper level. In the upper-level optimization, $W^*(A)$ is validated on the sub-dataset D_2. The validation loss, which is a function of the prompt embedding A, indicates the generalization performance of the finetuned SAM. We optimize A to minimize this validation loss for reducing the risk of overfitting and improving generalization. The two levels of problems share the same form of loss function designed for semantic segmentation. The two levels are optimized iteratively until convergence, as shown in Algorithm 1.

Finetuning SAM with LoRA. We employ Low-Rank Adaptation (LoRA) (Hu et al., 2021) for parameter-efficient finetuning of SAM. LoRA introduces an additional learnable matrix (known as a LoRA layer), which is of lower rank, as an update to the existing pretrained weight matrix. During finetuning, the focus is on optimizing this lower-rank matrix, while keeping the pretrained matrix static. Notably, the LoRA layer comprises considerably fewer parameters than the original matrix, enhancing the efficiency of the finetuning process. As shown in Fig. 3, a LoRA layer is added to each query and value projection layer of all attention blocks in SAM's mask decoder, including self-attention, image-to-token attention, and token-to-image attention. Each LoRA layer consists of two sequential linear layers where the first one projects the input token to a low-dimensional space and the second one projects from the low-dimensional space back to the original feature space so that the output of LoRA can be added to the output of the frozen Transformer layer (Fig. 3). For different attention blocks, the input tokens differ. In the self-attention block, the prompt embedding serves as all three input tokens. For image-to-token attention blocks, the prompt embedding is used for both key and value projections, while the image embedding is used for the query projection. Conversely, in the token-to-image attention blocks, the image embedding is employed for key and value projections, and the prompt embedding is utilized for the query projection. In the end, the learnable parameters of SAM encompass those in the LoRA layers, the transposed convolutions, and the multi-layer perceptron (MLP) head located in the mask decoder, with all other model parameters frozen, as shown in Fig. 1(a).

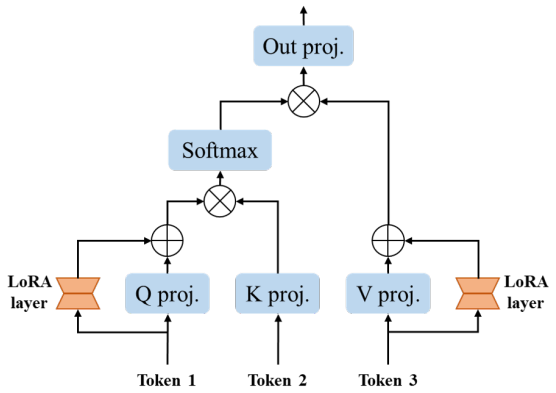

Figure 3. Finetune SAM with LoRA. We add LoRA layers to query and value projection layers within the attention block of SAM's mask decoder. 'Q proj.', 'K proj.', 'V proj.', and 'Out proj.' denote the projection layers for the query, key, value, and output.

3.2. Bi-Level Finetuning Framework

Lower-Level Optimization Problem. In the lower level, we tentatively fix the prompt embedding A and optimize SAM's finetunable parameters W on the first sub-dataset D_1 by minimizing the following loss:

$$\mathcal{L} = (1-\lambda)\mathcal{L}_{CE}(W, A; D_1) + \lambda\mathcal{L}_{Dice}(W, A; D_1), \quad (1)$$

where $\mathcal{L}_{CE}$ and $\mathcal{L}_{Dice}$ represent cross-entropy loss and dice loss, respectively, between predicted masks and ground-truth masks, and λ is a tradeoff parameter. The $(W, A; D_1)$ arguments indicate the loss depends on the model parameters and prompt embedding while computed on dataset D_1. The lower level aims to solve the following problem:

$$W^*(A) = \arg\min_W \mathcal{L}(W, A; D_1). \quad (2)$$

$W^*(A)$ implies the optimal solution W^* depends on A as W^* depends on the loss function which depends on A.

Algorithm 1 Optimization Algorithm for BLO-SAM

1: **Input:** Sub-datasets D_1, D_2. Learning rates η_1, η_2. Parameter initialization $W^{(0)}$, $A^{(0)}$.
2: **for** $t = 1, 2, 3, \cdots$ **do**
3: Sample a batch from D_1. Update $W^{(t)}$ via Eq. (2).
4: Sample a batch from D_2. Update $A^{(t)}$ via Eq. (3).
5: **end for**

Upper-Level Optimization Problem. In the upper level, we validate the finetuned model $W^*(A)$ on the second sub-dataset D_2. The learnable prompt embedding is updated by minimizing the validation loss:

$$\min_A \mathcal{L}(W^*(A), A; D_2), \tag{3}$$

which is the same loss function as Eq. (1) but with D_1 replaced by D_2 and W replaced by $W^*(A)$. The key in Eq. 3 is that the loss depends on $W^*(A)$ which also depends on A. Thus, we need to "unroll" $W^*(A)$ to correctly compute the gradient w.r.t. A, as detailed in Appendix A.

Bi-Level Optimization Framework. Integrating the aforementioned two optimization problems, we have a bi-level optimization framework:

$$\begin{aligned} \min_A \quad & \mathcal{L}(W^*(A), A; D_2) \\ s.t. \quad & W^*(A) = \arg\min_W \mathcal{L}(W, A; D_1) \end{aligned} \tag{4}$$

In this framework, the two optimization problems are interdependent. The output of the lower level, $W^*(A)$, serves as the input for the upper level. The optimization variable A in the upper level is used in the lower-level loss function.

3.3. Optimization Algorithm

We employ a gradient-based optimization algorithm to solve the problem in Eq. (4). Drawing inspiration from Liu et al. (2018), we perform a one-step gradient descent for Eq. (2) to approximate the optimal solution $W^*(A)$. Subsequently, we substitute the approximation of $W^*(A)$ into the upper-level optimization problem. A is updated by minimizing the approximated upper-level loss function via gradient descent. These steps constitute one global optimization step. We iteratively perform global steps between the lower level and upper level until convergence, as shown in Algorithm 1. Detailed derivations are provided in Appendix A.

4. Experiments

In this section, we evaluate BLO-SAM across a diverse set of semantic segmentation tasks from both general domains and medical domains, including human face components segmentation, car components segmentation, human body segmentation, teeth segmentation, gastrointestinal disease segmentation, and lung segmentation. To be compatible with SAM, each multi-class segmentation task is converted to multiple binary segmentation tasks, one for each class. Our experiments focus on ultra-low data regimes, where the number of training examples is less than ten.

4.1. Datasets

For the human facial components segmentation task, we employed the CelebAMask-HQ dataset (Lee et al., 2020), a collection of high-resolution face images accompanied by segmentation masks of various facial components including brow, eye, hair, nose, and mouth. For the car segmentation task, we used the dataset from David (2020), which contains images of cars and their segmentation masks with four semantic components: car body, wheel, light, and windows. We only used the car body, wheel, and windows as the 'light' category is missing in many data examples. For human body segmentation, we used the TikTok dances dataset (Roman, 2023). For teeth, gastrointestinal disease, and lung segmentation tasks, we utilized the children's dental panoramic radiographs dataset (Zhang et al., 2023b), Kvasir-SEG dataset (Jha et al., 2020), and JSRT dataset (Shiraishi et al., 2000), respectively. Every dataset is comprised of a training set and a test set. By randomly sampling a small number of examples from the original training set, we create a new few-shot training set which is then used to train our model and baselines. More details about the datasets can be found in Appendix C. We repeated each sampling three times at random, and the mean performance over the test set is reported in the main paper, while standard deviations are detailed in Appendix F. In our method, the sampled new training set is further randomly split into two subsets D_1 and D_2 with equal size. Baseline methods utilize the entire new training set without any subdivision.

4.2. Experimental Settings

Baselines and Metrics. We compared our method with a variety of baselines, including supervised, few-shot, and SAM-based approaches. The supervised methods include DeepLabV3 (Chen et al., 2017b), a widely adopted model in semantic segmentation, and SwinUnet (Cao et al., 2022), an UNet-like transformer designed for medical image segmentation. Few-shot learning methods include HSNet (Zhang et al., 2022) which uses cross-semantic attention to bridge the gap between low-level and high-level features, and SSP (Fan et al., 2022) which introduces a self-support matching strategy to capture consistent underlying characteristics of query objects. SAM-based baselines include vanilla SAM (Kirillov et al., 2023), Med-SA (Wu et al., 2023) and SAMed (Zhang & Liu, 2023) which use Adapter (Houlsby et al., 2019) and LoRA (Hu et al., 2021) for SAM finetuning respectively. Med-SA and SAMed update all learnable parameters on a single training set. For vanilla SAM, we con-

Table 1. Average Dice score (%) on the CelebAMask dataset, with different numbers of training examples. Standard deviations are in Appendix F. The overall performance is calculated by averaging the results for different facial components. The last column indicates whether manual prompts are needed.

Method	Training with 4 labeled examples						Training with 8 labeled examples						Manual Prompts
	Overall	Brow	Eye	Hair	Nose	Mouth	Overall	Brow	Eye	Hair	Nose	Mouth	
DeepLab	49.5	33.2	55.9	55.2	55.8	47.5	54.9	37.3	59.7	58.0	64.0	55.7	✗
SwinUnet	35.2	21.7	21.7	46.7	44.4	41.6	42.4	28.8	33.1	52.7	47.6	49.8	✗
HSNet	49.9	30.1	41.9	58.9	59.6	59.0	60.1	43.6	58.1	76.6	58.2	64.2	✗
SSP	56.9	**40.3**	33.6	73.2	71.6	65.6	60.0	45.4	31.2	76.6	74.0	72.7	✗
SAM	32.9	20.8	30.6	44.0	44.1	25.2	32.9	20.8	30.6	44.0	44.1	25.2	✓
Med-SA	62.9	36.3	63.6	77.6	71.0	66.0	67.7	42.9	65.5	82.4	74.6	**73.1**	✓
SAMed	58.2	28.3	55.9	78.5	65.1	63.0	65.0	39.5	66.0	82.4	70.5	66.4	✗
BLO-SAM	**65.9**	39.4	**65.5**	**82.8**	**74.3**	**67.6**	**69.9**	**45.8**	**71.1**	**83.6**	**76.1**	72.7	✗

Table 2. The comparison of BLO-SAM with baselines on the car components and human body datasets evaluated by Dice Score (%) with different numbers of training examples. Standard deviations are in Appendix F. The overall performance is calculated by averaging the results for different components.

Method	Car components								Human body	
	Training with 2 labeled examples				Training with 4 labeled examples				4 examples	8 examples
	Overall	Body	Wheel	Window	Overall	Body	Wheel	Window		
DeepLab	46.5	58.5	55.0	26.1	59.8	62.9	66.6	49.8	31.8	37.0
SwinUnet	31.4	22.2	33.4	38.7	47.3	49.3	53.7	38.9	30.9	56.8
HSNet	51.0	67.1	32.4	53.4	68.8	70.8	70.3	65.4	50.6	55.8
SSP	64.1	62.8	**76.2**	53.3	72.2	77.7	78.2	60.6	58.9	76.5
SAM	35.1	40.8	41.7	22.9	35.1	40.8	41.7	22.9	26.0	26.0
Med-SA	67.3	80.8	70.9	50.3	75.9	84.4	78.1	65.3	58.8	80.6
SAMed	60.4	78.8	65.0	37.5	74.0	85.3	72.5	64.2	63.8	81.5
BLO-SAM	**71.1**	**83.2**	74.5	**55.6**	**78.3**	**86.3**	**79.9**	**68.6**	**76.3**	**85.1**

structed three different types of prompts from each ground-truth mask, including a positive point from the foreground, a negative point from the background, and bounding boxes of target objects. For Med-SA, its prompts are extracted from the ground-truth masks as well. During inference, both SAM and Med-SA necessitate user-generated prompts, a requirement that is unfeasible when test images are numerous. Following the evaluation protocols of SAM and Med-SA, these prompts are derived in advance from the ground-truth masks. It is important to recognize that such an approach is impractical, as it needs to access the ground-truth masks of test images. Nevertheless, they remain the most accessible baselines we can compare against. All experiments are conducted on an A100 GPU with 80G memory.

We used the Dice score as the evaluation metric. The Dice score is defined as $\frac{2|A\cap B|}{|A|+|B|}$, where A and B represent the predicted and ground truth mask respectively.

Hyper-parameters. In our method, the trade-off parameter λ in Eq. (1) was set to 0.8 following the setting in Zhang & Liu (2023). LoRA layers and other non-frozen model components were optimized in the lower level using the AdamW optimizer (Loshchilov & Hutter, 2017). We set the initial learning rate to $5e$-3, betas to $(0.9, 0.999)$, and weight decay to 0.1. The learning rate in the i-th iteration followed the formula (Zhang & Liu, 2023): $\text{lr}_i = \text{lr}_0(1-i/\text{Iter}_{\text{max}})^{0.9}$. Here, lr_0 and Iter_{max} represent the initial learning rate and maximal training iteration, respectively. For updating the prompt embedding in the upper level, we used the same settings as in the lower-level optimization. We set the number of training epochs to 100 and selected the best checkpoint based on segmentation performance on the D_2 sub-dataset.

4.3. Results and Analysis

Human Facial Components Segmentation. In our evaluation of BLO-SAM for human facial components segmentation on the CelebAMask dataset, we have two settings with different numbers (4 or 8) of training examples. The results presented in Table 1 reveal some noteworthy observations.

Firstly, BLO-SAM demonstrates its strong capability to finetune SAM for accurate facial component segmentation even with a very small number of training examples. For instance, with just 4 labeled examples (2 for D_1 and the other 2 for

Table 3. The comparison of BLO-SAM with baselines on the teeth, gastrointestinal disease and lung datasets evaluated by Dice Score (%) with different numbers of training examples. Standard deviations are in Appendix F. The last three columns show the total number of model parameters, the number of trainable parameters (in millions), and the training time (in GPU hours).

Method	Teeth		Kvasir		Lung		Total	Trainable	Train Cost
	4 examples	8 examples	4 examples	8 examples	2 examples	4 examples	Param(M)	Param(M)	(GPU hours)
DeepLab	56.8	63.9	31.9	37.8	61.1	76.4	41.9	41.9	0.27
SwinUnet	28.6	50.6	37.8	39.0	62.1	76.4	27.2	27.2	0.29
HSNet	70.2	72.2	30.0	35.1	83.1	84.5	28.1	2.6	0.16
SSP	33.7	51.7	27.4	27.7	83.2	90.1	8.7	8.3	0.22
SAM	21.2	21.2	14.7	14.7	31.4	31.4	91.0	0	0
Med-SA	69.8	75.1	33.5	59.3	82.9	91.7	104.4	10.7	0.62
SAMed	69.9	76.4	42.7	57.1	84.4	91.8	91.0	1.1	0.46
BLO-SAM	**73.2**	**77.3**	**59.7**	**61.6**	**87.1**	**93.7**	91.0	1.1	0.57

D_2), BLO-SAM achieves an overall Dice score of 65.9%, and the score improves to 69.9% when the training dataset size is increased to 8. The effectiveness of BLO-SAM extends to individual facial components, exemplified by the 82.8% Dice score for hair segmentation with only 4 training examples, achieved without any manual input prompts.

Secondly, our BLO-SAM method demonstrates a significant improvement over the standard SAM. For instance, when finetuned with just 4 labeled examples, BLO-SAM dramatically improves the Dice score of SAM from 32.9% to 65.9%. These results clearly indicate the superiority of our method as an effective finetuning approach for SAM. It adeptly adapts the pretrained SAM to the specific data distributions of downstream tasks, using just a few examples.

Thirdly, BLO-SAM surpasses other SAM-based methods, including Med-SA and SAMed, which also finetune the pretrained SAM for specific semantic segmentation tasks. Unlike Med-SA or SAMed, which update all learnable parameters on a single training set, BLO-SAM employs bi-level optimization to update segmentation model weights and prompt embedding on disjoint subsets of the training data, effectively mitigating the risk of overfitting. While Med-SA's performance is close to that of our method, it requires manual prompts from users during testing. This requirement substantially restricts its practicality in real-world applications. In contrast, our method operates entirely autonomously, eliminating the need for manual prompting.

Finally, compared to general supervised methods such as DeepLab and SwinUnet, BLO-SAM shows a substantial advantage when trained with very few examples. For instance, with 4 labeled examples, DeepLab achieves an overall Dice score of 49.5%, whereas BLO-SAM attains a substantially higher Dice score of 65.9%. BLO-SAM also outperforms few-shot learning methods including HSNet and SSP, except for brow segmentation with 4 examples. This superiority is attributed to its ability to transfer the capacity of SAM to downstream tasks via finetuning.

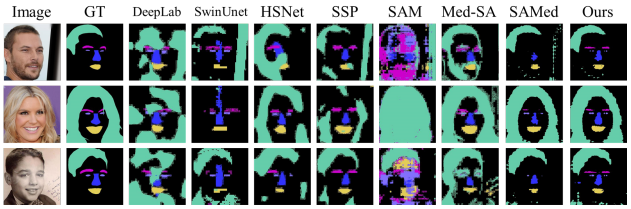

Figure 4. Qualitative results on some randomly sampled test images from the CelebAMask dataset. 'GT' denotes ground-truth segmentation masks.

Fig. 4 shows some qualitative comparisons. We can see that BLO-SAM generates more accurate segmentation masks compared to the baselines.

Car Segmentation and Human Body Segmentation. We further assess BLO-SAM's performance in segmenting car components. Table 2 demonstrates that BLO-SAM surpasses all baseline methods in terms of the overall Dice score. For individual components, BLO-SAM excels over the baselines in most categories, with only an exception in 'wheel'. In the task of human body segmentation, BLO-SAM similarly outperforms the baseline models, as evidenced in Table 2. Again, the superiority of our method lies in its strong ability to adapt SAM to the data distributions of downstream tasks with just a few data examples, while combating overfitting using the BLO framework and eliminating the need for manual prompts. We further increased the number of training examples in the body segmentation task. The results are deferred to Appendix I.

Medical Image Semantic Segmentation. To further evaluate our method, we perform experiments on several medical image segmentation tasks, where there may be a significant distribution shift from the pretraining data of SAM to the medical datasets. This distribution shift can significantly impair SAM's capability in the medical domain. We perform experiments on teeth, gastrointestinal disease, and lung segmentation tasks and report the results in Table 3. For all medical segmentation tasks, BLO-SAM attains the

Table 4. The comparison of BLO-SAM with baselines for out-of-domain performance.

	Human body	Car body	Teeth	Lung
SAM	26.0	40.8	21.2	31.4
SAMed	40.4	51.1	44.5	59.5
BLO-SAM	**69.9**	**68.2**	**65.5**	**68.9**

best performance compared with baselines. The advantage of BLO-SAM over the baselines is more evident in the 4-example case than in the 8-example case, which underscores BLO-SAM's strong ability to combat overfitting and improve generalization in few-shot settings. BLO-SAM's superiority can also be demonstrated by comparing the predicted segmentation masks, as shown in Appendix M.

Out-of-domain performance. We conducted experiments to evaluate BLO-SAM's adaptability to unseen tasks. We used BLO-SAM to train a model on the Kvasir dataset for the polyp segmentation task and subsequently evaluated its performance on the tasks of segmenting the human body, car body, lung, and teeth without further training, which have substantial domain differences compared to polyp segmentation. The same experimental protocol was applied to baseline methods as well for comparative analysis. The Dice scores (%) on test sets are presented in the Table 4. Our method substantially outperforms the baselines, which underscores our method's robustness and strong adaptability to unseen tasks. The inherent design of BLO-SAM, which mitigates overfitting by learning distinct parameter subsets on two disjoint data splits, significantly contributes to its improved out-of-domain generalization capabilities. Other results that show models' out-of-domain generalization performance can be found in Appendix E.

Computational Costs and Parameter Counts. We measured the training cost for all methods on an A100 GPU. We can see that BLO-SAM has comparable training time (Table 3) and inference time (Appendix D) to other SAM-based methods. More detailed analysis can be found in Appendix D. Furthermore, as shown in Table 3, BLO-SAM has minimal trainable parameters among all trainable methods, underscoring its parameter efficiency.

4.4. Ablation Studies

Ablation of Trainable Components of SAM. SAM comprises three primary components: image encoder, prompt encoder, and mask decoder. We have 4 ablation settings, including 1) finetuning the mask decoder (same as BLO-SAM), 2) the image encoder (via LoRA layers that are integrated into the inner Transformer blocks), 3) the prompt encoder (by updating the convolution layers), and 4) finetuning all the above components. This ablation experiment was

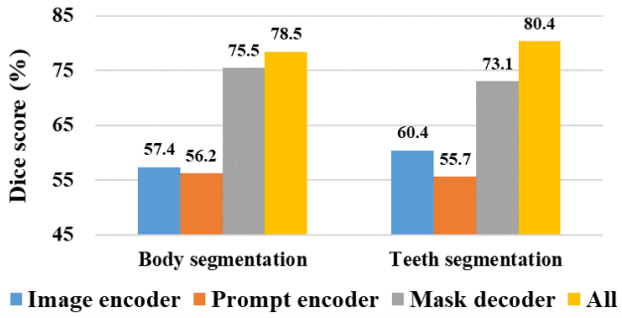

Figure 5. Ablation study of different trainable components. "All" represents finetuning all three components of SAM.

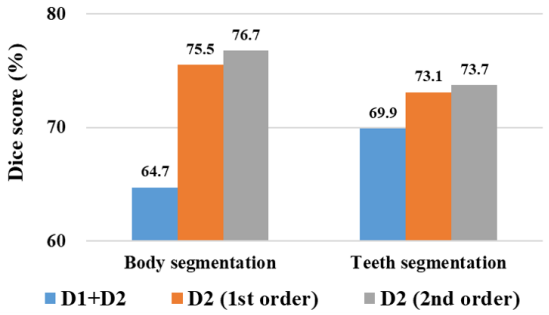

Figure 6. Ablation study on approaches for optimizing the prompt embedding.

conducted on body and teeth segmentation datasets, each with 4 labeled examples during finetuning. The results are shown in Fig. 5, where we have several key observations. Firstly, the mask decoder stands out as the most pivotal component for finetuning when adapting SAM to specific semantic segmentation tasks. Secondly, finetuning all three components of SAM yields improved performance, albeit at the expense of increased trainable parameters and training cost. For this reason, we only finetune the mask decoder in our BLO-SAM approach.

Ablation on Methods for Optimizing Prompt Embedding. In this study, we investigate the effectiveness of three approaches for optimizing prompt embedding. The experiments are performed on the body and teeth segmentation tasks with 4 labeled examples. The first approach optimizes both the prompt embedding A and model parameters W on the combination of D_1 and D_2. The second and third approaches optimize the prompt embedding on D_2 as in BLO-SAM but use first-order and second-order approximation, respectively, to compute the gradients in the BLO problem (see Appendix A for details). In the previously mentioned experiments, we use first-order approximation, which is computationally more efficient than the second-order approximation, to reduce computational cost.

Analysis of the results presented in Fig. 6 reveals that optimizing parameters A on sub-dataset D_2, while concurrently

Table 5. Ablation study on the influence of end-to-end mechanism comparing with performing two stages separately. "Eye-4" refers to the eye segmentation task with four training examples.

	Eye-4	Eye-8	Kvasir-4
Separate	58.3	64.6	52.9
End-to-end	**65.5**	**71.1**	**59.7**
	Kvasir-8	Human-4	Human-8
Separate	57.7	71.6	81.4
End-to-end	**61.6**	**76.3**	**85.1**

optimizing parameters W on sub-dataset D_1, yields superior performance compared to the simultaneous optimization of both parameter sets A and W on the combination of D_1 and D_2. This observation underscores the advantage of separately optimizing distinct parameter sets on disjoint sub-datasets, which is demonstrated to be effective in mitigating the risk of overfitting compared to optimizing all parameters on a single dataset. Moreover, the results show that second-order optimization holds the potential to further improve the final performance of the model, compared with the first-order method. However, it is crucial to acknowledge that these benefits come with computational challenges and necessitate increased computational resources.

Ablation of end-to-end optimization. We conducted an ablation study to verify the end-to-end tuning mechanism. Specifically, we compared our method, which performs the two stages end-to-end, against an ablation baseline that performs the two stages separately. In this baseline, the prompt embedding is initially set to random values, and the LoRA layers are trained using dataset D_1. Subsequently, with the LoRA layers fixed, the prompt embedding is trained on dataset D_2. As can be seen in Table 5, our approach outperforms this baseline. This underscores the advantages of end-to-end learning. In our method, the LoRA layers and prompt embedding are learned jointly so that they can mutually influence each other to achieve the overall best performance. Such a mechanism is missing in the baseline.

Ablation for the training set split strategy. We examined whether the performance of our method is significantly influenced by the strategies used to split the training data into two subsets, D_1 and D_2. We experimented with two strategies: in strategy 1, we randomly split the training data into two equal-sized subsets, while in strategy 2, we divide the data into a 3:1 ratio for D_1 and D_2. In strategy 2, we make D_1 larger than D_2 because the LoRA layers trained using D_1 have more parameters than the prompt embedding trained using D_2. To investigate whether the segmentation performance is significantly affected by the random seeds utilized to split data, we ran each splitting strategy three times with different random seeds.

Table 6. Ablation study on training set split strategy. "S1" and "S2" refer to the Strategy 1 and 2, respectively. "Teeth-4" refers to the teeth segmentation task with four training examples.

	Teeth-4	Wheel-4	Lung-4	Window-4
S1	73.0±0.3	79.6±0.3	93.8±0.4	68.7±0.5
S2	72.2±0.7	79.2±0.8	93.6±0.4	67.4±0.8

Table 7. Ablation study on only updating the model weights of LoRA layers (without updating prompt embedding). "Human-8" refers to the human body segmentation task with eight training examples.

	Nose-4	Mouth-4	Eye-4
No Prompt	59.6	57.9	46.2
BLO-SAM	**74.3**	**67.6**	**65.5**
	Brow-4	Kvasir-4	Human-8
No Prompt	30.0	52.4	77.5
BLO-SAM	**39.4**	**59.7**	**85.1**

Table 6 shows the average and standard deviation of the test Dice scores (%) across the three runs. The results show that Strategies 1 and 2 result in similar performance, which indicates that the performance of our method is stable under different split ratios. Furthermore, the small standard deviations indicate that our method's performance is relatively unaffected by the random seeds used for splitting data. These findings imply that extensive tuning to identify the optimal data splits can be avoided. Randomly splitting the training data into two equal-sized subsets is adequate for obtaining good segmentation performance.

Ablation on only updating LoRA layers. We performed this ablation study termed "No Prompt", where the prompt embedding was set to randomly sampled values, which is fixed during the training process, and only the LoRA layers were updated using D_1 and D_2. Table 7 presents the Dice scores on test sets. The performance under this ablation setting is inferior to our method. This underscores the importance of learning both the prompt embedding and LoRA layers, rather than solely focusing on the LoRA layers.

Appendix G presents three additional ablation studies.

5. Conclusion

In this paper, we propose BLO-SAM, a new approach for finetuning the Segment Anything Model (SAM) to perform downstream semantic segmentation tasks without requiring manual prompting. Leveraging a bi-level optimization strategy, we optimize the segmentation model parameters and prompt embedding on two different sub-datasets, mitigating the risk of overfitting and enhancing generalization. Experiments across diverse tasks with limited labeled training data strongly demonstrate the effectiveness of BLO-SAM.

Impact Statement

The paper that focuses on finetuning the Segment Anything Model (SAM) represents a significant stride in advancing semantic segmentation in few-shot settings, particularly in the context of both general and medical domains. Ethically, this endeavor raises questions about the responsible use of powerful AI technologies in healthcare, potentially improving diagnostic processes but also necessitating careful consideration of patient privacy and data security. The societal implications are broad, as the improved segmentation capabilities can benefit diverse industries, from autonomous vehicles to medical imaging. However, the responsible deployment of such advancements is crucial to avoid unintended consequences and biases. Striking a balance between innovation and ethical considerations will be pivotal in harnessing the full potential of the research for the betterment of society.

References

Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., Ring, R., Rutherford, E., Cabi, S., Han, T., Gong, Z., Samangooei, S., Monteiro, M., Menick, J., Borgeaud, S., Brock, A., Nematzadeh, A., Sharifzadeh, S., Binkowski, M., Barreira, R., Vinyals, O., Zisserman, A., and Simonyan, K. Flamingo: a visual language model for few-shot learning. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Cao, H., Wang, Y., Chen, J., Jiang, D., Zhang, X., Tian, Q., and Wang, M. Swin-unet: Unet-like pure transformer for medical image segmentation. In *European conference on computer vision*, pp. 205–218. Springer, 2022.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017a.

Chen, L.-C., Papandreou, G., Schroff, F., and Adam, H. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017b.

Cheng, H. K., Oh, S. W., Price, B., Schwing, A., and Lee, J.-Y. Tracking anything with decoupled video segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1316–1326, 2023.

Choe, S. K., Neiswanger, W., Xie, P., and Xing, E. Betty: An automatic differentiation library for multilevel optimization. *arXiv preprint arXiv:2207.02849*, 2022.

David, S. Semantics segmentation of car parts. Available: https://www.kaggle.com/datasets/intelecai/car-segmentation/data, 2020.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

Fan, Q., Pei, W., Tai, Y.-W., and Tang, C.-K. Self-support few-shot semantic segmentation. In *European Conference on Computer Vision*, pp. 701–719. Springer, 2022.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 1126–1135, 2017.

Franceschi, L., Frasconi, P., Salzo, S., Grazzi, R., and Pontil, M. Bilevel programming for hyperparameter optimization and meta-learning. In *International conference on machine learning*, pp. 1568–1577. PMLR, 2018.

Garg, B., Zhang, L., Sridhara, P., Hosseini, R., Xing, E., and Xie, P. Learning from mistakes–a framework for neural architecture search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 10184–10192, 2022.

Grazzi, R., Franceschi, L., Pontil, M., and Salzo, S. On the iteration complexity of hypergradient computation. In *International Conference on Machine Learning*, 2020.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

He, S., Bao, R., Li, J., Stout, J., Bjornerud, A., Grant, P. E., and Ou, Y. Computer-vision benchmark segment-anything model (sam) in medical images: Accuracy in 12 datasets. *arXiv preprint arXiv:2304.09324*, 2023.

Hosseini, R. and Xie, P. Image understanding by captioning with differentiable architecture search. In *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 4665–4673, 2022.

Hosseini, R., Zhang, L., Garg, B., and Xie, P. Fair and accurate decision making through group-aware learning. In *International Conference on Machine Learning*, pp. 13254–13269. PMLR, 2023.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pp. 2790–2799. PMLR, 2019.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Jha, D., Smedsrud, P. H., Riegler, M. A., Halvorsen, P., de Lange, T., Johansen, D., and Johansen, H. D. Kvasir-seg: A segmented polyp dataset. In *International Conference on Multimedia Modeling*, pp. 451–462. Springer, 2020.

Killamsetty, K., Li, C., Zhao, C., Chen, F., and Iyer, R. A nested bi-level optimization framework for robust few shot learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 7176–7184, 2022.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

Lai, X., Tian, Z., Chen, Y., Li, Y., Yuan, Y., Liu, S., and Jia, J. Lisa: Reasoning segmentation via large language model. *arXiv preprint arXiv:2308.00692*, 2023.

Lateef, F. and Ruichek, Y. Survey on semantic segmentation using deep learning techniques. *Neurocomputing*, 338: 321–348, 2019.

Lee, C.-H., Liu, Z., Wu, L., and Luo, P. Maskgan: Towards diverse and interactive facial image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5549–5558, 2020.

Lester, B., Al-Rfou, R., and Constant, N. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.243. URL https://aclanthology.org/2021.emnlp-main.243.

Liu, H., Simonyan, K., and Yang, Y. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018.

Liu, H., Tam, D., Muqeeth, M., Mohta, J., Huang, T., Bansal, M., and Raffel, C. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *arXiv preprint arXiv:2205.05638*, 2022.

Liu, H., Li, C., Wu, Q., and Lee, Y. J. Visual instruction tuning, 2023a.

Liu, X., Zheng, Y., Du, Z., Ding, M., Qian, Y., Yang, Z., and Tang, J. Gpt understands, too. *AI Open*, 2023b. ISSN 2666-6510. doi: https://doi.org/10.1016/j.aiopen.2023.08.012. URL https://www.sciencedirect.com/science/article/pii/S2666651023000141.

Long, J., Shelhamer, E., and Darrell, T. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Ma, J. and Wang, B. Segment anything in medical images. *arXiv preprint arXiv:2304.12306*, 2023.

Mazurowski, M. A., Dong, H., Gu, H., Yang, J., Konz, N., and Zhang, Y. Segment anything model for medical image analysis: an experimental study. *Medical Image Analysis*, 89:102918, 2023.

Mo, Y., Wu, Y., Yang, X., Liu, F., and Liao, Y. Review the state-of-the-art technologies of semantic segmentation based on deep learning. *Neurocomputing*, 493:626–646, 2022.

Moor, M., Banerjee, O., Abad, Z. S. H., Krumholz, H. M., Leskovec, J., Topol, E. J., and Rajpurkar, P. Foundation models for generalist medical artificial intelligence. *Nature*, 616(7956):259–265, 2023.

Qin, X., Song, X., and Jiang, S. Bi-level meta-learning for few-shot domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15900–15910, 2023.

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. Improving language understanding by generative pre-training. *OpenAI*, 2018.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Roman, K. Human segmentation dataset - tiktok dances. Available: www.kaggle.com/datasets, 2023.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pp. 234–241. Springer, 2015.

Shiraishi, J., Katsuragawa, S., Ikezoe, J., Matsumoto, T., Kobayashi, T., Komatsu, K.-i., Matsui, M., Fujita, H., Kodera, Y., and Doi, K. Development of a digital image database for chest radiographs with and without a lung nodule: receiver operating characteristic analysis of radiologists' detection of pulmonary nodules. *American Journal of Roentgenology*, 174(1):71–74, 2000.

Shu, J., Xie, Q., Yi, L., Zhao, Q., Zhou, S., Xu, Z., and Meng, D. Meta-weight-net: Learning an explicit mapping for sample weighting. In *NeurIPS*, 2019.

Sinha, A., Malo, P., and Deb, K. A review on bilevel optimization: From classical to evolutionary approaches and applications. *IEEE Transactions on Evolutionary Computation*, 22(2):276–295, 2017.

Tancik, M., Srinivasan, P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J., and Ng, R. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33: 7537–7547, 2020.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.

Touvron, H., Martin, L., Stone, K. R., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D. M., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A. S., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I. M., Korenev, A. V., Koura, P. S., Lachaux, M.-A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E. M., Subramanian, R., Tan, X., Tang, B., Taylor, R., Williams, A., Kuan, J. X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., and Scialom, T. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288, 2023b.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Wang, X., Girshick, R., Gupta, A., and He, K. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7794–7803, 2018.

Wang, Z., Yu, J., Yu, A. W., Dai, Z., Tsvetkov, Y., and Cao, Y. SimVLM: Simple visual language model pretraining with weak supervision. In *International Conference on Learning Representations*, 2022.

Wu, J., Fu, R., Fang, H., Liu, Y., Wang, Z., Xu, Y., Jin, Y., and Arbel, T. Medical sam adapter: Adapting segment anything model for medical image segmentation. *arXiv preprint arXiv:2304.12620*, 2023.

Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., and Luo, P. Segformer: Simple and efficient design for semantic segmentation with transformers. In *Neural Information Processing Systems (NeurIPS)*, 2021.

Yang, J., Gao, M., Li, Z., Gao, S., Wang, F., and Zheng, F. Track anything: Segment anything meets videos. *arXiv preprint arXiv:2304.11968*, 2023.

Zhang, K. and Liu, D. Customized segment anything model for medical image segmentation. *arXiv preprint arXiv:2304.13785*, 2023.

Zhang, Q., Chen, M., Bukharin, A., He, P., Cheng, Y., Chen, W., and Zhao, T. Adaptive budget allocation for parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.10512*, 2023a.

Zhang, W., Fu, C., Zheng, Y., Zhang, F., Zhao, Y., and Sham, C.-W. Hsnet: A hybrid semantic network for polyp segmentation. *Computers in biology and medicine*, 150:106173, 2022.

Zhang, Y., Ye, F., Chen, L., Xu, F., Chen, X., Wu, H., Cao, M., Li, Y., Wang, Y., and Huang, X. Children's dental panoramic radiographs dataset for caries segmentation and dental disease detection. *Scientific Data*, 10(1):380, 2023b.

A. Detailed Optimization Algorithm

In this section, we offer a detailed description of the optimization algorithm of BLO-SAM. We employ a gradient-based optimization algorithm to tackle the problem outlined in Eq. 4. Drawing inspiration from Liu et al. (2018), we approximately update $W^*(A)$ via one-step gradient descent in the lower level optimization. Then we plug the approximate $W^*(A)$ into the learning process of prompt embedding in the upper level and update A via one-step gradient descent. By using the one-step gradient descent updates for the bi-level optimization framework, we reduce the computational complexity. The detailed derivation of the update is as follows.

Lower-level. For the lower level, we perform a one-step gradient descent for Eq. (2) to approximate the optimal solution $W^*(A)$ on D_1. Specifically, at step t with an initial $W^{(t)}$ and a learning rate η_1, the updated $W^{(t+1)}$ is computed via gradient descent as follows:

$$W^{(t+1)} = W^{(t)} - \eta_1 \frac{d\mathcal{L}(W^{(t)}, A^{(t)}; D_1)}{dW^{(t)}} \tag{5}$$

Upper-level. Subsequently, we substitute $W^{(t+1)}$ as an approximation $W^*(A)$ into the upper-level optimization problem. Employing a similar one-step gradient descent, we approximate the optimal solution for A by minimizing the loss on D_2. At step t with an initial $A^{(t)}$ and a learning rate η_2, the updated $A^{(t+1)}$ is calculated as follows:

$$A^{(t+1)} = A^{(t)} - \eta_2 \frac{d\mathcal{L}(W^*(A), A^{(t)}; D_2)}{dA^{(t)}} \tag{6}$$

Inspired by Liu et al. (2018), we apply the unrolled model for the parameters of prompt embedding, A. In such a setting, the gradient w.r.t. A is:

$$\nabla_A \mathcal{L}_{D_2}(W^*(A), A) \approx \nabla_A \mathcal{L}_{D_2}(W - \xi \nabla_W \mathcal{L}_{D_1}(W, A), A) \tag{7}$$

where ξ is the learning rate of the lower-level optimization problem. Applying the chain rule to the approximate gradient yields:

$$\begin{aligned} &\nabla_A \mathcal{L}_{D_2}(W - \xi \nabla_W \mathcal{L}_{D_1}(W, A), A) \\ =&\nabla_A \mathcal{L}_{D_2}(W^*, A) - \xi \nabla^2_{A,W} \mathcal{L}_{D_1}(W, A) \cdot \nabla_{W^*} \mathcal{L}_{D_2}(W^*, A) \end{aligned} \tag{8}$$

for the second part of Eq. (8), we can apply the infinite difference approximation to simplify it to be:

$$\begin{aligned} &\nabla_A \mathcal{L}_{D_2}(W^*, A) - \xi \nabla^2_{A,W} \mathcal{L}_{D_1}(W, A) \cdot \nabla_{W^*} \mathcal{L}_{D_2}(W^*, A) \\ \approx&\frac{\nabla_A \mathcal{L}_{D_1}(W^+, A) - \nabla_A \mathcal{L}_{D_1}(W^-, A)}{2\epsilon} \end{aligned} \tag{9}$$

where, $W^{\pm} = W \pm \epsilon \nabla_{W^*} \mathcal{L}_{D_2}(W^*, A)$, and $\epsilon = 0.01/\|\nabla_{W^*} \mathcal{L}_{D_2}(W^*, A)\|_2$. If we set the ξ in Eq. (8) to be 0, we can get the first-order optimization for prompt embedding, otherwise we get the second-order optimization for the prompt embedding. For our main method, we utilize the first-order optimization method for its low computational cost. In the ablation studies in the main paper, we analyze the effect of using the second-order optimization method.

B. Preliminaries of Segment Anything Model (SAM)

The Segment Anything Model (SAM) (Kirillov et al., 2023) follows a comprehensive workflow, as shown in Fig. 7, beginning with the encoding of an input image and a prompt that indicates the object to segment. SAM comprises three key components: an image encoder that processes the input image and generates an image embedding, which captures the visual features of the input image; a prompt encoder that encodes the provided prompts, which captures the semantics of the object or region to segment; and a lightweight mask decoder that takes both image and prompt features as input and generates a segmentation mask for the object or region described in the prompt. The image encoder is a ViT (Dosovitskiy et al., 2020), which outputs a sequence of image tokens (vectors). For the prompts, SAM accommodates various types of prompts, such as points, bounding boxes, and coarse masks. Point or bounding box prompts are represented by positional encodings (Tancik et al., 2020). Mask prompts are encoded using a convolutional neural network. If no prompts are manually provided to SAM,

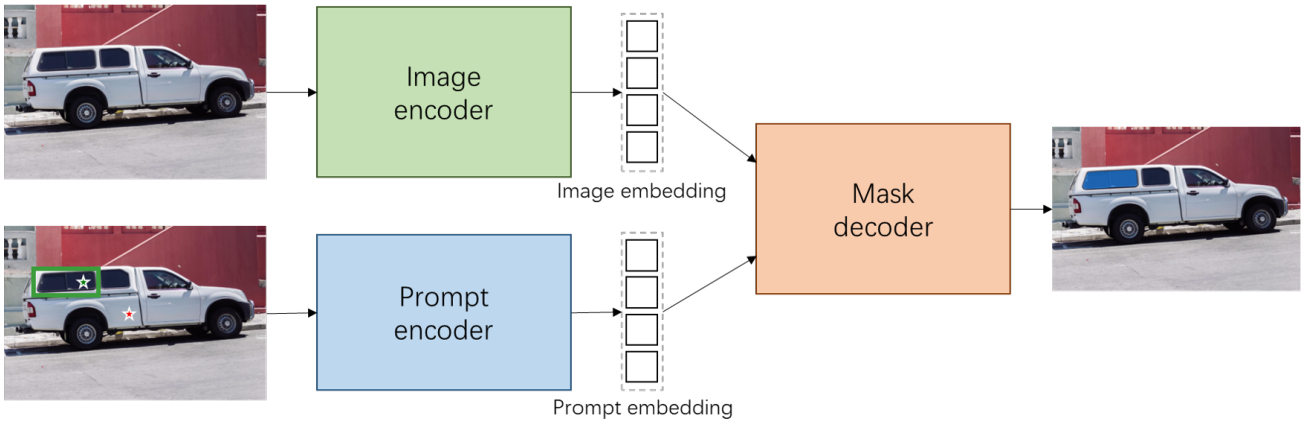

Figure 7. SAM overview. The image encoder takes an image as input to get the corresponding image embedding. The prompt encoder takes the input prompts, indicated as green point, red point and the rectangle shown in the image, to generate the prompt embedding. The mask decoder takes the image and prompts embedding as input to output the final masks.

Table 8. Number of test examples in different tasks.

Dataset	CelebAMask	Car	Body	Teeth	Kvasir	Lung
Test size	2000	100	2000	70	500	147

it dynamically uses a default embedding as the input prompt. The mask decoder, which is a modification of a Transformer decoder block (Vaswani et al., 2017), efficiently maps the image embedding, prompt embeddings, and an output token to a mask. In conclusion, SAM is a powerful and versatile tool for promptable segmentation, capable of handling a wide range of segmentation tasks efficiently and effectively. There are three model versions of the SAM with three different types of image encoders, scaling from ViT-B, ViT-L and ViT-H. ViT-B represents the encoder with the smallest model size. And, in our experiments, we use the ViT-B for computational efficiency by default.

C. Datasets

We evaluate our method on six tasks, including three tasks in the general domain, and three tasks in the medical domain. Correspondingly, we used a total of six datasets in our experiments, each of which was publicly available. The statistics of the test sets are listed in Table 8.

For the facial components segmentation task, we use the CelebAMask-HQ dataset [2], which is a large-scale face image dataset that has 30,000 high-resolution face images selected from the CelebA dataset by following CelebA-HQ. Each image has its segmentation mask of facial attributes corresponding to CelebA. We use the last 2,000 examples as the test set. For the car components segmentation task, we use the Car segmentation dataset [3], which contains 211 images of cars and their segmentation masks. The images exhibit a diverse range of car models and environmental contexts, ensuring that our segmentation method is tested across various visual scenarios. We split the last 100 examples as the test set. For the human body segmentation task, we use the Human Segmentation Dataset - TikTok Dances [4], which includes 2615 images of a segmented dancing people that are extracted from the videos from TikTok. We split the last 2000 examples as the test set, ensuring a robust evaluation of our method across a spectrum of dance scenarios.

For the teeth segmentation task, we use the Children's Dental Panoramic Radiographs Dataset [5], which is the world's first dataset of children's dental panoramic radiographs for caries segmentation and dental disease detection by segmenting and detecting annotations is published in this dataset. This dataset has already split its original examples into train and test sets, thus we just follow the original setting. For the gastrointestinal disease segmentation task, we use the Kvasir dataset [6], which contains 1000 polyp images and their corresponding ground truth. The diverse set of polyp images captures various

[2]https://drive.google.com/open?id=1badu11NqxGf6qM3PTTooQDJvQbejgbTv
[3]https://www.kaggle.com/datasets/intelecai/car-segmentation
[4]https://www.kaggle.com/datasets/tapakah68/segmentation-full-body-tiktok-dancing-dataset
[5]https://www.kaggle.com/datasets/truthisneverlinear/childrens-dental-panoramic-radiographs-dataset/data
[6]https://www.kaggle.com/datasets/abdallahwagih/kvasir-dataset-for-classification-and-segmentation

shapes, sizes, and locations within the gastrointestinal tract, contributing to the robustness of our segmentation algorithm. We split the last 500 examples as the test set. For lung segmentation, we utilize the JSRT (Japanese Society of Radiological Technology) database [7], containing 247 chest radiographs annotated with precise lung masks. This valuable dataset enhances the robustness of our segmentation method, capturing diverse clinical scenarios. The last 147 examples serve as our test set, ensuring a comprehensive evaluation of our algorithm's performance.

D. Training and Inference Cost

In Table 9, we show the training and inference cost comparison for all methods on the teeth segmentation task that are trained with 4 labeled examples. All the experiments are conducted on an A100 GPU with 80G memories. We use the GPU hours/seconds to measure the time cost for training and inference, respectively. From the results, we can see that BLO-SAM can achieve comparable performance in the training time, and BLO-SAM is among the smallest inference cost groups upon the inference time. Specifically, from the comparison of SAMed and BLO-SAM, We can see that the application of the bi-level optimization strategy will increase the training time slightly, but it will not influence the inference time at all. And, compared with other SAM-based methods (e.g. vanilla SAM and Med-SA), BLO-SAM also shows a superior inference time, because it doesn't need to input the prompts manually. Finally, compared with other baselines, including the supervised and few-shot methods, BLO-SAM shows slightly inferior performance in both training and inference costs, but BLO-SAM also shows a much better semantic segmentation performance.

Table 9. The training and inference time cost comparisons for the teeth segmentation task with 4 examples.

Method	DeepLab	SwinUnet	HSNet	SSP	SAM	Med-SA	SAMed	BLO-SAM
Training (GPU hours)	0.27	0.29	0.16	0.22	0	0.62	0.46	0.57
Inference (GPU seconds)	9.5	9.6	14.7	37.9	11.2	32.9	10.4	10.4

E. Out-of-domain performance

We conducted experiments in polyp and lung segmentation tasks to evaluate our method's out-of-domain generalization ability. For polyp segmentation, we trained a model using the KVasir dataset and then tested it on the CVC-ClinicDB polyp segmentation dataset without additional training. Similarly, in lung segmentation, we trained a model on the JSRT dataset and assessed it on the NLM-MC lung segmentation dataset, again without further training. This experimental protocol was applied to baseline methods as well for comparison. The Dice scores on test sets presented in Table 10 show that our method outperforms the baselines, underscoring its superior out-of-domain generalization capabilities. This enhanced robustness can be attributed to our method's design, which mitigates overfitting by learning two parameter subsets on two separate data subsets.

Table 10. The comparison of BLO-SAM with baselines for out-of-domain performance.

	Polyp segmentation	Lung segmentation
Train Set	Kvasir	JSRT
Test Set	CVC-ClinicDB	NLM-MC
SAM	20.8	36.7
SAMed	29.5	80.9
BLO-SAM	**37.0**	**85.6**

F. Detailed Results

We present a comprehensive analysis of all results, including mean and standard values, in Table 11, Table 12, and Table 13. A notable trend emerges as we observe the standard values: consistently, BLO-SAM outperforms the baselines by exhibiting smaller standard deviations across the majority of experiments. This trend indicates that BLO-SAM demonstrates enhanced stability compared to the baselines. The smaller standard deviations underscore the method's robustness and reliability,

[7] http://db.jsrt.or.jp/eng.php

Table 11. The comparison of BLO-SAM with baselines on the CelebAMask dataset evaluated by Dice Score (%), with different numbers of training examples. The overall performance is calculated by averaging the results for different facial components.

Method	Training with 4 labeled examples					
	Overall	Brow	Eye	Hair	Nose	Mouth
DeepLab	49.5±3.5	33.2±0.7	55.9±2.9	55.2±5.7	55.8±3.1	47.5±5.3
SwinUnet	35.2±1.8	21.7±1.8	21.7±2.5	46.7±1.6	44.4±2.0	41.6±0.9
HSNet	49.9±1.9	30.1±2.9	41.9±1.5	58.9±2.2	59.6±1.6	59.0±1.1
SSP	56.9±0.6	**40.3±0.6**	33.6±1.0	73.2±0.3	71.6±0.5	65.6±0.8
SAM	32.9±2.3	20.8±2.4	30.6±1.9	44.0±1.7	44.1±1.0	25.2±4.7
Med-SA	62.9±0.9	36.3±1.2	63.6±0.6	77.6±0.9	71.0±0.9	66.0±1.1
SAMed	58.2±1.5	28.3±1.4	55.9±2.7	78.5±1.3	65.1±1.8	63.0±0.5
BLO-SAM	**65.9±0.6**	39.4±0.9	**65.5±0.4**	**82.8±0.8**	**74.3±0.8**	**67.6±0.1**

Method	Training with 8 labeled examples					
	Overall	Brow	Eye	Hair	Nose	Mouth
DeepLab	54.9±1.4	37.3±0.7	59.7±1.9	58.0±2.3	64.0±1.1	55.7±0.8
SwinUnet	42.4±1.4	28.8±1.5	33.1±1.7	52.7±1.6	47.6±0.9	49.8±1.3
HSNet	60.1±1.9	43.6±2.3	58.1±0.7	76.6±2.9	58.2±2.9	64.2±0.8
SSP	60.0±0.9	45.4±0.7	31.2±0.2	76.6±1.4	74.0±1.2	72.7±0.9
SAM	32.9±2.1	20.8±2.4	30.6±0.9	44.0±1.7	44.1±1.0	25.2±4.7
Med-SA	67.7±0.9	42.9±0.7	65.5±1.5	82.4±0.6	74.6±0.8	73.1±0.7
SAMed	65.0±1.0	39.5±1.4	66.0±0.6	82.4±0.6	70.5±1.2	66.4±1.2
BLO-SAM	**69.9±0.7**	**45.8±0.6**	**71.1±1.2**	**83.6±0.5**	**76.1±1.3**	**72.7±0.1**

showcasing its ability to consistently yield precise results across diverse segmentation tasks.

G. Other Ablations

In this section, we show some results for other ablation studies, without other declarations, all the ablation studies are conducted on the body and teeth segmentation tasks with 4 labeled examples in default.

Sensitivity Analysis of the Trade-off Parameter λ. In this experiment, we investigate how different settings of the trade-off parameter (λ) in Eq. 1 affect the model's final performance. Analyzing the results presented in Fig. 8, it is evident that setting λ to a value in the middle ground yields the optimal performance for both tasks. It is important to balance the two losses as the cross-entropy loss primarily concentrates on the classification for each pixel independently, making it susceptible to overfitting in the presence of class imbalance, where images may consist of a substantial number of background pixels. In contrast, the Dice loss mitigates this issue by prioritizing spatial overlap. These results show the importance of striking a balance between pixel-level classification accuracy and spatial overlap. The default value of 0.8 suggested by (Zhang & Liu, 2023) is a good choice.

Ablation of the Setting the Rank of the LoRA Layers. Analyzing the results in Fig. 9, it becomes apparent that, for body and teeth segmentation tasks, setting the number of ranks to 4 yields the best generalization on test examples. The choice of the number of ranks is primarily determined by the complexity of specific tasks; increasing the rank excessively can lead to a more complex model that overfits a small training set and fails to generalize to unseen test examples.

Comparison with Full Finetuning. In this experiment, we compare our method with full finetuning, denoting as FT-SAM, in which we set all the parameters of vanilla SAM to be trainable and tuned them without any modification. As shown in Table 10, FT-SAM suffers severe overfitting on the body segmentation task, in which FT-SAM only achieves the dice score of 47.5%, while BLO-SAM achieves the dice score of 75.5%. This is mainly because, with very limited training examples, full finetuning is easy to overfit to the training examples, causing a very poor generalization on the test examples. And, for the teeth segmentation task, BLO-SAM can also achieve comparable performance with the FT-SAM on the test set.

Table 12. The comparison of BLO-SAM with baselines on the car components (left of the vertical bar) and human body (right of the vertical bar) datasets evaluated by Dice Score (%) with different numbers of training examples. The overall performance is calculated by averaging the results for different components.

Method	Car components								Human body	
	Training with 2 labeled examples				Training with 4 labeled examples				4 examples	8 examples
	Overall	Body	Wheel	Window	Overall	Body	Wheel	Window		
DeepLab	46.5±4.6	58.5±5.6	55.0±3.2	26.1±5.0	59.8±1.5	62.9±1.4	66.6±0.8	49.8±2.2	31.8±2.3	37.0±0.8
SwinUnet	31.4±6.8	22.2±6.5	33.4±5.2	38.7±8.7	47.3±3.1	49.3±5.6	53.7±1.7	38.9±2.0	30.9±3.6	56.8±0.5
HSNet	51.0±1.5	67.1±2.2	32.4±1.0	53.4±1.2	68.8±0.9	70.8±0.8	70.3±0.8	65.4±1.0	50.6±3.1	55.8±2.4
SSP	64.1±1.0	62.8±1.1	**76.2±1.1**	53.3±0.9	72.2±0.9	77.7±0.3	78.2±0.5	60.6±1.8	58.9±0.1	76.5±0.1
SAM	35.1±1.6	40.8±2.5	41.7±1.1	22.9±1.2	35.1±1.6	40.8±2.5	41.7±1.1	22.9±1.2	26.0±1.3	26.0±1.3
Med-SA	67.3±2.1	80.8±1.5	70.9±0.5	50.3±4.2	75.9±1.3	84.4±1.0	78.1±0.3	65.3±2.5	58.8±0.7	80.6±0.4
SAMed	60.4±1.5	78.8±1.4	65.0±1.6	37.5±1.6	74.0±1.4	85.3±0.6	72.5±0.6	64.2±3.0	63.8±1.3	81.5±1.2
BLO-SAM	**71.1±0.7**	**83.2±0.6**	74.5±0.5	**55.6±1.0**	**78.3±0.5**	**86.3±0.1**	**79.9±0.9**	**68.6±0.4**	**76.3±0.8**	**85.1±0.5**

Table 13. The comparison of BLO-SAM with baselines on the teeth, gastrointestinal disease and lung datasets evaluated by Dice Score (%) with different numbers of training examples.

Method	Teeth		Kvasir		Lung	
	4 examples	8 examples	4 examples	8 examples	2 examples	4 examples
DeepLab	56.8±0.3	63.9±1.0	31.9±0.7	37.8±0.3	61.1±0.3	76.4±0.5
SwinUnet	28.6±1.6	50.6±1.1	37.8±0.3	39.0±0.8	62.1±0.1	76.4±0.3
HSNet	70.2±1.3	72.2±0.4	30.0±0.7	35.1±0.8	83.1±0.5	84.5±0.4
SSP	33.7±0.5	51.7±1.2	27.4±0.3	27.7±0.4	83.2±0.3	90.1±0.6
SAM	21.2±1.6	21.2±1.6	14.7±3.8	14.7±3.8	31.4±0.1	31.4±0.1
Med-SA	69.7±0.3	76.2±0.4	33.5±3.5	59.3±0.4	82.9±0.5	91.7±0.4
SAMed	69.8±0.5	75.1±2.2	42.7±0.5	57.1±0.7	84.4±0.5	91.8±0.4
BLO-SAM	**73.2±0.7**	**77.3±0.2**	**59.7±0.2**	**61.6±0.3**	**87.1±0.3**	**93.7±0.1**

H. Comparison with LISA

We conducted experiments to compare our method with LISA. To utilize LISA (Lai et al., 2023) for segmenting teeth, polyps (on the Kvasir dataset), and car wheels, we employed specific prompts for each task, including "please segment the dental caries in the input image", "please segment the polyp area in the input image", and "please segment the car wheels in the input image". Table 14 shows the Dice scores on test sets. The term "Teeth-4" refers to the teeth segmentation task with four training examples. Our method outperforms LISA substantially, which is attributed to our method's capability to prevent overfitting by learning different parameter subsets on different data subsets.

I. Train with More Examples

We conduct experiments to explore what would occur when increasing the number of training examples on the body segmentation task. We increased the number of training sets to 128 and 512, and conducted experiments for Med-SA, SAMed, and BLO-SAM. As shown in Table 11, After increasing the number of training examples, the performance of BLO-SAM can be further improved, as it achieves the highest dice score of 92.8% among all experiments when training with 512 labeled examples. What's more, when faced with very limited training examples, BLO-SAM also shows a strong capability to overcome the risk of overfitting, which can be demonstrated by the biggest performance gap with Med-SA and SAMed when training with only 4 labeled examples.

Table 14. The comparison of BLO-SAM with LISA

	Teeth-4	Kvasir-4	Eye-4	Lung-4	Mouth-4
LISA	47.8	33.9	54.7	76.6	57.3
BLO-SAM	**73.2**	**59.7**	**65.5**	**93.7**	**67.6**

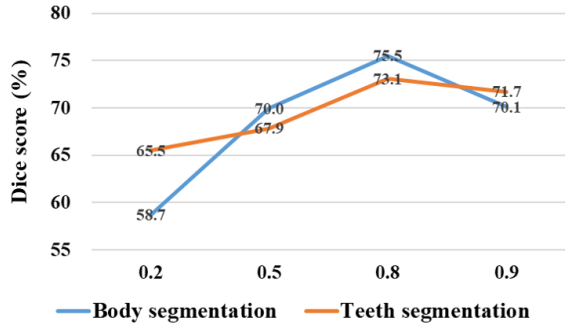

Figure 8. Ablation study of optimization methods for prompt embedding.

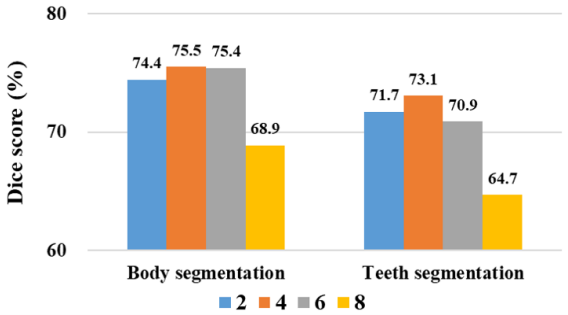

Figure 9. Ablation study of the setting of the number of ranks for added LoRA layers.

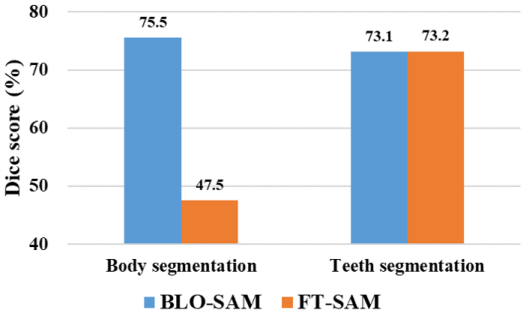

Figure 10. Ablation study of comparing with the full fine-tuning. "FT-SAM" denotes the full finetuning of SAM.

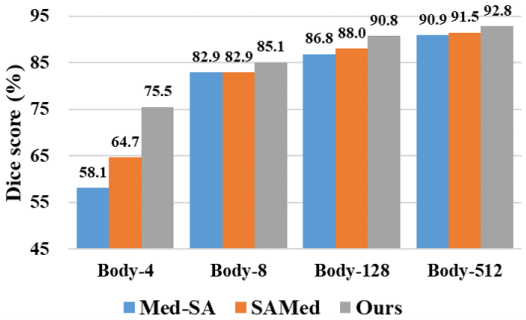

Figure 11. Ablation study of training with different number of examples on body segmentation task.

J. Comparison with Auto-SAM.

We experimented with an additional baseline, Auto-SAM. In this baseline, given an input image, we first use scatter dots in a mesh format to prompt SAM to generate segments for each image. Each segment is enclosed within the smallest possible rectangular image region, which is then fed into a trained CNN to classify whether the segment belongs to the target object or background. Segments classified as the target object are combined to form the final segmentation output. To train the CNN, segments in the training set are labeled based on whether at least 50% of their pixels match the target object in the ground-truth segmentation map. Table 15 shows Dice scores on test sets. Our approach surpasses the baseline in performance. In the baseline method, SAM initially produces preliminary segmented patches, which are subsequently classified to derive the final segmentation mask, with these two processes occurring separately. If the initial segmented patches are inaccurate, such as merging parts of the foreground objects with the background in one segment, these errors cannot be rectified during the classification phase. Conversely, our method streamlines the process by directly creating the final segmentation mask from the input, thus circumventing the intermediate errors inherent in the baseline approach.

K. Results on other benchmarks.

We conducted additional experiments using datasets from Med-SA and SAMed, Specifically, we utilized the ISIC2018 dataset for skin lesion segmentation and the Synapse dataset for segmentation of the gallbladder, pancreas, liver, and kidney. The Dice scores (%) are presented in Table 16. The term "Skin-4" refers to the skin lesion segmentation task with four training examples. Our method surpasses both Med-SA and SAMed, further underscoring its superiority compared to baseline approaches.

Table 15. The comparison of BLO-SAM with Auto-SAM

	Teeth-4	Human-4	Kvasir-4	Eye-4
Auto-SAM	40.6	48.7	27.9	40.1
BLO-SAM	**73.2**	**76.3**	**59.7**	**65.5**

Table 16. The comparison of BLO-SAM with baselines on the other two benchmarks.

	Skin-4	Gallbladder-8	Pancreas-8	Gallbladder-10	Liver-20	Kidney-20
Med-SA	58.2	23.5	10.8	32.8	60.6	40.5
SAMed	62.5	27.2	23.9	32.1	60.8	39.8
BLO-SAM	**66.7**	**34.3**	**29.8**	**37.1**	**65.2**	**44.3**

L. Comparison against the standard UNet.

We compared our method with UNet for the tasks of the human body and teeth segmentation. Tabel 17 shows the Dice scores on the test sets. The term ”Human-4” refers to the human body segmentation task with four training examples. Our method outperforms UNet by a large margin. This superiority is attributed to our method’s ability to transfer the capacity of pretrained SAM to downstream tasks via finetuning. Unlike SAM, UNet was not pretrained on large-scale data, making it prone to overfitting when the number of training examples is very limited, such as 4 and 8.

M. Qualitative Results

In Figures 12, 13, 14, 15, and 16, we present the segmentation masks generated by different methods. A noteworthy observation is that, across various tasks, the segmentation masks predicted by BLO-SAM consistently exhibit a superior level of detail, demonstrating finer prediction of target components with minimal interference from the background. This is particularly evident when compared to the segmentation masks produced by alternative baselines. The qualitative results underscore the efficacy of BLO-SAM in capturing intricate features while maintaining a cleaner distinction between the foreground and background, emphasizing its robust performance across diverse segmentation tasks.

Table 17. The comparison of BLO-SAM with UNet

	Human-4	Human-8	Teeth-4	Teeth-8
UNet	33.5	37.1	55.6	64.3
BLO-SAM	**76.3**	**85.1**	**73.2**	**77.3**

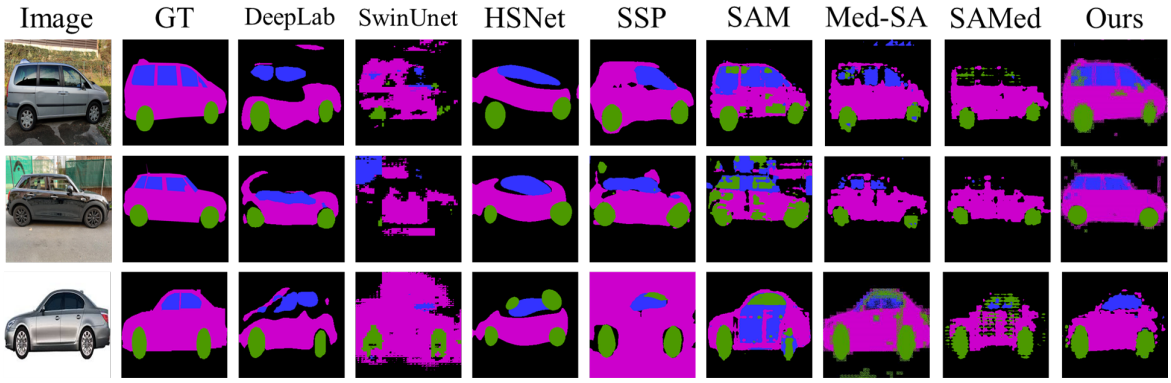

Figure 12. Qualitative results on some randomly sampled test examples from the car dataset.

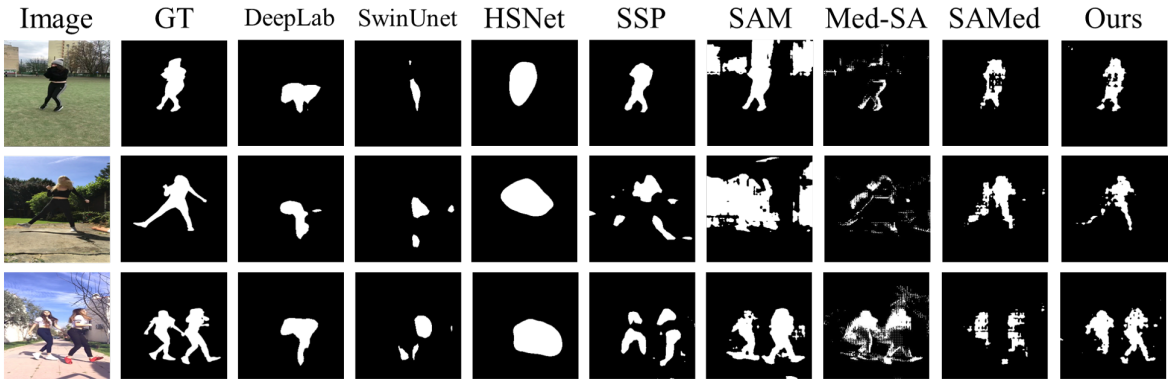

Figure 13. Qualitative results on some randomly sampled test examples from the body dataset.

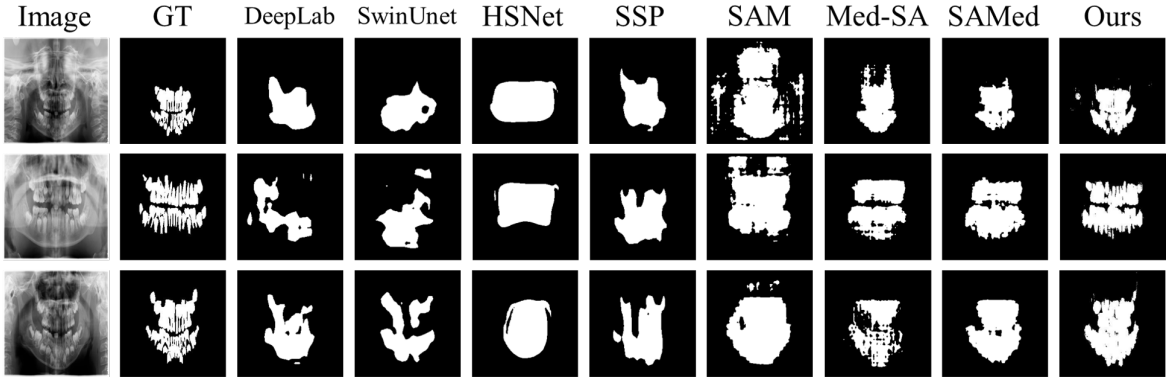

Figure 14. Qualitative results on some randomly sampled test examples from the teeth dataset.

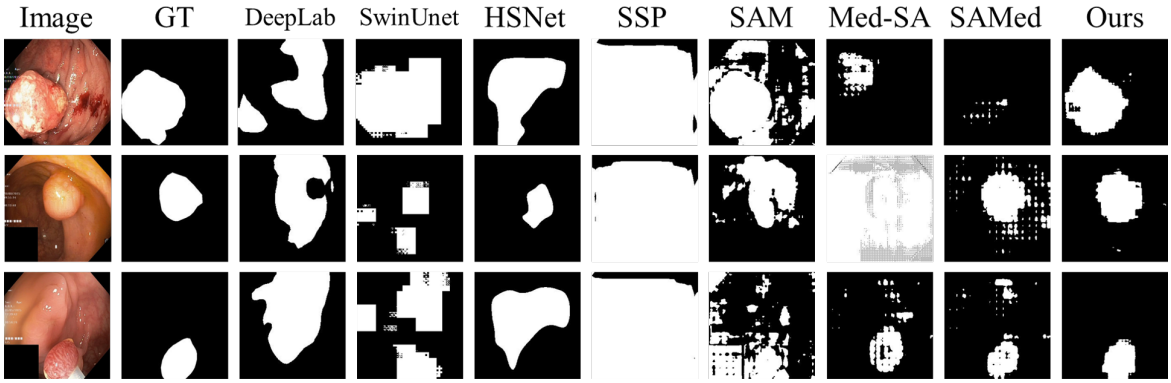

Figure 15. Qualitative results on some randomly sampled test examples from the gastrointestinal disease dataset.

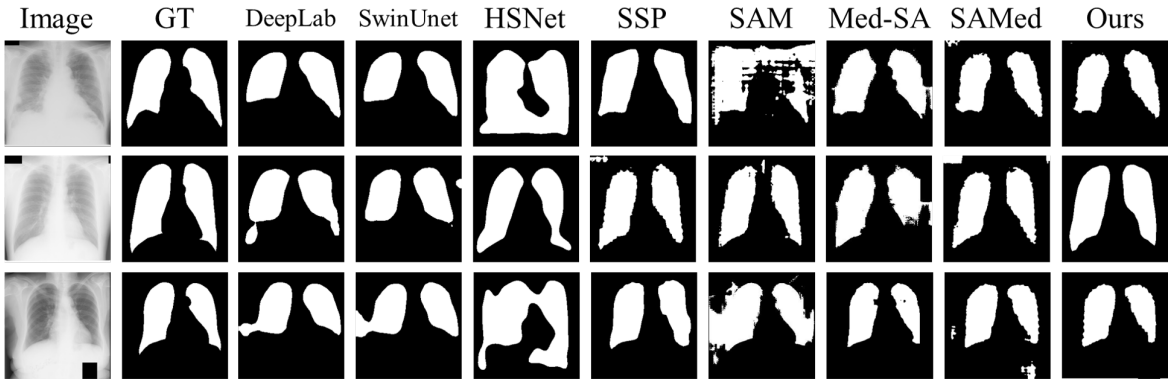

Figure 16. Qualitative results on some randomly sampled test examples from the lung dataset.