8-2024

A Mixture-of-Experts Approach to Fine-Tuning the Segment Anything Model

Rajat Sahay
rs6287@rit.edu

A Mixture-of-Experts Approach to Fine-Tuning the Segment Anything Model

Rajat Sahay

A Mixture-of-Experts Approach to Fine-Tuning the Segment Anything Model

by

Rajat Sahay

August 2024

A Thesis submitted to the
B. Thomas Golisano College of Computing and Information Sciences
Department of Software Engineering
in partial fulfillment of the requirements for the
Master of Science Degree in **Data Science**
at the Rochester Institute of Technology

A Mixture-of-Experts Approach to Fine-Tuning the Segment Anything Model

Rajat Sahay

Committee Approval:

Dr. Andreas Savakis *Advisor* Date
Department of Computer Engineering

Dr. Travis Desell Date
Department of Software Engineering

Dr. Qi Yu Date
School of Information

Abstract

The emergence of large, general-purpose foundation models has sparked significant interest in the broader machine learning community. Among the many models being released, the Segment Anything Model (SAM) has demonstrated exceptional capabilities for object segmentation in various settings. Given its expansive training data, SAM has been used for image segmentation across various downstream tasks ranging from tumour segmentation to aerial object detection. However, the majority of pretraining data used by SAM consists of naturally-occurring images, which have significantly different characteristics from images of tumours or images taken by drones. In order to align SAM to these previously unseen domains, we need to fine-tune the model to leverage its prior knowledge and learn generalizable features from newer images to increase performance on a given target dataset. However, given the extremely large size and number of parameters, traditional fine-tuning methods are too costly to be applied to foundation models such as SAM. To overcome this limitation, a new family of methods known as Parameter-Efficient Fine-Tuning (PEFT) techniques have emerged to effectively and efficiently tailor these large models to application domains outside their training data. While there has been considerable research on developing new PEFT techniques, different methods modify the representation of a model differently, making it a non-trivial task to select the most appropriate method for a particular domain of interest. To this end, we propose a new framework, Mixture-of-PEFTs (MoPEFT), that is inspired by tradition Mixture-of-Experts (MoE) methodologies and use it to fine-tune SAM. Our MoPEFT framework incorporates three different PEFT techniques as submodules and learns to dynamically activate the ones that are best suited for a given data-task setup. We test our method on the Segment Anything Model across 22 different datasets spread over 5 domains and show that MoPEFT consistently outperforms other fine-tuning methods on the MESS benchmark.

Acknowledgments

I would like to begin by sincerely thanking my advisor Dr. Andreas Savakis for all his support and mentorship over the past two years. I would also like to my colleagues at the Vision and Image Processing Lab- Georgi, Rahi, Navin, and Alec. This research would not have been possible without their help. I am also grateful to Dr. Travis Desell and Dr. Qi Yu for agreeing to serve on my thesis defense committee. Finally, I want to thank my family and friends for their constant and continued support throughout this entire experience.

To my parents, who taught me how to adapt efficiently to new circumstances.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

The machine learning research community has witnessed an explosion in the development of foundation models for language and vision, such as CLIP [34], GPT-4 [1], PaLM [4] and the Segment Anything Model (SAM) [14]. SAM is a promptable model for image segmentation that is pretrained on over 1 billion masks and 11 million images. It has demonstrated performance comparable to state-of-the-art approaches in multiple applications related to segmentation tasks. Moreover, SAM's zero-shot and few-shot capabilities have garnered significant attention across multiple domains [13, 52]. However, prior works [25, 29] have shown that despite noteworthy proficiency in segmenting real-world objects in natural images, SAM has difficulty with objects outside its training domain.

Following the pretraining-fine-tuning paradigm [38, 50], it is desirable to fine-tune SAM in order to enhance its performance in the application domain of interest. However, fine-tuning foundation models can be costly due to their large number of parameters. This motivates the development of efficient fine-tuning methods with the goal of achieving comparable performance to full fine-tuning while employing as few trainable parameters as possible. Interest in Parameter-Efficient Fine-Tuning methods (PEFT) has increased significantly since the advent of foundation models [8, 12, 15, 32].

Recent studies [7, 16] have shown that some PEFT methods are more effective at fine-tuning with the objective of reducing overfitting on the target domain, especially in data-sparse environments. However, we find that combining different PEFT methods often yields better results without a substantial loss in efficiency. This is because different techniques operate on different

parts of the transformer architecture, making it possible to utilize more than one technique at a time. We also run additional experiments on object detection tasks, and find a similar, recurring problem. More details can be found in Appendix A.

In light of this, we propose a new framework, called Mixture-of-PEFTs (MoPEFT), that incorporates different PEFT methods as submodules and learns to dynamically activate the fine-tuning method(s) that best suit the data or task of interest. Inspired by the Mixture-of-Experts approach [18, 28, 33], MoPEFT switches between different PEFT methods using a *gating mechanism* that learns to favor the method that positively contributes to a given task. In addition, since the number of parameters introduced by each PEFT is very small, compared to the entire SAM architecture, combining multiple PEFT methods has little effect on the efficiency of our framework. In this thesis, we consider the three most commonly used PEFT techniques- Low Rank Adaptation (LoRA) [8], Prefix Tuning [12], and Adapters [7]. Our experiments shed light on the effectiveness of these methods across segmentation tasks in multiple domains, and demonstrate gains when combined together in our MoPEFT framework.

1.2 Thesis Contributions

The contributions for this thesis can be outlined as follows:

- We conduct a comprehensive survey of the widely-used PEFT methods and benchmark their performance across segmentation tasks in multiple domains.
- We introduce our MoPEFT framework, which incorporates multiple PEFT methods as submodules and learns to dynamically activate or deactivate the appropriate submodule based on the given task.
- We show that our MoPEFT framework achieves better performance than individual PEFT methods across multiple domains in the MESS benchmark.

1.3 Motivation

Image segmentation is a fundamental task in computer vision and plays an important part in a significant number of applications across varied domains ranging form tumour segmentation in medical images to object identification

for autonomous driving. However, given the breadth of domains and the amount of differences (both coarse and fine-grained) in their constituent images, it is necessary to have a model that can adapt to these changes without a reduction in performance. While there have been recent advances in creating large, 'foundation' models which specialize in a singular domain, it is extremely tough and costly to adapt these models to newer domains. This led to a new area of fine-tuning, known as Parameter-Efficient Fine-Tuning (PEFT) which uses specific techniques that effectively adapt these large models to unseen domains in a cost-efficient way. However, there has been significant debate on *which* PEFT method to use for a particular domain, since different methods show varying performance and there is no one-size-fits-all technique that can be applied in any given scenario. This thesis addresses this issue by taking the three broad categories of PEFT methods and combining them into one dynamic framework. Our proposed MoPEFT method leverages the best parts of all constituent PEFT techniques and surpasses their individual performance.

1.4 Document Structure

The rest of the document is structured as follows: Chapter 2 discusses the background material and contains an overview of related works in fine-tuning and foundation models. Chapter 3 covers our proposed methodology and provides the fundamental intuitions behind the implementation. Chapter 4 presents an overview of the MESS benchmark, which is the collection of datasets spread across different domains that we use as our test-bed, and provides practical details on implementation. It also presents our results compared to current state-of-the-art mechanisms, and also discusses different ablation studies that we ran. Finally, Chapter 5 contains our concluding remarks and provides possibilities for future works.

Chapter 2

Background

2.1 Segment Anything Model

We begin by describing the architecture and workings of the SAM in more detail. These provide a better insight into the intricacies of the model, which set the stage for our parameter-efficient fine-tuning techniques. The Segment Anything Model [14] is a large-scale segmentation model released by Meta in April 2023. It deals with the concept of *promptable* segmentation, which translates the idea of prompting from NLP to panoptic segmentation. In the case of SAM, a prompt can be a set of foreground/background points, a rough box or mask, free-form text, or, in general, any information specifying a region-of-interest within an image. The goal of SAM is then to return a valid segmentation mask given any prompt. The requirement of a valid mask refers to the fact that even when the prompt is ambiguous or referring to multiple objects, the generated mask must reasonably segment at least one of those objects. The introduction of this promptable segmentation task allows it to be used as both a pretraining objective and to solve general downstream segmentation tasks using prompt engineering.

SAM consists of three main components: a large-scale image encoder, a prompt encoder, and a lightweight mask decoder. The image encoder utilizes a pretrained Vision Transformer (ViT) to process high-resolution inputs and produces feature maps at a 1/16 scale of the original image. The prompt encoder enables region-of-interest selection using sparse prompts (points, bounding boxes), dense prompts (masks) or text prompts. Both encoders feed into the mask decoder, which updates the image and prompt embeddings through a cross-attention mechanism. Following the standard implementation of SAM, for the purposes of our experiments, we rescaled all our inputs to 1024×1024.

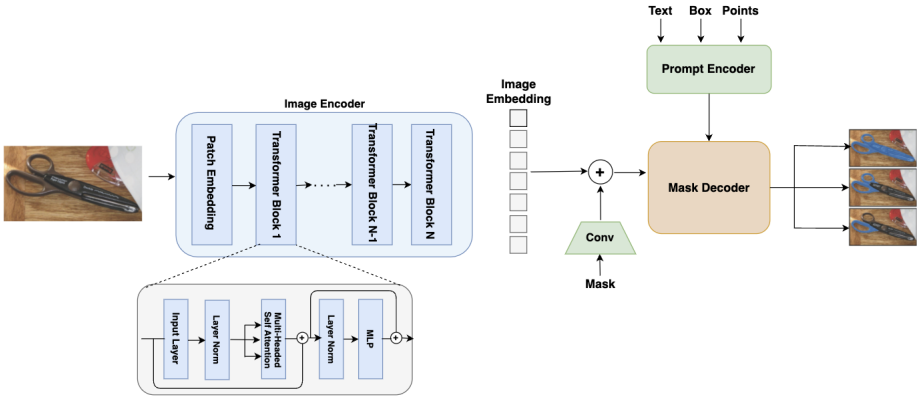

Figure 2.1: The Segment Anything Model [14].

Since the image encoder outputs a 16× downscaled embedding, we obtain a 64×64 vector which is passed through a 1×1 and a 3×3 convolutional layer to reduce the channel dimension. Every convolutional layer is followed by a `LayerNorm` layer to enable layer normalization.

While multiple variants of SAM were released based on different sizes of the image encoder, our experiments in this study are based on the ViT-B/16 version, keeping in mind the significant computational costs incurred during fine-tuning.

2.2 Parameter Efficient Fine-Tuning

Parameter-Efficient Fine-Tuning (PEFT) methods aim to minimize the computational and memory requirements incurred during fine-tuning by selectively updating only a small subset of model parameters, while keeping the majority frozen. PEFT techniques originally gained popularity in the Natural Language Processing (NLP) domain after being used to align large language models (LLMs) to specific domains [26, 48]. These techniques have also proven useful in the visual domain [37], and have been adapted to enhance the performance of Vision Transformers under domain shifts [3, 12].

PEFT encompasses methods such as adapter-based techniques [7], prompt-driven fine-tuning [15, 16], and low-rank adaptation (LoRA) [8]. In our work, we focus on dynamically incorporating all three PEFT techniques into SAM to improve its performance on semantic segmentation tasks. We explain all three techniques in greater detail in Section 3.2.

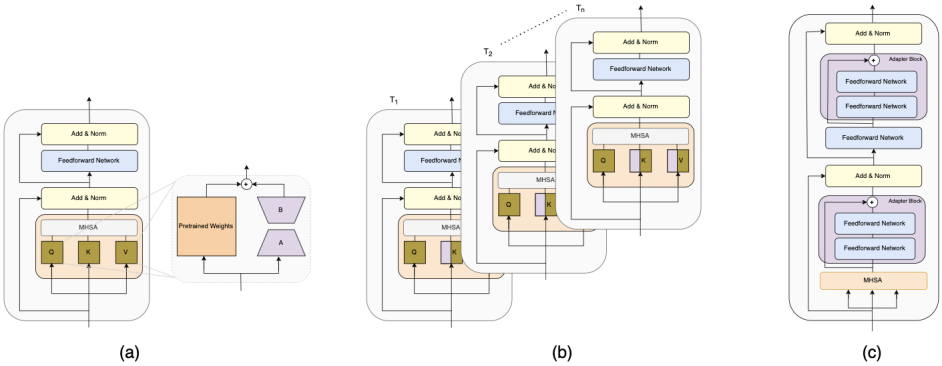

Figure 2.2: An overview of the PEFT methods used in our MoPEFT framework: (a) Low Rank Adaptation (LoRA) [8], (b) Prefix Tuning [16], and (c) Bottleneck Adapters [7].

2.3 Mixture-of-Experts

Mixture-of-Experts (MoE) is designed to expand model capacity while introducing minimal overhead during training and inference. A traditional MoE architecture [18] maintains a set of experts (neural networks) and one or more trainable gates that select a combination of experts specific to each given input. Despite being conceptually similar, this work does not aim to improve traditional MoE architectures. We only mirror the original goal of MoE in our work, expanding model capacity without excessively increasing computational overhead. A major difference between MoPEFT and MoE is that the submodules in MoPEFT are not combined explicitly by summation like conventional MoE, but in sequential order. This causes them to affect each other implicitly.

Chapter 3

Mixture-of-PEFTs (MoPEFT)

We now present the details of our proposed MoPEFT framework. We set up a foundation for the framework by defining the task at hand, giving a deeper overview into how each of the PEFT techniques work, and outlining how they are integrated into our unified framework.

3.1 Task Formulation

We consider a very large model M, which cannot be efficiently fine-tuned due to computational costs, and assume we have a collection of PEFT methods $FT = \{ft_1, ft_2...ft_n\}$ with negligible trainable parameters compared to M, i.e. $\sum_{i=1}^{n} |ft_i| << |M|$. Here, $|\circ|$ denotes the number of trainable parameters for a given model or fine-tuning technique. Our goal is to design a framework that incorporates $\{ft_1, ft_2...ft_n\}$ as individual, independent submodules and learns to dynamically activate different ft_i based on different data-task scenarios. This would ensure that a singular framework would be capable of achieving optimal results in terms of both accuracy and efficiency without permuting through all data-task combinations for every datapoint.

3.2 Parameter-Efficient Fine-Tuning Methods

In this section, we provide a brief overview of the PEFT techniques used in our framework and how they align the image encoder to the target dataset. A pictorial representation of each of the PEFT techniques can be found in

Figure 2.2.

3.2.1 Low Rank Adaptation (LoRA)

Low Rank Adaptation (LoRA) [8] exploits the *low rank* structure inherent in deep learning models to align them to specific tasks. With LoRA, we adapt SAM by updating the parameterized weight matrices of the multi-head self-attention (MHSA) mechanism within each transformer block in the image encoder. The pretrained weight matrix $W_0 \in \mathbb{R}^{d\times k}$ is updated as $W_0 + \Delta W$, where $\Delta W \in \mathbb{R}^{d\times k}$ is a low-rank matrix decomposed as $\Delta W = BA$. Here, $B \in \mathbb{R}^{d\times r}, A \in \mathbb{R}^{r\times k}$ and the rank $r << min(d, k)$. During fine-tuning, the pre-trained weights remain frozen, and ΔW serves as the trainable parameter. The decomposition of $\Delta W = BA$ as a product of two low-rank matrices effectively reduces the memory and computational cost of fine-tuning.

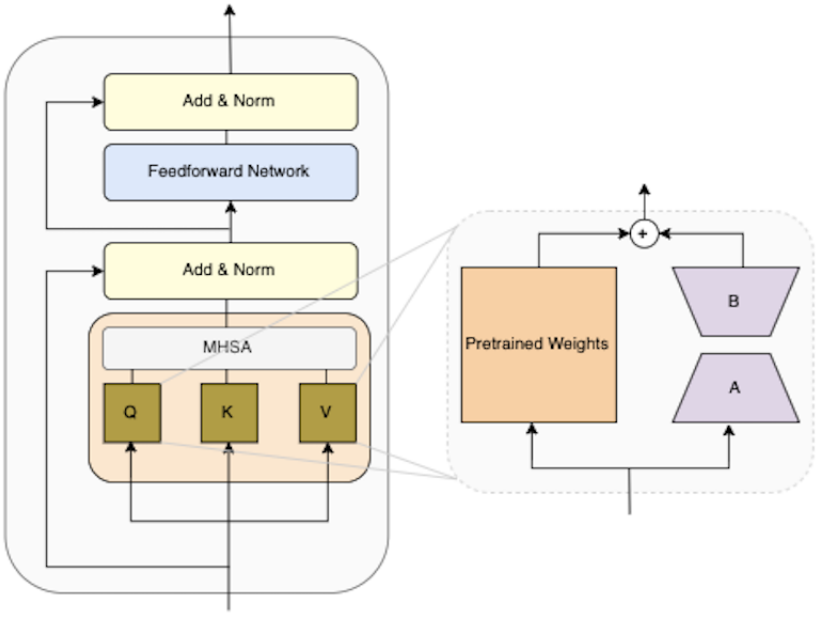

Figure 3.1: A pictorial representation of fine-tuning using Low Rank Adaptation (LoRA) [8].

For a SAM image encoder (a ViT-B/16 model) represented as E_i containing l transformer blocks, we can define the LoRA process on each block as follows.

$$h = W_0 x + \Delta W x = W_0 x + B\Lambda A \tag{3.1}$$

where W_0 represents the pretrained weight matrix, x denotes the input for each transformer block, and h represents the output. ΔW represents the parameterized weight matrix with B and A representing the left and right singular values of ΔW, while the diagonal matrix Λ contains singular values $\lambda_{i(1\leq i\leq r)}$. This restructures the attention process as follows

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{3.2}$$

where (ignoring $\sqrt{d_k}$ as a scaling parameter)

$$\begin{aligned} Q &= (W_q + A_q\Lambda_q B_q)x \\ K &= W_k x \\ V &= (W_v + A_v\Lambda_v B_v)x \end{aligned} \tag{3.3}$$

We preserve the original projection weight matrices W_q, W_k, and W_v as frozen while A_q, Λ_q, B_q, and A_v, Λ_v, B_v serve as adaptable LoRA parameters.

3.2.2 Prefix Tuning

Prefix Tuning [16] prepends a number of tunable, task-specific vectors to the input of the multi-head self-attention in *each* transformer block, whose original tokens can attend to as if they were virtual tokens. This method was originally developed for natural language processing and was eventually extended to vision applications as Deep Visual Prompt Tuning (VPT-Deep) [12]. We use VPT-Deep for all our experiments and call it Prefix Tuning to maintain uniformity with literature in the field. We denote the original sequence length L_0, the number of tunable vectors (i.e., prefix length) as L, and the Transformer layer input as $h_{in} \in \mathbb{R}^{D_{hidden} \times L_0}$. First, three linear projections, $W_Q, W_K, W_V \in \mathbb{R}^{D_{hidden} \times D_{hidden}}$ transform h_{in} into Query (Q), Key (K), and Value (V) matrices. The two prefix matrices $P_K \in \mathbb{R}^{D_{hidden} \times L}$ and $P_V \in \mathbb{R}^{D_{hidden} \times L}$ are pre-pended to K and V. The prefix matrix P is reparametrized by a feedforward network to stabilize the optimization procedure.

3.2.3 Adapters

Adapters [7] align the model to the target task by adding a trainable MLP after the feedforward layer in each Transformer block. The MLP consists of a down+up projection that condenses and recovers the size of the original hidden token space. This is better represented pictorially in Figure 3.3.

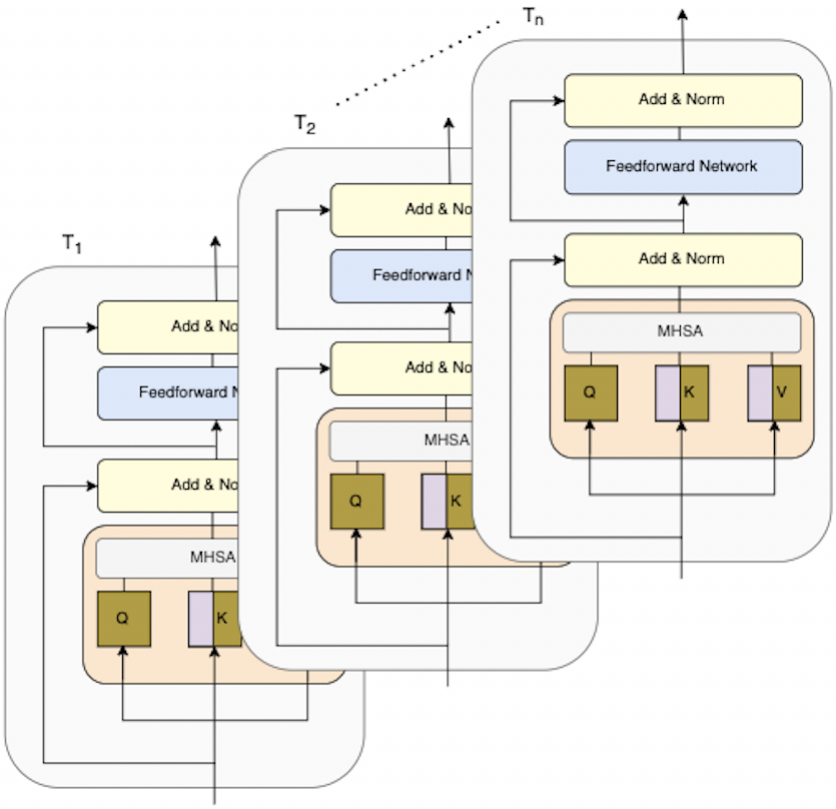

Figure 3.2: A pictorial representation of fine-tuning using Prefix Tuning [16].

Mathematically, we can denote the Adapter operation as

$$Z_A = W_1^T \phi(W_2^T Z_{FN}) \tag{3.4}$$

where, $W_1 \in \mathbb{R}^{D_{hidden} \times D_{mid}}, W_2 \in \mathbb{R}^{D_{mid} \times D_{hidden}}$. Here, D_{hidden} represents the hidden token space in the Transformer block, and D_{mid} represents the condensed embedding space of the Adapter MLP. Z_{FN} is the output of the feedforward network of the Transformer block after the residual connection and the layer normalization steps.

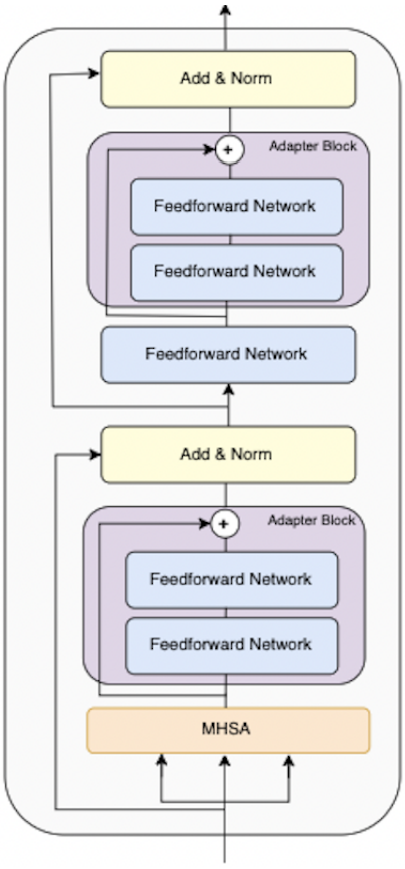

Figure 3.3: A pictorial representation of fine-tuning using Adapters [7].

3.3 Proposed Method

3.3.1 Intuition

During the analysis of individual PEFT methods, we observed that different methods often involve different parts of the Vision Transformer model in the image encoder of SAM. For instance, Adapters add an MLP after the feedforward layer in each Transformer block, while Prefix Tuning prepends tunable tensors before the multi-head self-attention layers. This unique property makes it possible to essentially combine multiple PEFT techniques in the proposed framework without interfering with each other.

Keeping the above in mind, we propose a unified MoPEFT framework which takes a hybrid approach by incorporating multiple PEFT methods as submodules. At a high level, MoPEFT shows better performance than its individual components due to two main reasons. Firstly, MoPEFT learns to dynamically access individual submodules based on the given task. This means that for a given data-task sample, a particular PEFT method may be allotted different weights or turned off entirely to ensure optimal performance in all cases. Secondly, we find that our MoPEFT framework generally outperforms the best-performing individual PEFT technique in multiple domains, suggesting that there may be benefits due to compounding effects that lead to better model effectiveness, as multiple PEFT techniques are used together. We show how we incorporate these different techniques under one framework in the next section.

3.3.2 Gating Mechanism

To achieve fine-grained control over the activation of the individual PEFT techniques that make up our MoPEFT framework, we take inspiration from current Mixture-of-Experts (MoE) methods [18,36,46]. Similar to the Sparsely-Gated-MoE method [42], we add a gating mechanism that dynamically links different PEFT methods to the relevant layers in the image encoder of SAM. As depicted in Figure 4.1, we add three trainable gates, one for each PEFT technique. Intuitively, if a particular PEFT technique is useful for a given data-task setup, then the output of the corresponding gate would be set to high. This would ensure that the specific PEFT plays a more important role during the execution.

For LoRA, our gate is not added directly in the form of the traditional MLP architecture as seen in MoE literature. Instead, we make use of the inherent *scaling factor*, α already present in the LoRA architecture as a pseudo-gating

mechanism. A higher α assigns more weight to the LoRA activations, while a lower α makes the effect of LoRA negligible. Thus, we already have a gating mechanism in place. To integrate this with our broader framework, we make the scaling factor learnable by using a feedforward network instead of specifying the constant manually.

For Prefix Tuning, we design a gating function $G_P \in (0, 1)$ that is applied to the Prefix vectors P_K and P_V keeping the representations of the original Key and Value tokens K and V intact. G_P is estimated using another feedforward network which takes in the input provided to the specific ViT layer.

For Adapters, we make use of the residual connection between the Adapter MLP and the feedforward network of the ViT Transformer block. This connection is responsible for summing up the input to the Adapter MLP. Our Adapter Gating Function $G_A \in (0, 1)$ estimates the importance of the Adapter MLP using a feedforward network with sigmoid activation. The Adapter MLP is essentially bypassed if $G_A = 0$.

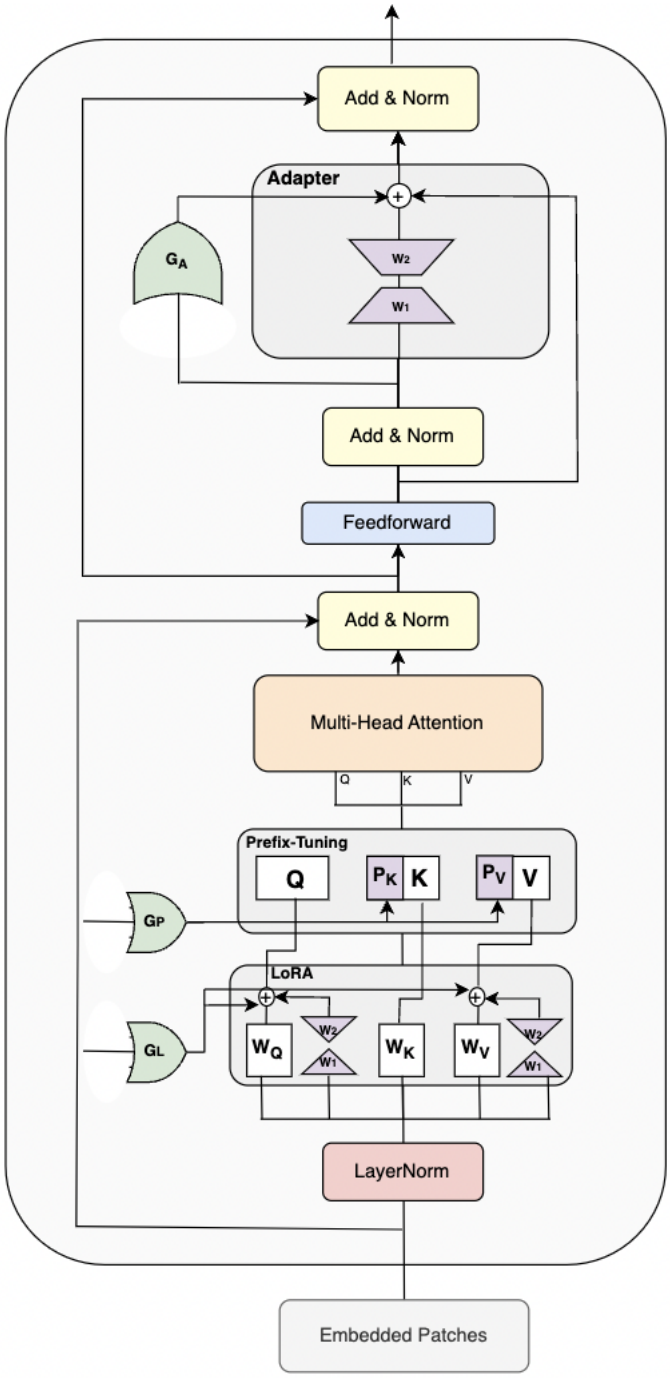

Figure 3.4: An overview of the proposed MoPEFT framework

Chapter 4

Experimental Results

We provide a brief overview of the MESS benchmark that was used for our evaluation and the implementation details. Following that, we compare our method with other PEFT techniques and individually analyze the significance of each technique and the gating mechanism.

4.1 Datasets

We employ the Multi-domain Evaluation of Semantic Segmentation (MESS) benchmark [2], which measures the mIOU score of models performing semantic segmentation tasks on 22 datasets spread across five major domains- General, Earth Monitoring, Medical Imaging, Engineering, and Agriculture and Biology. The collection of these datasets cover a variety of applications and allow us to conduct a holistic evaluation of our method in multiple domain-specific applications.

4.1.1 The MESS Benchmark

Inspired by the HELM Benchmark [17] proposed for the evaluation of large language models, Blumenstiel et al. proposed the Multi-domain Evaluation for Semantic Segmentation (MESS) [2] benchmark for zero-shot semantic segmentation tasks. It reveals the various challenges for the application of zero-shot semantic segmentation on domain-specific datasets, such as sensitivity to semantic prompts and label specificity across different domains. The authors of the MESS benchmark originally formulated a taxonomy over 120 datasets by specifying meta-characteristics and identifying visual characteristics of downstream tasks influencing the performance of zero-shot semantic segmentation

Domain	Dataset	Sensor Type	Segment Size	Number of Classes	Class similarity
General	BDD100K [53]	Visible Spectrum	Medium	19 (Medium)	Low
	Dark Zurich [39]	Visible Spectrum	Medium	20 (Medium)	Low
	MHPv1 [27]	Visible Spectrum	Small	19 (Medium)	High
	FoodSeg103 [49]	Visible Spectrum	Medium	104 (Many)	High
	ATLANTIS [30]	Visible Spectrum	Small	56 (Many)	Low
	DRAM [19]	Visible Spectrum	Medium	12 (Medium)	Low
Earth Monitoring	iSAID [47]	Visible Spectrum	Small	16 (Medium)	Low
	ISPRS Postdam [10]	Multispectral	Small	6 (Few)	Low
	WorldFloods [24]	Multispectral	Medium	3 (Binary)	Low
	FloodNet [35]	Visible Spectrum	Medium	10 (Few)	Low
	UAVid [22]	Visible Spectrum	Small	8 (Few)	High
Medical Imaging	Kvasir-Instr. [11]	Visible Spectrum	Medium	2 (Binary)	Low
	CHASE DB1 [40]	Microscopic	Small	2 (Binary)	Low
	CryoNuSeg [23]	Microscopic	Small	2 (Binary)	Low
	PAXRay-4 [41]	Electromagnetic	Large	4x2 (Binary)	Low
Engineering	CorrosionCS [6]	Visible Spectrum	Medium	4 (Few)	High
	DeepCrack [20]	Visible Spectrum	Small	2 (Binary)	Low
	PST900 [43]	Visible Spectrum	Medium	5 (Few)	High
Agriculture and Bio	ZeroWaste-f [5]	Electromagnetic	Small	5 (Few)	Low
	SUIM [9]	Visible Spectrum	Medium	8 (Few)	Low
	CUB-200 [45]	Visible Spectrum	Medium	201 (Many)	High
	CEFID [51]	Visible Spectrum	Small	3 (Few)	High

Table 4.1: Multi-domain benchmark (MESS) for zero-shot semantic segmentation models consists of 5 sensor types, different segment mask sizes and a total of 448 classes [2]

models. They then refined this taxonomy in multiple empirical-to-conceptual iterations and selected a representative set of datasets to make the benchmark informative, reproducible, and manageable. These datasets cover a variety of applications, resulting in a holistic evaluation of domain-specific applications.

We provide a short introduction of each dataset as follows: The *General* datasets include datasets with everyday scenes but are limited to more specific use-cases and niche image themes compared to standard image evaluation datasets. More specifically, the general domain focuses on driving (both during day and night time), food, and images of body parts. The *Earth Monitoring* datasets include iSAID [47] which consists of 15 object categories photographed through satellites. ISPRS Potsdam [10] and WorldFloods [24] are responsible for providing multispectral data and the authors employ IRRG false color mapping for their main evaluation. To maintain consistency, we replicate these settings for our work as well. Finally, UAVid [22] and FloodNet [35] are drone datasets that cover urban scenes. The *Medical Imaging* datsaets cover four different modalities within medical images itself- RGB images, whole slide imagery, retinal scans, and X-Ray scans. The benchmark also includes four different *Engineering* datasets. CorrosionCS [6] consists of close-up images of different stages of corrosion on bridges and other buildings. Similarly, DeepCrack [20] shows magnified images of cracks. PST900 [43] shows

Domain	Dataset	Vocabulary	Number of images	Task
General	BDD100K [53]	Generic	100	Driving
	Dark Zurich [39]	Generic	50	Driving
	MHPv1 [27]	Task-spec.	980	Body Parts
	FoodSeg103 [49]	Generic	2,135	Ingredients
	ATLANTIS [30]	Generic	1,295	Maritime
	DRAM [19]	Generic	718	Paintings
Earth Monitoring	iSAID [47]	Generic	4,055	Objects
	ISPRS Postdam [10]	Generic	504	Land Use
	WorldFloods [24]	Generic	160	Floods
	FloodNet [35]	Task Specific	5,571	Floods
	UAVid [22]	Task Specific	840	Objects
Medical Imaging	Kvasir-Instr. [11]	Generic	118	Endoscopy
	CHASE DB1 [40]	Domain Specific	20	Retina scan
	CryoNuSeg [23]	Domain Specific	30	WSI
	PAXRay-4 [41]	Domain Specific	180	X-Ray
Engineering	CorrosionCS [6]	Task Specific	44	Corrosion
	DeepCrack [20]	Generic	237	Cracks
	PST900 [43]	Generic	929	Conveyor
Agriculture and Bio	ZeroWaste-f [5]	Generic	288	Thermal
	SUIM [9]	Generic	110	Underwater
	CUB-200 [45]	Domain Specific	5,794	Bird Species
	CEFID [51]	Generic	21	Crops

Table 4.2: Multi-domain benchmark (MESS) for zero-shot semantic segmentation models consists of 22 downstream tasks, 3 different vocabularies, and 25,079 images [2]

thermal imagery for firefighter-related objects. The original MESS benchmark consists of ZeroWaste-f [5] which is a collection of different types of recyclable waste on a conveyor belt. The final domain, *Agriculture and Bio* covers biological-related datasets like SUIM [9] which is an underwater imagery dataset showing aquatic plants and fish, CUB-200 [45] which shows different species of birds, and CWFID [51] which shows agriculturally significant images like crop seedling and weeds.

4.2 Implementation Details

We use the Segment Anything Model [14] for all our fine-tuning and experiments. The traditional implementation of SAM consists of an image encoder (we use ViT-B for our experiments), a Prompt Encoder and a Mask Decoder. However, to better equip SAM for end-to-end semantic segmentation, we freeze the Prompt Encoder, always providing constant prompt tokens to the Mask Decoder when fine-tuning. Additionally, we apply full fine-tuning to the Mask

Decoder, since it is an extremely lightweight module.

For consistency, we include public implementations for all PEFT methods in our framework. We use a batch size of 4 and the Adam optimizer with a learning rate of $1x10^{-4}$ as a default with a weight decay of $1x10^{-4}$. All PEFT methods are implemented in the same codebase to ensure a fair comparison. We largely follow the default PEFT-specific hyperparameters and keep them unchanged across domains for uniformity. Unless otherwise specified, we set the LoRA rank $r = 8$ prefix length $L = 20$, and the adapter bottleneck size $D_{mid} = 64$ for our experiments.

4.3 Comparison with state-of-the-art

Table 4.4 shows the performance of our MoPEFT framework against the three most commonly used PEFT methods, i.e., LoRA [8], Prefix Tuning (VPT Deep) [12], and Adapters [7]. We compare these methods against a vanilla SAM framework (Baseline), fully fine-tuning the SAM decoder on the target dataset (decoderFT), and 'simple' Visual Prompt Tuning (VPT) [12], which is similar to Prefix Tuning except that the tunable tensors are added to only the first Transformer block as opposed to all of them. We measure the Mean Intersection-over-Union (mIOU) to compare performance across all method and datasets.

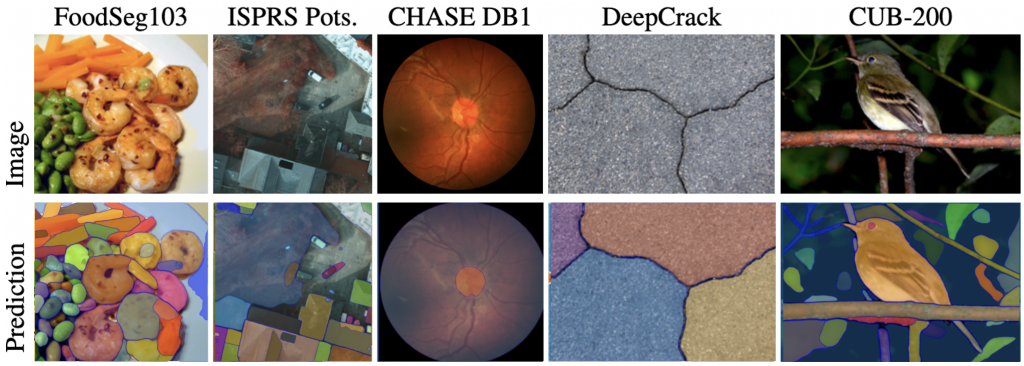

Figure 4.1: SAM [14] predictions for a range of domain-specific datasets from the MESS benchmark.

Domain	Dataset	Baseline	decoderFT	MoPEFTs
General	BDD100K [53]	41.58	42.84	50.93
	Dark Zurich [39]	20.91	23.42	31.19
	MHPv1 [27]	29.38	31.40	41.84
	FoodSeg103 [49]	10.48	14.93	22.99
	ATLANTIS [30]	17.33	20.62	30.03
	DRAM [19]	57.38	58.83	67.25
Earth Monitoring	iSAID [47]	62.59	63.14	68.29
	ISPRS Postdam [10]	29.73	29.92	40.42
	WorldFloods [24]	46.45	48.75	63.17
	FloodNet [35]	39.72	40.94	50.01
	UAVid [22]	60.19	60.96	71.12
Medical Imaging	Kvasir-Instr. [11]	46.82	48.32	71.92
	CHASE DB1 [40]	23.56	25.95	42.49
	CryoNuSeg [23]	38.06	40.36	59.88
	PAXRay-4 [41]	41.07	43.73	59.42
Engineering	CorrosionCS [6]	20.88	21.93	35.61
	DeepCrack [20]	59.02	62.27	72.59
	PST900 [43]	21.39	21.89	29.46
Agriculture and Bio	ZeroWaste-f [5]	0.43	1.12	2.99
	SUIM [9]	14.13	15.42	19.07
	CUB-200 [45]	38.41	40.29	48.46
	CEFID [51]	16.74	19.62	24.71

Table 4.3: Comparison of our MoPEFT framework with baseline and decoder-only fine-tuned SAM variants across multiple domains. Scores shown are mIOU scores.

4.4 Analysis of the Gating Mechanism

The results in this section provide a better understanding of what the MoE learns during fine-tuning. To gain a better understanding of our gating mechanism, we conduct an analysis by tracking the frequency of the selection of each PEFT technique across different datasets during inference. We present our detailed results in Figure 4.2.

Most notable in our results is the fact that different datasets give more preference to different PEFT techniques. For instance, the graph depicting iSAID [47] (an *Earth Monitoring* dataset in the MESS benchmark [2]), tends to select LoRA more often than the other two PEFT methods. Similarly, Kvasir-Instrument [11] (a *Medical Imaging* dataset in the MESS benchmark [2]) tends to select Adapters more often, instead of LoRA or Prefix Tuning. This obser-

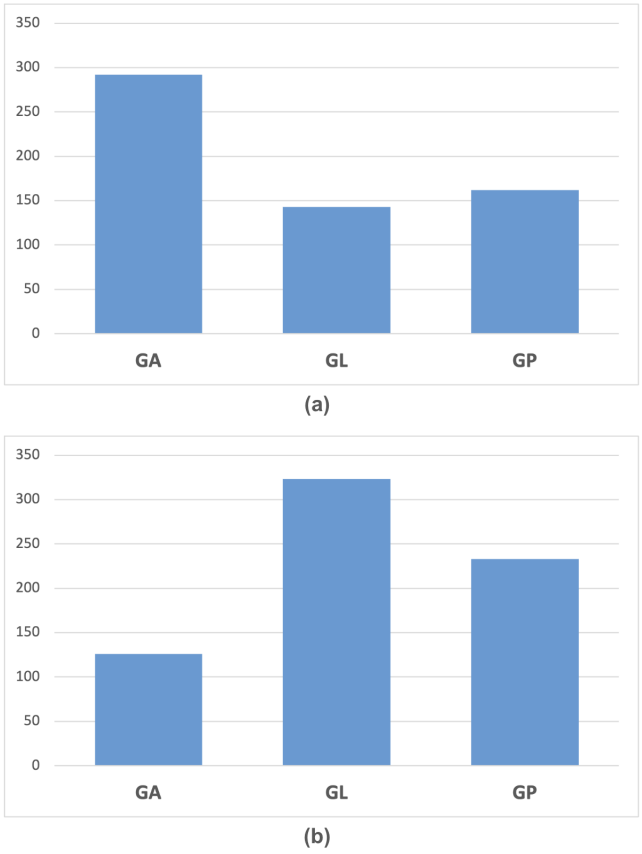

Figure 4.2: Number of times each PEFT method is called during inference. Different datasets display distinct patterns. We show results on **(a)** Kvasir-Instr. (*Medical Imaging*) [11] and **(b)** iSAID (*Earth Monitoring*) [47].

Domain	Dataset	LoRA	VPT Deep	VPT	Adapter	MoPEFTs
General	BDD100K [53]	49.39	46.24	43.18	47.03	50.93
	Dark Zurich [39]	30.82	27.16	24.49	26.72	31.19
	MHPv1 [27]	39.21	35.59	32.72	36.17	41.84
	FoodSeg103 [49]	22.45	19.91	16.02	20.05	22.99
	ATLANTIS [30]	28.03	27.61	24.91	27.91	30.03
	DRAM [19]	64.48	60.23	58.89	63.79	67.25
Earth Monitoring	iSAID [47]	66.29	65.61	64.71	64.82	68.29
	ISPRS Postdam [10]	38.25	33.52	31.42	35.77	40.42
	WorldFloods [24]	59.53	56.13	52.29	54.23	63.17
	FloodNet [35]	46.79	43.97	42.81	41.09	50.01
	UAVid [22]	69.43	65.39	61.19	63.59	71.12
Medical Imaging	Kvasir-Instr. [11]	66.97	58.31	52.23	62.06	71.92
	CHASE DB1 [40]	37.22	32.48	28.39	30.85	42.49
	CryoNuSeg [23]	54.93	48.12	44.81	36.22	59.88
	PAXRay-4 [41]	56.05	52.83	46.62	51.35	59.42
Engineering	CorrosionCS [6]	30.68	26.14	24.94	27.07	35.61
	DeepCrack [20]	69.83	66.02	63.81	65.82	72.59
	PST900 [43]	27.12	25.29	22.93	26.01	29.46
Agriculture and Bio	ZeroWaste-f [5]	2.53	3.83	2.36	1.43	2.99
	SUIM [9]	16.79	16.62	13.48	16.58	19.07
	CUB-200 [45]	47.35	45.28	42.56	44.79	48.46
	CEFID [51]	23.94	22.51	21.19	23.85	24.71

Table 4.4: Comparison of our MoPEFT framework with different PEFT fine-tuned SAM variants across multiple domains. Scores shown are mIOU scores.

vation supports our initial claim that our gating mechanism learns to dynamically select appropriate PEFT techniques based on the provided data-task setup. This reinforces the significance of the MoPEFT framework in tailoring its selection to the unique characteristics of diverse datasets enhancing its effectiveness across different domains.

4.5 Analysis of individual PEFT methods

4.5.1 Analysis of LoRA

From Table 4.4 we notice that LoRA usually performs the best out of all the representative PEFT methods. While it does not outperform our MoPEFT framework due to compounding effects, it perfroms significantly better than traditional decoder fine-tuning on all domains. On average, we see an increase of 7.7% across all datasets when compared to decoder fine-tuning, and an increase of 9.5% compared to the baseline performance.

4.5.2 Analysis of Prefix Tuning

Visual Prompt Tuning (denoted by VPT in Table 4.4) and Visual Prefix Tuning or Deep Visual Prompt Tuning (denoted by VPT-Deep) show similar performance to each other. On average, we see that Prefix Tuning generally outperforms Prompt Tuning, a phenomenon that has also been observed during fine-tuning Large Language Models (LLMs) [15].

4.5.3 Analysis of Adapter

The performance of Adapter in Table 4.4 is shown to be on-par with Prefix Tuning. While it consistently outperforms Visual Prompt Tuning, it usually fails to outperform LoRA. While other papers [3, 32] tuned hyperparameters for each specific dataset, we went with standard parameters and a bottleneck size of 48 across all our datasets. We also slightly deviate from the original Adapter implementation [7] and port the AdapterFusion approach [31] for visual tasks, adding only one Adapter layer in each ViT block instead of two.

4.6 Efficiency Comparison

We benchmark the efficiency of different PEFT methods against MoPEFT and full fine-tuning of SAM (including the image encoder). Table 4.5 shows a quantative comparison of the number of parameters, training, and inference time relative to full fine-tuning. All experiments were conducted on the FloodNet [35] dataset.

4.6.1 Parameter Efficiency

As the number of trainable parameters in all PEFT methods are negligible compared to full fine-tuning, combining multiple PEFT methods in our framework still does not lead to significant increases in the overall number of trainable parameters.

4.6.2 Training and Inference Efficiency

Due to parameter efficiency, all representative PEFT methods train comparatively faster than full fine-tuning and incorporating multiple PEFTs into MoPEFT is only slightly slower due to the additional training of the gating mechanisms. In terms of inference time, we see that decoder fine-tuning has no increase compared to full fine-tuning, since full fine-tuning also includes fine-tuning the decoder. However, the inference time of other PEFT methods

Method	Params.	$\mathbf{T_{train}}$	$\mathbf{T_{inference}}$
Full fine-tuning	100%	100%	100%
Decoder fine-tuning	-	38%	100%
LoRA	0.24%	55%	104%
Prompt Tuning (VPT)	0.08%	49%	109%
Prefix Tuning (VPT-Deep)	0.17%	54%	114%
Adapter	0.83%	55%	108%
MoPEFT	1.47%	63%	124%

Table 4.5: Number of trainable parameters and time required during training and inference relative to full fine-tuning.

are considerably larger since they require more FLOPs during test time. Due to multiple gating mechanisms and combinations of other PEFT methods, MoPEFT has a significantly larger inference times compared to other techniques. We aim to develop newer techniques to reduce this overhead as part of future endeavors.

4.7 Ablation Studies

This section presents a deeper overview into the selection of hyperparameters for the optimal configuration of our MoPEFTs framework. We show ablations by varying the main parameter of each of our PEFT methods and measuring the change in the performance of our overall framework. As such, we conduct experiments with changing the rank of the LoRA matrices, the Prefix Length of the embedded tensors, and the bottleneck size of the Adapter MLP. While there have been previous studies [13, 32] on understanding the effect of these changes, our goal was to see if varying these parameters brought about a change in our MoPEFT framework. This also gave us a chance to observe the compounding effect of implementing multiple PEFT methods together from a different perspective. We conduct these ablation studies on only two out of the four domains: earth monitoring and medical imagery.

4.7.1 Effect of rank r in LoRA

Table 4.6 shows the effect seen on overall performance when we vary the rank of the A and B matrices in LoRA. The rightmost column is the baseline performance of vanilla SAM without MoPEFT. We observe that a increasing rank correlates to a higher overall performance. However, a very high rank

Datasets	rank = 4	rank = 8	rank = 16	**Basline**
iSAID [47]	59.89	64.91	68.29	62.59
ISPRS Postdam [10]	31.92	38.27	40.42	29.73
WorldFloods [24]	56.24	60.19	63.17	46.45
FloodNet [35]	39.57	42.22	50.01	39.72
UAVid [22]	62.03	68.94	71.15	60.19
Kvasir-Instrument [11]	60.72	67.43	71.92	46.82
CHASE DB1 [40]	36.61	40.05	42.49	23.56
CryoNuSeg [23]	51.09	54.39	59.88	38.06
PAXRay-4 [41]	48.11	52.83	59.42	41.07

Table 4.6: Effect of LoRA rank on task performance. Scores shown are mIOU scores.

would also start to disobey the 'parameter-efficient' rule of the PEFT. We hope future works can design a more efficient way of incorporating high-rank LoRA matrices in our MoPEFT framework.

4.7.2 Effect of L in Prefix Tuning

Table 4.7 shows the variance in performance as a function of the prefix length. We compare four different prefix lengths against the baseline performance of vanilla SAM. The results illustrate that increasing the prefix length from $L = 5$ to $L = 20$ shows a gradual increase in performance, while performance drops for $L = 25$. This indicates that, unlike the rank of LoRA matrices, simply increasing the prefix length may not work for all scenarios. Moreover, a larger L leads to significantly increased delays in training and inference due to costly multi-head attention. In summary, using more trainable parameters for prefix tuning does not always guarantee better performance.

4.7.3 Effect of D_{mid} in Adapters

Table 4.8 shows the effect of varying the bottleneck size D_{mid} of the Adapter MLP on the overall performance of the MoPEFT framework. The results show that the performance of the Adapter increases gradually with an increase in the bottleneck size. This suggests that a larger bottleneck size could be beneficial for an Adapter. However, this also has the same problem as the rank in LoRA. While $D_{mid} = 256$ shows the best performance, it also has $3.5\times$ more trainable parameters than $D_{mid} = 48$, which we use in our main results in Table 4.4. Continually increasing the bottleneck size affects the parameter-

Datasets	**L=5**	**L=10**	**L=20**	**L=25**	**Baseline**
iSAID [47]	62.14	63.14	68.29	64.18	62.59
ISPRS Postdam [10]	28.92	29.92	40.42	37.94	29.73
WorldFloods [24]	46.75	48.75	63.17	64.02	46.45
FloodNet [35]	39.94	40.94	50.01	51.22	39.72
UAVid [22]	58.12	60.96	71.10	68.72	60.19
Kvasir-Instrument [11]	43.35	48.32	71.92	70.04	46.82
CHASE DB1 [40]	25.20	25.95	42.49	43.11	23.56
CryoNuSeg [23]	39.21	40.36	59.88	52.93	38.06
PAXRay-4 [41]	40.43	41.07	59.42	58.32	41.07

Table 4.7: Effect of Prefix Length on task performance. Scores are mIOU scores.

efficiency aspect of our framework.

Adapters	**$D_{mid} = 48$**	**$D_{mid} = 64$**	**$D_{mid} = 128$**	**$D_{mid} = 256$**	**Baseline**
iSAID [47]	59.02	64.82	65.03	68.29	62.59
ISPRS Postdam [10]	19.23	35.77	36.24	40.42	29.73
WorldFloods [24]	13.01	54.23	57.29	63.17	46.45
FloodNet [35]	21.39	41.09	44.68	50.02	39.72
UAVid [35]	10.43	63.59	69.82	71.12	60.19
Kvasir-Instrument [11]	38.41	62.06	67.31	71.92	46.82
CHASE DB1 [40]	16.74	30.85	34.69	42.49	23.56
CryoNuSeg [23]	32.21	36.22	44.20	59.88	38.06

Table 4.8: Effect of Adapter bottleneck size on task performance. Scores are mIOU scores.

Chapter 5

Conclusion

5.1 Conclusion

In this work, introduce a new framework, Mixture-of-PEFTs (MoPEFT), that is inspired by Mixture-of-Experts methods. Our MoPEFT framework dynamically learns to activate or deactivate a particular PEFT technique based on a given data-task setup. Since each PEFT technique modifies the internal representation of the model on its own different way, our framework allows us to selectively utilize the best representation for a given scenario. Moreover, this also helps us mitigate the non-trivial task of choosing a particular PEFT technique for a specific use-case. To test out our framework, we present a comprehensive study of the three most widely used PEFT techniques- LoRA [8], Prefix Tuning [16], and Adapters [7]. We take the Segment Anything Model [14] and apply the mentioned PEFT techniques to it, fine-tuning it across a variety of different datasets across five different domains. We benchmark their efficacy and finally combine them all into our MoPEFT framework by adding them to the same model and gating them to control their effectiveness on the overall result. We then compare our MoPEFT framework against traditional PEFT techniques on the same benchmark. Our results show that MoPEFT usually outperforms all traditional fine-tuning techniques on multiple datasets across different domains.

5.2 Future Work

There are two major avenues of future work that we envision for this work. Firstly, our experiments are focused primarily on the Segment Anything Model. However, our MoPEFT framework only requires an underlying Vision Trans-

former model to function. This means that other vision foundation models such as Florence [54] or VISION-MAE [21] can also be fine-tuned using our proposed methods. It would be interesting to see how other foundation models in different tasks fare after being fine-tuned using MoPEFT.

Our work currently focuses on the three most widely-used PEFT techniques-LoRA, Prefix Tuning, and Adapters. Since these are the most simple and form the basis for more sophisticated fine-tuning methodologies, it made sense to have them as the building blocks for our MoPEFT framework. However, we can also swap one or more of the techniques used with more advanced techniques that come from the same *family* of PEFT techniques. This means that, in our framework, LoRA can be swapped with another technique that modifies the representation of the model in a similar way (e.g. BitFit [55] or GaLoRE [56]).

Bibliography

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Benedikt Blumenstiel, Johannes Jakubik, Hilde Kühne, and Michael Vössing. What a mess: Multi-domain evaluation of zero-shot semantic segmentation. *Advances in Neural Information Processing Systems*, 36, 2024.

[3] Tianrun Chen, Lanyun Zhu, Chaotao Ding, Runlong Cao, Yan Wang, Zejian Li, Lingyun Sun, Papa Mao, and Ying Zang. Sam fails to segment anything? – sam-adapter: Adapting sam in underperformed scenes: Camouflage, shadow, medical image segmentation, and more, 2023.

[4] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

[5] N. Cohen, Y. Newman, and A. Shamir. Semantic segmentation in art paintings. *Computer Graphics Forum*, 41(2):261–275, 2022.

[6] Seyed Mohammad Hassan Erfani, Zhenyao Wu, Xinyi Wu, Song Wang, and Erfan Goharian. Atlantis: A benchmark for semantic segmentation of waterbody images. *Environmental Modelling & Software*, 149:105333, 2022.

[7] Ruidan He, Linlin Liu, Hai Ye, Qingyu Tan, Bosheng Ding, Liying Cheng, Jia-Wei Low, Lidong Bing, and Luo Si. On the effectiveness of adapter-based tuning for pretrained language model adaptation. *arXiv preprint arXiv:2106.03164*, 2021.

[8] 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.

[9] Md Jahidul Islam, Chelsey Edge, Yuyang Xiao, Peigen Luo, Muntaqim Mehtaz, Christopher Morse, Sadman Sakib Enan, and Junaed Sattar. Semantic segmentation of underwater imagery: Dataset and benchmark. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1769–1776, 2020.

[10] Isprs ISPRS. 2d semantic labeling contest, 2014.

[11] Debesh Jha, Sharib Ali, Krister Emanuelsen, Steven A Hicks, Vajira Thambawita, Enrique Garcia-Ceja, Michael A Riegler, Thomas de Lange, Peter T Schmidt, Håvard D Johansen, et al. Kvasir-instrument: Diagnostic and therapeutic tool segmentation dataset in gastrointestinal endoscopy. In *MultiMedia Modeling: 27th International Conference, MMM 2021, Prague, Czech Republic, June 22–24, 2021, Proceedings, Part II 27*, pages 218–229. Springer, 2021.

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

[13] Lei Ke, Mingqiao Ye, Martin Danelljan, Yu-Wing Tai, Chi-Keung Tang, Fisher Yu, et al. Segment anything in high quality. *Advances in Neural Information Processing Systems*, 36, 2024.

[14] 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.

[15] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

[16] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.

[17] Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu,

Ananya Kumar, et al. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*, 2022.

[18] Bin Lin, Zhenyu Tang, Yang Ye, Jiaxi Cui, Bin Zhu, Peng Jin, Junwu Zhang, Munan Ning, and Li Yuan. Moe-llava: Mixture of experts for large vision-language models. *arXiv preprint arXiv:2401.15947*, 2024.

[19] Yahui Liu, Jian Yao, Xiaohu Lu, Renping Xie, and Li Li. Deepcrack: A deep hierarchical feature learning architecture for crack segmentation. *Neurocomputing*, 338:139–153, 2019.

[20] Yahui Liu, Jian Yao, Xiaohu Lu, Renping Xie, and Li Li. Deepcrack: A deep hierarchical feature learning architecture for crack segmentation. *Neurocomputing*, 338:139–153, 2019.

[21] Zelong Liu, Andrew Tieu, Nikhil Patel, Alexander Zhou, George Soultanidis, Zahi A. Fayad, Timothy Deyer, and Xueyan Mei. Vision-mae: A foundation model for medical image segmentation and classification, 2024.

[22] Ye Lyu, George Vosselman, Gui-Song Xia, Alper Yilmaz, and Michael Ying Yang. Uavid: A semantic segmentation dataset for uav imagery. *ISPRS journal of photogrammetry and remote sensing*, 165:108–119, 2020.

[23] Amirreza Mahbod, Gerald Schaefer, Benjamin Bancher, Christine Löw, Georg Dorffner, Rupert Ecker, and Isabella Ellinger. Cryonuseg: A dataset for nuclei instance segmentation of cryosectioned h&e-stained histological images. *Computers in biology and medicine*, 132:104349, 2021.

[24] Gonzalo Mateo-Garcia, Joshua Veitch-Michaelis, Lewis Smith, Silviu Vlad Oprea, Guy Schumann, Yarin Gal, Atılım Güneş Baydin, and Dietmar Backes. Towards global flood mapping onboard low cost satellites with machine learning. *Scientific reports*, 11(1):7249, 2021.

[25] Maciej A Mazurowski, Haoyu Dong, Hanxue Gu, Jichen Yang, Nicholas Konz, and Yixin Zhang. Segment anything model for medical image analysis: an experimental study. *Medical Image Analysis*, 89:102918, 2023.

[26] John J Nay, David Karamardian, Sarah B Lawsky, Wenting Tao, Meghana Bhat, Raghav Jain, Aaron Travis Lee, Jonathan H Choi, and Jungo Kasai. Large language models as tax attorneys: a case study in

legal capabilities emergence. *Philosophical Transactions of the Royal Society A*, 382(2270):20230159, 2024.

[27] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *Proceedings of the IEEE international conference on computer vision*, pages 4990–4999, 2017.

[28] Hien D Nguyen and Faicel Chamroukhi. Practical and theoretical aspects of mixture-of-experts modeling: An overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1246, 2018.

[29] Lucas Prado Osco, Qiusheng Wu, Eduardo Lopes de Lemos, Wesley Nunes Gonçalves, Ana Paula Marques Ramos, Jonathan Li, and José Marcato Junior. The segment anything model (sam) for remote sensing applications: From zero to one shot. *International Journal of Applied Earth Observation and Geoinformation*, 124:103540, 2023.

[30] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[31] Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. Adapterfusion: Non-destructive task composition for transfer learning, 2021.

[32] Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. Adapterhub: A framework for adapting transformers. *arXiv preprint arXiv:2007.07779*, 2020.

[33] Sara Pieri, Sahal Shaji Mullappilly, Fahad Shahbaz Khan, Rao Muhammad Anwer, Salman Khan, Timothy Baldwin, and Hisham Cholakkal. Bimedix: Bilingual medical mixture of experts llm. *arXiv preprint arXiv:2402.13253*, 2024.

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

[35] Maryam Rahnemoonfar, Tashnim Chowdhury, Argho Sarkar, Debvrat Varshney, Masoud Yari, and Robin Roberson Murphy. Floodnet: A high resolution aerial imagery dataset for post flood scene understanding. *IEEE Access*, 9:89644–89654, 2021.

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

[37] Rajat Sahay and Andreas Savakis. On aligning sam to remote sensing data. In *Geospatial Informatics XIV*, volume 13037, pages 10–18. SPIE, 2024.

[38] Rajat Sahay, Georgi Thomas, Chowdhury Sadman Jahan, Mihir Manjrekar, Dan Popp, and Andreas Savakis. On the importance of attention and augmentations for hypothesis transfer in domain adaptation and generalization. *Sensors*, 23(20):8409, 2023.

[39] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Guided curriculum model adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7374–7383, 2019.

[40] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Guided curriculum model adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7374–7383, 2019.

[41] Constantin Seibold, Simon Reiß, Saquib Sarfraz, Matthias A Fink, Victoria Mayer, Jan Sellner, Moon Sung Kim, Klaus H Maier-Hein, Jens Kleesiek, and Rainer Stiefelhagen. Detailed annotations of chest x-rays via ct projection for report understanding. *arXiv preprint arXiv:2210.03416*, 2022.

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

[43] Shreyas S Shivakumar, Neil Rodrigues, Alex Zhou, Ian D Miller, Vijay Kumar, and Camillo J Taylor. Pst900: Rgb-thermal calibration, dataset and segmentation network. In *2020 IEEE international conference on robotics and automation (ICRA)*, pages 9441–9447. IEEE, 2020.

[44] Xian Sun, Peijin Wang, Zhiyuan Yan, Feng Xu, Ruiping Wang, Wenhui Diao, Jin Chen, Jihao Li, Yingchao Feng, Tao Xu, et al. Fair1m: A benchmark dataset for fine-grained object recognition in high-resolution remote sensing imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 184:116–130, 2022.

[45] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. Cub-200. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[46] Yaqing Wang, Subhabrata Mukherjee, Xiaodong Liu, Jing Gao, Ahmed Hassan Awadallah, and Jianfeng Gao. Adamix: Mixture-of-adapter for parameter-efficient tuning of large language models. *arXiv preprint arXiv:2205.12410*, 1(2):4, 2022.

[47] Syed Waqas Zamir, Aditya Arora, Akshita Gupta, Salman Khan, Guolei Sun, Fahad Shahbaz Khan, Fan Zhu, Ling Shao, Gui-Song Xia, and Xiang Bai. isaid: A large-scale dataset for instance segmentation in aerial images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 28–37, 2019.

[48] Chaoyi Wu, Weixiong Lin, Xiaoman Zhang, Ya Zhang, Yanfeng Wang, and Weidi Xie. Pmc-llama: Towards building open-source language models for medicine, 2023.

[49] Xiongwei Wu, Xin Fu, Ying Liu, Ee-Peng Lim, Steven CH Hoi, and Qianru Sun. A large-scale benchmark for food image segmentation. In *Proceedings of the 29th ACM international conference on multimedia*, pages 506–515, 2021.

[50] Dongkuan Xu, Ian EH Yen, Jinxi Zhao, and Zhibin Xiao. Rethinking network pruning–under the pre-train and fine-tune paradigm. *arXiv preprint arXiv:2104.08682*, 2021.

[51] Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Bin Xiao, Ce Liu, Lu Yuan, and Jianfeng Gao. Unified contrastive learning in image-text-label space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19163–19173, June 2022.

[52] Jinyu Yang, Mingqi Gao, Zhe Li, Shang Gao, Fangjing Wang, and Feng Zheng. Track anything: Segment anything meets videos. *arXiv preprint arXiv:2304.11968*, 2023.

[53] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2636–2645, 2020.

[54] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, Ce Liu, Mengchen Liu, Zicheng Liu, Yumao Lu, Yu Shi, Lijuan Wang, Jianfeng Wang, Bin Xiao, Zhen Xiao, Jianwei Yang, Michael Zeng, Luowei Zhou, and Pengchuan Zhang. Florence: A new foundation model for computer vision, 2021.

[55] Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models, 2022.

[56] Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. Galore: Memory-efficient llm training by gradient low-rank projection, 2024.

[57] Zihan Zhong, Zhiqiang Tang, Tong He, Haoyang Fang, and Chun Yuan. Convolution meets lora: Parameter efficient finetuning for segment anything model, 2024.

Appendices

Appendix A

First Appendix

In addition to the experiments conducted in Chapter 4, we also conducted a benchmarking study of PEFT techniques on the FAIR-1M dataset [44]. Since FAIR-1M is an object detection dataset, as opposed to segmentation, we had to modify the overall structure of the Segment Anything Model in order to accommodate the new task. As such, we added a `Masks2Boxes` head at the end of the pipeline which takes the maximally-enclosing area within a given segmentation mask to generate a bounding box for object detection. Inspired by [57], we also add an MLP to the mask decoder in order to predict the class along with the bounding box. Our modified mask decoder structure is illustrated in Figure A.1 A complete overview of the pipeline can be seen in Figure **??**. As such, the evaluation metric for this task is set to the mean average precision (mAP) instead on the mIOU, as the latter is used in image segmentation tasks. Since the mask decoder was always fine-tuned completely, we did not need to make any other changes in the fine-tuning procedure. The results from our experiments on FAIR-1M can be found in Table A.1.

A.1 Dataset Description

We begin by giving a brief overview of the FAIR-1M dataset used for evaluation. The dataset consists of 15,000 images containing more than 1 million fine-grained objects in high-resolution remote sensing images. The resolution in the dataset ranges from 0.3m to 0.8m and is spread across multiple countries and regions. The dataset is divided into 5 major categories and 37 sub-categories, all of which can be found in Table A.1.

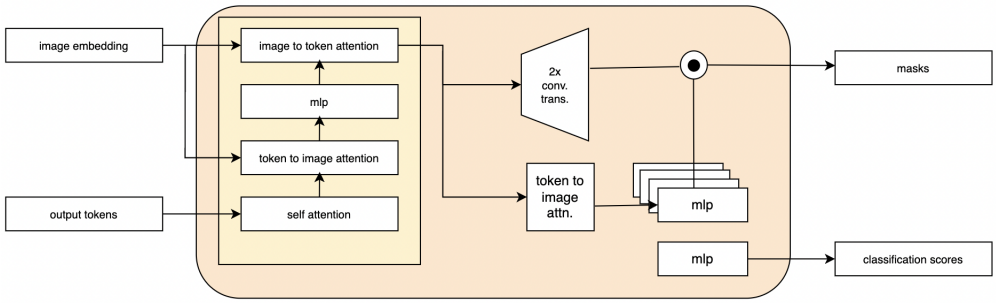

Figure A.1: Our modified structure of the SAM mask decoder (inspired from [57]).

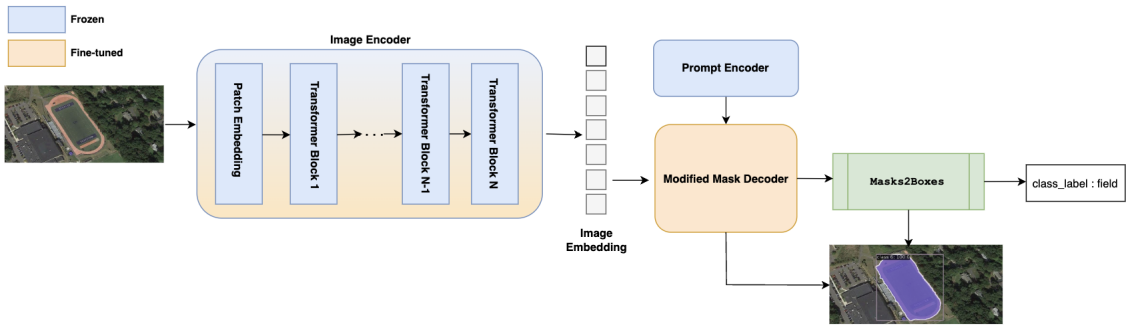

Figure A.2: Overall pipeline of our SAM model applied to the FAIR-1M dataset.

A.2 Results

This section shows the results of applying different variants of a fine-tuned SAM model on the FAIR-1M dataset. We apply LoRA [8], Simple Prompt Tuning [15], and Prefix Tuning [16] to all 37 sub-categories in the dataset.

From Table A.1, we can see that the model has no trouble generalizing to the first and fourth categories (Airplane and Court), but other categories show significant degradation in performance. We attribute this to the fine-grained nature of the images within these categories. Regardless of whatever fine-tuning technique has been applied, a photo of a Tennis Court would always implicitly have more useful information for detection than a Small Car, especially when depicted from the perspective of an aerial image. We also see that Prefix Tuning usually outperforms simple Prompt Tuning. This is in line with previous literature both in vision [15] as well as language models [31]. However, Prefix Tuning does require more parameters to be fine-tuned since we add a prefix to the beginning of every Transformer block instead of just the first one (as we do in Prompt Tuning).

		mAP	**Mean mAP**	**mAP**	**Mean mAP**	**mAP**	**Mean mAP**	**mAP**	**Mean mAP**
(Category, Class Label)		**Baseline**		**Prompt Tuning**		**Prefix Tuning**		**LoRA**	
Airplane	Boeing737	37.39	49.83	38.92	52.27	45.09	54.26	48.17	56.75
	Boeing747	85.24		78.39		83.16		84.20	
	Boeing777	17.25		18.81		23.48		23.49	
	Boeing787	53.95		56.98		62.64		67.14	
	C919	13.31		25.37		20.88		22.91	
	A220	48.24		49.71		51.24		56.49	
	A321	70.52		65.84		69.59		78.38	
	A330	71.02		73.63		70.77		74.29	
	A350	66.93		75.41		75.81		72.10	
	ARJ21	34.50		39.61		39.89		40.31	
Ship	PassengerShip	16.36	27.97	19.75	31.39	20.74	34.02	24.62	37.72
	Motorboat	61.46		63.84		67.59		68.13	
	FishingBoat	8.77		13.03		14.61		20.41	
	Tugboat	40.13		32.66		36.55		38.27	
	EngineeringShip	12.24		20.75		15.03		20.31	
	LiquidCargoShip	21.30		26.16		29.53		38.16	
	DryCargoShip	38.75		38.62		42.72		44.13	
	Warship	24.76		36.35		45.41		47.74	
Vehicle	SmallCar	12.70	23.19	26.31	37.24	34.19	41.39	35.92	40.61
	Bus	23.69		54.89		59.72		62.52	
	CargoTruck	41.16		56.75		65.75		65.19	
	DumpTruck	45.75		61.28		66.31		69.72	
	Van	55.82		76.38		77.61		63.21	
	Trailer	13.20		23.19		25.27		26.01	
	Tractor	4.12		10.62		14.96		12.83	
	Excavator	11.72		21.48		23.35		21.81	
	TruckTractor	0.56		4.27		5.38		8.24	
Court	BasketballCourt	50.45	54.45	62.17	77.46	65.43	80.09	66.19	81.23
	TennisCourt	80.56		85.67		90.60		92.84	
	FootballField	55.81		72.66		74.87		75.30	
	BaseballField	85.45		89.36		89.45		90.59	
Road	Intersection	59.35	37.20	61.34	40.51	64.67	45.32	59.47	49.76
	Roundabout	20.65		22.62		29.46		47.62	
	Bridge	31.60		37.58		41.83		42.18	

Table A.1: Comparison of performance of different PEFT techniques on FAIR-1M [44]