Title	MMtuning : an advanced multi-adapter framework for efficient multimodal large language model fine-tuning and its application
Sub Title	
Author	キョウ, 立(Qiao, Li) 杉浦, 一徳(Sugiura, Kazunori)
Publisher	慶應義塾大学大学院メディアデザイン研究科
Publication year	2024
Jtitle	
JaLC DOI	
Abstract	
Notes	修士学位論文. 2024年度メディアデザイン学 第1143号
Genre	Thesis or Dissertation
URL	https://koara.lib.keio.ac.jp/xoonips/modules/xoonips/detail.php?koara_id=KO40001001-00002024-1143

Master's Thesis
Academic Year 2024

MMtuning: An Advanced Multi-adapter Framework for Efficient Multimodal Large Language Model Fine-Tuning and Its Application

Keio University
Graduate School of Media Design

LI QIAO

A Master's Thesis
submitted to Keio University Graduate School of Media Design
in partial fulfillment of the requirements for the degree of
Master of Media Design

LI QIAO

Master's Thesis Advisory Committee:

Professor Kazunori Sugiura	(Main Research Supervisor)
Professor Akira Kato	(Sub Research Supervisor)

Master's Thesis Review Committee:

Professor Kazunori Sugiura	(Chair)
Professor Hideki Sunahara	(Co-Reviewer)
Professor Nanako Ishido	(Co-Reviewer)

Abstract of Master's Thesis of Academic Year 2024

MMtuning: An Advanced Multi-adapter Framework for Efficient Multimodal Large Language Model Fine-Tuning and Its Application

Category: Science / Engineering

Summary

Multimodal deep learning models, especially some multimodal large language models (MM-LLMs), have demonstrated strong comprehensive capabilities in several real-world application domains. MM-LLMs require additional fine-tuning (retraining) in real-world downstream tasks and application deployment. However, traditional fine-tuning methods require high computational resources and datasets of a certain size. To address this, we introduce MMtuning—to our knowledge, an innovative and first approach Parameter-Efficient Fine-Tuning (PEFT) framework tailored for the Multimodal Large Language Models (MM-LLMs), i.e., competitive fine-tuning can be achieved at low-cost. MMtuning is a generalized paradigm that efficiently integrates input information from different modalities with limited computational resources. Through IEB (Interaction Enhancement Block) and MWB (Modality Weights Block), MMtuning builds a multimodal skill allocation matrix that effectively extracts the optimal subset of multimodal skills from the dynamic Skill Inventory built by the Multi-Adapter block according to the input from downstream tasks. Experiments based on ScienceQA and Visual7W two VQA datasets, validate the potency of MMtuning, demonstrating its superiority in multimodal learning with extant baselines that employ equivalent parameter volumes. In addition, according to the results of the NASA Task Load Index (NASA-TLX), the MMtuning-based visual-language multimodal model fine-tuning web platform can significantly reduce users' technological barriers and workload in acquiring multimodal models, compared to the fine-tuning process of programming and complex software.

Keywords:

Deep Learning, multimodal large language models, Parameter-Efficient Fine-Tuning, media literacy

Keio University Graduate School of Media Design

LI QIAO

Contents

List of Figures

List of Tables

Acknowledgements

I am indebted to Professor Kazunori Sugiura for guiding me not only about research but with many aspects of my life.

Chapter 1
Introduction

1.1. Research background and Problem statement

The pretrain-and-finetune paradigm has become the standard for many tasks utilizing deep learning models for real-world applications. The Model pre-trained on large-scale data requires only a small amount of tuning to adapt to specific downstream tasks [1]. This approach has achieved various applications on various unimodal tasks and demonstrated excellent performance in unimodal multitasking scenarios. Further, the success of pre-trained single-modal models, especially notable Large language models (LLMs), has paved the way for the emergence of powerful multimodal large language models (MM-LLMs). Along with MM-LLMs, the paradigm of pretrain-and-finetune has been generalized to the field of multimodal tasks. And, in recent years, multimodal large models (MM-LLMs) have been widely used in deep learning tasks, real-world industries, and everyday life.

Through pre-trained on a vast corpus of instruction-output pairs, MM-LLMs such as Blip2 [2], LLaVA [3], and Qwen-VL [4], have shown remarkable performance on a variety of multimodal tasks in following human instructions [5]. For example, utilizing multimodal language models to detect defects in existing code bases and categorizing them turns out to be an extremely common way [6]. Moreover, multimodal recommender systems are widely used in various social software due to the integration of multiple modality user information, which significantly enhances the ability to build detailed user profiles with analysis and recommendation [7]. Despite this, following the pretrain-and-finetune paradigm, deploying these MM-LLMs in real-world downstream tasks and applications also requires additional fine-tuning, which means training those models again in the new datasets for better performance. However, following the escalating size of MM-LLMs, traditional supervised fine-tuning methods face a strong reliance on

computational resources and large-scale datasets, which makes the cost of fine-tuning non-negligible.

In addition to this, in the current context of rapid development of AI and deep learning, non-AI professionals or ordinary designers face several challenges in acquiring and applying fine-tuned multimodal models. First, the fine-tuning process of multimodal models usually requires specialized programming skills and an in-depth understanding of deep learning principles, which makes the technical threshold higher. Second, due to the large scale of multimodal model parameters, fine-tuning requires a large amount of computational resources and storage space, which is a great challenge for users with limited resources. Therefore, how to efficiently enable non-AI professionals to easily accomplish the fine-tuning of multimodal models under limited resources and get the multimodal model that loads their task requirements has also become an urgent problem that needs to be solved at present.

In this context, recently Parametric-Efficient Fine-Tuning (PEFT) methods have become the preferred solution for fine-tuning large-scale deep learning models with limited computational resources and datasets. PEFT methods such as LoRA [8], (IA)3 [9], and LoHA [10] fine-tune pre-trained models by freezing the main parameters and training only the adapter layer inserted into them. These methods can significantly reduce the number of model parameters that need to be trained during the fine-tuning process, which is only about 5-10 percent of the original model parameters, and effectively avoid the problem that the fine-tuning effect is not as good as the non-fine-tuning effect due to the mismatch between the amount of data and the size of the model. Overall, this technique not only saves computational resources but also significantly improves the performance of the model in new downstream tasks. In addition, MoE(Mixture of Experts)-based PEFT's architecture MoLoRA [11], Polytropon (Poly) [12], and customized Polytropon (C-Poly) [13] have shown excellent fine-tuning capabilities in multi-tasking scenarios with a single model with limited resources. These architectures make adept utilization of the gating mechanism to achieve dynamic tuning of the outputs of multiple adapters based on the input information, thus enhancing the multi-task capability of a single model. Extensive experimental results have also proven that the above fine-tuning methods can fulfill the demands of multitasking

scenarios with limited computational resources [11–14].

However, despite the excellent performance of PEFT techniques in reducing the cost of fine-tuning and enhancing performance, their research in multimodal tasks is still in its early stages. There is a lack of an efficient and generalized PEFT paradigm that can flexibly cope with the complex structure of multimodal inputs while efficiently enhancing the performance of MM-LLMs in downstream tasks. Accordingly, the development of a cost-effective framework for multimodal model fine-tuning is particularly imperative. Such a framework not only needs to adapt to the complexity of multimodal tasks but also needs to lower the technical threshold to help non-AI professionals efficiently apply multimodal models, thus further promoting the application of multimodal technologies in real-world scenarios.

1.2. Research Objectives of Fine-tuning

To address those issues above, we deeply explore PEFT strategies for MM-LLMs and investigate several core topics. Specifically, we focus on evaluating the applicability of PEFT strategies originally designed for single-modality pre-trained models when faced with the complexity and diversity of multimodal tasks. In addition, we explore how to leverage the differences in modality between input information to improve the model's learning ability and generalization performance. Based on the above considerations, I innovatively propose the MMtuning framework - an advanced PEFT approach specifically tailored for building and fine-tuning MM-LLMs directly from pre-trained unimodal models.

As shown in Figure 1.1, MMtuning combines a parallel multi-adapter architecture with a fine-grained mechanism designed to construct a multimodal skill allocation matrix for multimodal tasks. The left half of the framework consists of four adapters, each with access to task-specific multimodal expertise, forming a diverse collection of the multimodal skill inventory. The right half integrates the Interactive Enhancement Block (IEB) and the Modal Weight Block (MWB). The synergistic interactions between IEB and MWB enable MMtuning to dynamically weight the outputs of the adapters based on the appropriate multimodal skill allocation matrix constructed by the input feature for optimal multimodal skill

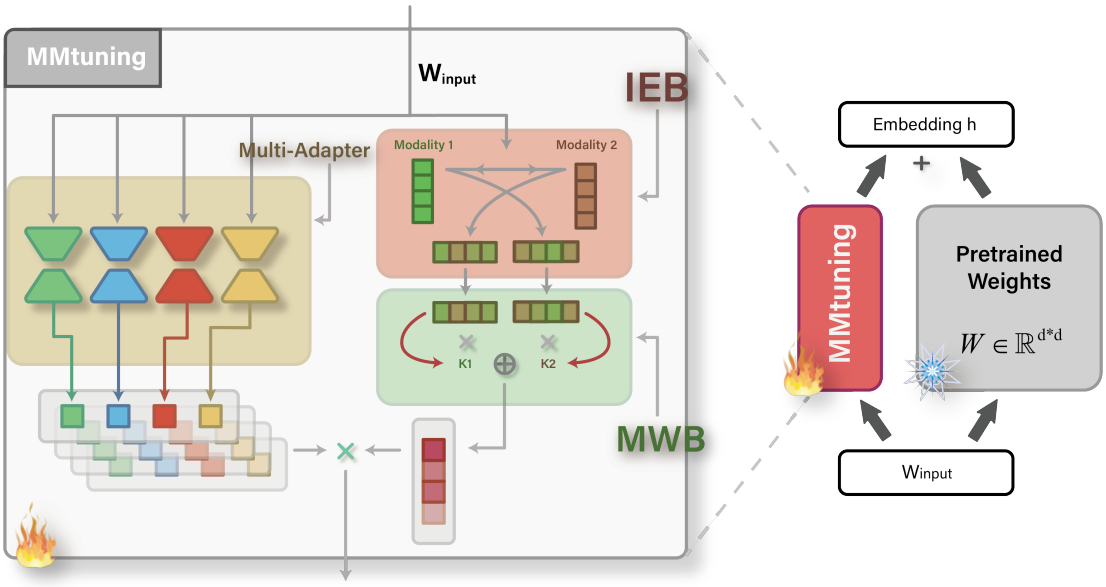


Figure 1.1 An overview of the MMtuning framework. In this framework, the left side features the Multi-Adapter block, which is illustrated in the figure as four LoRA adapters. The Interaction Enhancement Block (IEB) and the Modality Weights Block (MWB) are positioned on the right side and are responsible for processing modality-specific routing matrices. At the bottom, the four red squares in varying shades represent the multimodal skill allocation matrix.

subset. Disparate from the conventional PEFT methods, MMtuning effectively utilizes the similarity and heterogeneity between different modalities' input information and achieves precise regulation and adaptive alignment of adapter outputs. The experimental results reveal that MMtuning can not only efficiently integrate and utilize the rich information of multimodal input, but also maintain excellent parameter efficiency in the fine-tuning process, which highlights its flexibility and efficiency in handling multimodal tasks.

1.3. Research Objectives of Application

Moreover, Our goal is not only to develop an innovative fine-tuning framework for multimodal large-scale pre-trained models but also to help general designers or non-AI professionals overcome the knowledge and skill barriers in acquiring and applying multimodal deep learning models through the MMtuning framework. In the current era of rapid development of AI and deep learning, this framework aims to enhance the media literacy of these user groups while lowering the technical barriers they face when developing multimodal deep learning-related applications. To this end, based on the MMtuning framework, we build a user-friendly web platform designed to make it more convenient for non-deep learning experts (e.g., general designers or software engineers) to access and utilize visual-language multimodal models. The platform provides a more intuitive and accessible interactive interface than the traditional way of accessing visual-language multimodal models. Users upload their specific datasets and select the appropriate language processing and visual recognition models based on cost considerations and task requirements to obtain customized and fine-tuned visual-language multimodal models.

To validate the usability of this platform, we used the NASA Task Load Index (NASA-TLX) [15] to evaluate the workload of users in the process of acquiring visual-language multimodal models. The evaluation results show that the MMtuning web platform significantly reduces the workload and technical barriers for users compared to writing code directly or using other multimodal application methods. This reduced burden enables more users with non-deep learning backgrounds to efficiently apply multimodal models, promoting the broader adoption and accessibility of visual-language multimodal models. In summary, the MMtuning platform enables more users to seamlessly utilize multimodal deep learning models for their designs or applications by simplifying the model acquisition process, thus accelerating innovation and driving the widespread implementation of multimodal technologies in multiple fields.

1.4. Composition of the Thesis

The structure of this thesis is organized as follows:

Chapter 1 is the introduction, which focuses on the background of the study, the research objectives, and the overall structure of the thesis, and describes the current status of multimodal model fine-tuning and its applications. It focuses on the deficiencies in the current technology of multimodal model fine-tuning and the difficulty in acquiring multimodal models for people with non-AI backgrounds. The proposed solution to address the above issues in this research is also presented.

Chapter 2 reviews the related work on multimodal models and fine-tuning techniques, firstly, it introduces the concepts related to multimodal learning and large-scale multimodal models and discusses the challenges encountered in practical applications, and then it analyzes a variety of fine-tuning techniques, such as supervised fine-tuning, parameter-efficient fine-tuning, and other methods.

Chapter 3 describes the design and implementation of the MMtuning framework in detail, introducing the core modules of MMtuning V1, V2, V3 and their innovations, including the multimodal skill allocation matrix, Interaction Enhancement Block (IEB), the Modality Weights Block (MWB), etc. It also explores the design ideas of the MMtuning Web Platform as well as the specific user interface.

Chapter 4 shows the experimental setup, main results, and discussion. Through ablation experiments, the effectiveness of each Block in the MMtuning framework is evaluated. The performance of the MMtuning framework is verified through extensive and detailed comparison experiments. In addition to this, the role of MMtuning in modal integration and knowledge sharing is explored in depth, so that the performance of the model in downstream tasks can be substantially improved with limited resources. Finally, the usability of the MMtuning Web Platform is analyzed and verified from the results of specific user experiences (reflecting the reduction of technical barriers through the metric of workload).

Chapter 5 summarizes the main conclusions and contributions of the study, points out the limitations of the study, and proposes future research directions.

Chapter 2
Related Works

2.1. Multimodal Model

2.1.1 Multimodal Learning

In the field of machine learning, multimodal learning involves how to effectively characterize input information from different types and, based on this, achieve translation, alignment, and fusion between this heterogeneous information, ultimately leading to co-learning and processing of multiple input types [16]. The methodology humans perceive and acknowledge the world is multimodal, covering a variety of sensory channels such as vision, hearing, touch, smell, and taste. These different modes of perception can be seen as corresponding to different modalities of information input. Similarly, in most real-world research problems, multiple types of data are often involved. Therefore, for machines or AI systems to be primed with the ability to comprehend and interpret the world, they must have the ability to simultaneously process and align these diverse inputs, and thus understand the complex connections between these multimodal signals [17, 18]. Given this, research related to multimodal learning has become increasingly active and important in recent years. Research in this area is focused on developing algorithms that can synthesize and analyze data from different sources and types. Multimodal models are the achievements of such research efforts, as they are not only able to process multiple types of data inputs but also effectively integrate this information to provide a more comprehensive and in-depth understanding of expertise in various domains. However, given the inherent limitations of machines, which is the lack of a complete human perceptual system with a mature multimodal knowledge system, enabling machines to emulate human cognitive abilities through multimodal learning and to acquire a wide range of specialized

knowledge to cope with complex tasks in the real world is a challenging research topic [19]. In recent years, the rise of large language models (LLMs) with strong logical reasoning and knowledge comprehension capabilities has laid a solid foundation for the emergence of more robust and adaptive LLMs-based multimodal models and has also pushed multimodal AI to understand the world's knowledge from a holistic perspective and to demonstrate a more generalized ability to solve real-world problems [20].

2.1.2 Multimodal Large Language Models

Multimodal Large Language Models (MM-LLMs) [5] take the reasoning and decision-making abilities of large language models as the cognitive core and support various multimodal tasks by learning representations of different modalities. Alongside the size and capabilities of unimodal pre-trained models continuing to expand, integrating existing single-modal models into multimodal models, particularly leveraging powerful large language models(LLMs) as a foundation [3, 4, 21, 22], has increasingly become a prevalent approach. This method aims to reduce the computational cost of training a multimodal model from scratch and enhance the efficiency of multimodal pre-training. Despite numerous excellent unimodal models providing high-quality representations, essentially the underlying models of different modalities are usually pre-trained independently. Therefore, how to effectively facilitate LLMs alignment and understand multimodal information to collaborate with models from other modalities for inference remains a crucial challenge [23].

To address the above challenges, Blip2 utilizes the pre-trained Q-former module to bridge the gap between visual and textual modalities, facilitating LLMs in better leveraging the information from different modalities for inference. Flamingo [24] and Qwen-VL [4], transform the visual representations output by the visual encoder into fixed-size vectors containing a predetermined number of image tokens through a cross-attention module to achieve alignment between visual and textual space. In contrast, LLaVA [3] simply projects the visual features into the textual space, reaching the fusion of visual and textual information. However, all the methods mentioned above rely on vast pre-training datasets, complex pre-training steps, and high computational resource requirements. It remains a challenge to

effectively fine-tune the model to fit small-sized datasets with limited resources.

2.1.3 Multimodal modeling applications

Audio-Visual Speech Recognition (AVSR) [25] can be considered an early application in the field of multimodal learning. Its core goal is to enable models to integrate visual and auditory information to recognize the content of a speaker's words. It has been demonstrated that multimodal information inputs complement the final recognition performance of the model. Specifically, the information provided by visual modalities (lip-reading) and auditory modalities (speech signals) are mostly independent of each other, and each contributes significantly to the final recognition result [25–27]. It is inferred that rich and diverse input information can provide multiple perspectives to the model, thus enhancing its recognition capability and robustness. This also reveals the advantages of multimodal approaches over unimodal ones, especially in real-world tasks that require complex and diverse information, multimodal models can capture the essential features of the information more comprehensively, thus enhancing the accuracy and reliability of task execution.

Another important multimodal modeling application area focuses on the analysis and understanding of human behavior and habits. The most common applications in this area include multimodal recommender systems and multimodal sentiment analysis. Multimodal recommender systems predict the habits and preferences of users by collecting information from multiple dimensions of users, such as browsing history, purchase records, user reviews, and other sources of data, to achieve personalized recommendations [7]. Such systems have been widely used in a variety of fields such as health services, entertainment, personalized sports activity suggestions, personalized dietary information push, personalized motivational information, and so on. Meanwhile, multimodal information also provides a deeper understanding of sentiment analysis [28]. For example, multimodal sentiment analysis can play an important role in various scenarios such as mental health assessment, lie detection technology, interrogation situations, and the entertainment industry by integrating multiple data sources such as human facial expressions, body postures, voice features (pitch and timbre), textual information, and even physiological signals (electroencephalogram EEGs and heart rate varia-

tions). Nevertheless, in terms of ironic emotions (sarcastic speech) and complex emotion analysis (the inclusion of sadness in a person's happiness), which require context-specific analysis of emotions, current multimodal models of perception are still unable to reach the level of human subtlety of emotions [29].

In recent years, with the boost of Large Language Models (LLMs) with powerful reasoning and summarization capabilities, the field of Natural Language Processing (NLP) has gradually developed a new paradigm of utilizing LLMs with generalized capability to deal with various kinds of language-related problems. This shift has also influenced the development path of language-related multimodal applications [20]. Nowadays, many multimodal models tend to be constructed based on advanced language models or their architectures, thus shifting the focus of multimodal tasks from simple prediction and classification to more complex tasks [30], such as captioning, visual question answering (VQA), Speech summarization, and integrated image and text analysis. In this regard, OpenAI's visual language GPT-4 [21] and speech-to-text Whisper [31] are the most representative applications. These applications not only demonstrate the power of cross-modality information processing but also open up new possibilities for the development of multimodal models in web and mobile applications. Meanwhile, the in-depth research on image generation models, such as Diffusion Models [32] and Variational Auto-Encoders (VAEs) [33], has contributed to the emergence and development of popular Text-to-Image multimodal applications such as Stable Diffusion WebUI [34] and DALL-E [35]. These advances have not only enriched the application scope of multimodal learning but also laid a solid foundation for the development of more affluent AI applications in the future.

2.1.4 Challenges in Acquiring and Using Multimodal Models

Although multimodal models, especially multimodal large language models, have been widely used in many AI applications, the process of acquiring and using these models still faces the challenges of technical cost, high demand for computational resources, and large datasets. At the technical acquisition level, the current mainstream practice is to download pre-trained multimodal models from

GitHub or Hugging Face [36], the largest open platform in the world for machine learning models and datasets, and fine-tune them to fit the requirements of specific downstream tasks. However, this approach requires users to be proficient in programming Python-based deep learning frameworks such as PyTorch [37] or TensorFlow [38]. In addition, both running and fine-tuning these models require significant computational resources, which is a major obstacle for users who do not have high-performance computing facilities to support them. Moreover, the high-quality, multi-source datasets required for traditional methods of fine-tuning multimodal models are often large and complex in structure, which contributes to barriers to the acquisition and use of multimodal models.

2.2. Fine-tuning Technology

2.2.1 Supervised Fine-tuning and Instruction tuning

Numerous high-performance pre-trained models are emerging in the current field of Artificial Intelligence (AI). Pre-trained models are machine learning models that can effectively capture relevant expertise from large amounts of labeled and unlabeled data, with large model parameters as well as complex training objectives [39]. However, as datasets become progressively larger, deep learning is emerging as the algorithm with the strongest performance performance and robustness on large datasets. Current models such as BERT [40] and GPT [41] represent large-scale pre-trained architectures built upon deep learning principles. These models have demonstrated strong performance across a wide range of generalized tasks, including classification and prediction. However, to achieve optimal performance and adaptability for specific downstream tasks, fine-tuning these pre-trained models is often essential.

To avoid the cost of re-training a new model from scratch for a new specific task, supervised fine-tuning (SFT) of existing pre-trained models has become an important technical aspect in applying deep learning models [42–45]. Supervised fine-tuning (SFT) is a method for adapting a pre-trained large-scale model to a specific downstream task by leveraging labeled data [46]. The core of the supervised fine-tuning technique lies in the further training of an existing large-scale

pre-trained model by utilizing labeled datasets so that it can be better adapted to a specific downstream task or domain of expertise [47]. This process not only strengthens the capability of the model to understand and perform a specific task but also enhances its generalization performance in the face of new data [48]. In a word, supervised fine-tuning is to introduce domain-specific knowledge and task-related annotations on top of the existing generalized knowledge of the pre-trained model, so that the model can show higher accuracy and better adaptability to specific tasks, thus realizing significant performance improvement.

In the current context of the flourishing development of pre-trained Large Language Models (LLMs) and multimodal LLMs, Supervised Finetuning (SFT) techniques play an indispensable role in adapting these models to new tasks or learning domain-specific knowledge [42, 44, 46]. Initially, SFT was mainly used to align pre-trained large language models with downstream domain tasks, such as code generation [49]. Subsequently, with the continuous development of the multimodal field, especially multimodal large language models (MM-LLMs), and improvement of corresponding VQA benchmarks for training and evaluation [50, 51], the instruction tuning method derived from SFT has been more widely used. In particular, Instruction Tuning [42, 52] has become a mainstream practice in this field. It utilizes datasets containing natural language instructions to guide pre-trained large language models in learning new knowledge thus improving their task-specific performance. In this process, the instruction-output pair dataset becomes the central resource, where the instruction part represents the specific requirements of humans. In contrast, the output part defines the response that the model should produce in response to these instructions.

The main reason why Instruction Tuning has been widely used in the practice of fine-tuning LLMs and MM-LLMs is that it can effectively constrain the model's outputs to ensure that their response properties or domain knowledge are consistent with the desired goals. Additionally, with the powerful language and reasoning capabilities of large language models, instruction tuning MM-LLMs on labeled datasets by following human instructions to output corresponding content [53]. This provides humans with an effective means to adjust the behavior of models but also makes the model behave in a more controllable and highly predictable manner when performing tasks. Moreover, since the input format of the

command fine-tuning is essentially the same as the input and output formats used in the pre-training phase, it facilitates the rapid adaptation of the LLMs to new application scenarios without any changes to the model structure, thus increasing the model's usefulness and efficiency while maintaining flexibility. Currently, instruction tuning has become the main strategy for fine-tuning MM-LLMs, such as InstructBLIP [54], PaLI-3 [55], and Video-LLaVA [56], which have achieved competitive performance on many visual-language tasks. In summary, by training on a diverse instruction fine-tuning dataset, this method significantly improves the interactivity of multimodal models and the performance in few-shot situations.

However, in the current research field, due to the inherent complexity of modal alignment and multimodal information processing, it is notable that the main fine-tuning strategies typically require large instruction datasets, complex fine-tuning processes, and costly computational resources to achieve optimal generalization performance [52]. For example, the InstructBLIP [54] model collects and integrates a large number of instructional datasets, and only performs instruction tuning on the Qformer component of BLIP2 [2] which has a relatively small number of parameters while freezing the weights of the visual and language models. Despite this, due to the powerful pre-trained MLLMs and rich and diverse datasets, the model still shows generalization capabilities in visual-language tasks. In addition, the X-VILA [57], LLava [3], and Qwen-VL [4] models choose a supervised instruction fine-tuning strategy that only freezes the multimodal connector and large language model, which undoubtedly increases the demand for computational resources. All of the above methods are based on complex pre-trained multi-language large models and then further instruction tuning. From the above, instruction fine-tuning still poses a significant economic and technical barrier for general AI application developers or small businesses. The high-cost investment includes acquiring high-quality and diverse labeled data and the computational overhead of implementing the fine-tuning process, such as the use of GPU clusters. The combination of these factors makes many potential user groups face unbearable cost pressure in obtaining efficient models related to large languages. Therefore, reducing the cost burden of fine-tuning has become a pressing issue in current research and practice.

2.2.2 Parameter-Efficient Fine-tuning

To reduce the cost of fine-tuning large-scale Transformer-based models for downstream tasks, Parameter-Efficient Fine-tuning (PEFT) methods gained widespread recognition and application. PEFT methods achieve efficient fine-tuning by introducing small, trainable Adapters within existing layers of models, with the core principle being to freeze the parameters of the base model and update only the Adapter weights. Initially, Houlsby et al [58] introduced adapters into BERT for fine-tuning NLP tasks. Subsequently, diverse variants of adapters have emerged and applied to different components of the Transformer-based architecture. Among these, Low-Rank Adapters (LoRA) [8] is one of the most widely used Adapter schemes.LoRA strives to enhance parameter efficiency by learning the weights of Low-Rank decomposition matrices, allowing for concurrent fine-tuning of the model with the corresponding dominant matrices without increasing inference costs. Building upon LoRA, Liu et al [9] proposed $(IA)^3$, which scales on the activation layers by a learnable vector, functioning as an adapter, to realign internal activation for specific downstream tasks, enabling fine-tuning of the base model. Compared to LoRA, $(IA)^3$ achieves effective fine-tuning with fewer parameters and avoids the computational overhead of parallelization. Additionally, methods such as Prompt tuning [59], P-tuning [60], Prefix tuning [61], and P-tuning2 [62] facilitate efficient fine-tuning by appending a fixed number of learnable prompt tokens, essentially trainable vectors, into the inputs of each model layer. Based on these soft prompt token approaches, learnable vector parameters serve as task-specific prompts customized for the downstream task.

In the context of PEFT in multi-task learning (MTL), MoE-based Multi-Adapter structures are widely adopted. For example, MultiLoRA [63] utilizes multiple LoRA instances running in parallel as a Multi-Adapter method, models can strike a balance in the contribution of each adapter ' s subspace. Experimental results reveal that multiple LoRAs outperform a single LoRA with an equal rank in multiple datasets and tasks. In addition, Poly [12] builds a task-related sparse routing matrix based on a prior Skill Inventory. This matrix selects and combines a subset of adapters for each task and processes input information through these adapters. In this way, the adapter can dynamically process input information for different task requirements. Based on Poly's study, MHR [64], OrchMoE [14] and

C-Poly [13] optimize the routing matrix which improves the extraction of task-related expertise in Multi-Adapter, resulting in superior adaptability to the needs of multi-task learning. Recently, MoLoRA [11] successfully utilized the Mixture of Experts (MoE) architecture within the PEFT paradigm. The routing matrix manages the output of a single lightweight adapter, enabling joint learning between different adapters to better accommodate multi-task scenarios. However, the study of PEFT methods for multimodal tasks is at an early stage of exploration and requires further research.

In addition, a series of PEFT methods have been developed for pre-trained multimodal models such as BLIP [65], CLIP [66], All in one [67], and CLIPBert [68]. These methods include Aurora [69], UniAdapter [70], and MAPLE [71], which are capable of fine-tuning pre-trained multimodal models to exhibit robust performance in specific subtasks or new tasks under resource-constrained conditions. However, firstly, these methods are not designed for large multimodal language models and do not consider how to fine-tune multimodal models more effectively with huge parameter amounts and powerful language models. Secondly, a central premise of these fine-tuning methods is the reliance on multimodal models that have been pre-trained on a large scale. As the number of input modalities increases, the difficulty of obtaining multimodal models that have been pre-trained on a large amount of data increases, which may result in existing fine-tuning methods no longer being applicable when confronted without pre-trained multimodal models in the future.

2.2.3 Mixture of Experts

In the domains of NLP and CV, the Mixture of Experts (MoE) architecture has achieved widespread recognition as an effective strategy to enhance the parameter capacity of pre-trained large-scale models. [72–77]. The MoE architecture consists of a set of expert networks, each of which can learn different expertise and a gated network whose task is to assign the appropriate expertise in response to the input information to obtain the appropriate output. Much of the research related to MoE architectures has focused on optimizing the routing mechanism to more proficiently capture different proportions of expert knowledge, thereby improving the robustness and generalization of the model. Examples include routing via

stochastic activation of a single expert [78], sparse routing by selecting the top-k most prominent experts [11,73,77], and routing by performing a weighted average of all activated experts [79].

As discussed in Section 2.2.2, the Mixture of Experts (MoE) architecture has been widely used in the field of multi-task PEFT and has demonstrated excellent performance in enabling a single large language model to handle multi-task fine-tuning scenarios. The MoE architecture improves the capacity and flexibility of the model by dynamically assigning expert modules to handle different sub-tasks without significantly increasing the computational burden. This characteristics makes MoE perform particularly excellent in multi-task learning. However, when introducing the MoE architecture in multimodal tasks, the key challenge is how to effectively fuse information from different modalities and ensure efficient collaboration among the expert modules. One of the main tasks of this study is to explore the applicability of the MoE architecture and its potential limitations in the field of multimodal PEFT. Then, manipulated the in-depth research and innovative work based on it. To this end, our research focuses on addressing the aforementioned challenges to develop a novel fine-tuning framework that can both capitalize on the advantages of the MoE architecture and overcome its challenges for multiple modal inputs. Ultimately, we have designed the optimal deep learning fine-tuning framework for multimodal large language models, named MMtuning.

MMtuning is a MoE-like framework. In it, we adopted a multimodal skill allocation matrix (akin to a routing matrix) to integrate Multi-Adapters' multimodal expertise output. Unlike previous methodologies [72, 75] that utilized distinct experts at the token level, MMtuning operates at the sample level. Given that each adapter contains expertise from different modalities, the multimodal skill allocation matrix activates all adapter outputs in different proportions.

Chapter 3
Methodology

3.1. MMtuning V1

3.1.1 Multimodal Skills Inventory and Allocation Matrix

In previous research on efficient multi-task parameter fine-tuning [12–14, 64], the concepts of " skill inventory (Φ) " and " skill allocation matrix(W_{skill}) " were introduced to build a flexible and efficient fine-tuning framework. Those frameworks aim to dynamically combine the optimal skill sets from the skill inventory Φ through the skill allocation matrix W_{skill} according to the requirements of different tasks. It is worth noting that the number of skills N in the skill inventory is smaller than the number of tasks T ($N < T$), which allows multiple tasks to share some common skills, thereby achieving efficient parameter reuse. Through this approach, the model can effectively capture and reuse the common skills shared across tasks, while flexibly combining and applying exclusive skills according to the needs of specific tasks T_i, thus improving the overall performance and efficiency of multi-task fine-tuning.

I have extended the concept above to multimodal parameter-efficient fine-tuning and introduced a more fine-grained mechanism to control skill allocation in multimodal tasks:

In multimodal tasks, particularly in scenarios involving multimodal large language models (MM-LLMs) such as Visual Question Answering (VQA), the model needs to dynamically combine different modalities' information based on varying input queries to generate appropriate outputs. Consequently, different inputs x in multimodal tasks often necessitate distinct combinations of multimodal skills. However, the semantic or feature differences between inputs can vary significantly, ranging from highly similar to completely different. For instance, the semantic

distance between two questions related to the same image and contextual description is relatively small, whereas the difference between two questions based on entirely different images and textual inputs is much more pronounced. Therefore, the model must accurately discern the degree of variation between inputs x, as this directly determines the diversity and relevance of the skill sets required for generating outputs.

To address this challenge, the model first learns a comprehensive multimodal skill inventory Φ, which contains the fundamental skills necessary for performing a specific multimodal task. Subsequently, for each input sample x_i, the model needs to extract a subset of multimodal skills relevant to that input from the skill inventory Φ, ensuring that similar inputs share comparable skill sets. This strategy has been validated through the clustering analysis discussed in Section 3.c, demonstrating its effectiveness in extracting input-relevant skill sets.

However, this similarity-based skill extraction approach remains coarse-grained, as it does not fully account for the inherent heterogeneity between modalities and the subtle semantic differences among inputs in multimodal tasks. For example, even within the same image and contextual description, processing different human instructions may require significantly different multimodal skills. To further refine the skill set, I have designed a multimodal skill allocation mechanism that incorporates a skill allocation matrix W_{skill}. This matrix learns specific skill weights for each input modality, thereby enhancing the precision of the multimodal skill sets extracted from the skill inventory Φ.

Specifically, the skill allocation matrix W_{skill} dynamically optimizes and refines the most suitable subset of multimodal skills based on the shared skill sets derived from similar inputs. This mechanism ensures that the model can flexibly adjust its skill sets to accommodate the unique characteristics of each input x_i. Ultimately, the optimization objective is to maximize the log-likelihood over the dataset D by identifying the most relevant skills for each input 3.1, thereby improving the overall performance and generalization capability of the multimodal task.

$$\sum_{(x_i,y_i)\in D} \log p(y_i \mid x_i, W_{\text{skill}}, \Phi) p(W_{\text{skill}} \mid x_i) p(\Phi \mid x_i) \tag{3.1}$$

To achieve the objective outlined above, I conducted extensive research and experimentation. As discussed in Chapter 2.2.2, in the context of Parameter-

Efficient Fine-Tuning (PEFT) for Multi-Task Learning (MTL), the Mixture of Experts (MoE) architecture has demonstrated significant effectiveness in acquiring diverse domain-specific knowledge for various downstream tasks. Drawing inspiration from these findings, I hypothesized that the MoE architecture could also be well-suited for dynamically extracting multimodal domain-specific skill inventory Φ and allocating different skill sets for output from multimodal data. This framework enables the effective application of extracted knowledge to diverse and complex multimodal tasks, demonstrating superior performance and adaptability [80]. After a comprehensive comparative analysis and thorough investigation, I designed the MMtuning V1 framework, which leverages the strengths of the MoE architecture to efficiently learn comprehensive skill inventory Φ and corresponding skill sets from a specific multimodal task.

3.1.2 Parameter efficient Dynamical Multimodal Skill Inventory

In prior research, the learnable adapter module has been widely utilized to efficiently extract domain-specific skill inventory Φ from large-scale datasets, thereby enhancing model performance across diverse input scenarios [11–14, 63, 64]. To construct a comprehensive multimodal skill inventory Φ on a multimodal dataset D and dynamically extract appropriate multimodal skill sets based on varying inputs x_i, I have designed a Multi-Adapter Block. These learnable adapters are capable of learning multimodal skills specific to the multimodal dataset D. Moreover, the adapters can dynamically extract the most relevant multimodal skill sets according to the input x_i, ensuring the model's adaptability to different input contexts.

Furthermore, to facilitate efficient fine-tuning of parameters on multimodal large language models (MM-LMMs), the MMtuning framework incorporates the Low-Rank adapter (LoRA) [8] techniques in the Multi-Adapter Block. The LoRA instantiates capable tuning of the model to specific downstream task requirements without significantly increasing the model complexity. In the experiments where each adapter is implemented as a low-rank adapter. Although theoretically, all adapters in MMtuning can be replaced by other parameter-efficient adapters such

as LoHa [10], DyLoRA [81] or AdaLora [82] for building multimodal skill inventory. However, LoRA was chosen as the adapter of choice in this study due to its powerful ability to extract knowledge and its flexibility to seamlessly integrate into the various components of the Transformer architecture, such as query, key, value, and feedforward layers.

In simple terms, the technique of LoRA involves injecting a pair of low-rank matrices (B and A) into a linear transformation in the Transformer architecture. While freezing the linear transformation reduces the computational overhead during training, it also ensures that the input and output dimensions are consistent with those of the linear transformation. The linear mapping of LoRA ($\mathbb{R}^d \rightarrow \mathbb{R}^d$) can be expressed as Equation3.2:

$$h = W_0 + \Delta W x = W_0 + BAx \tag{3.2}$$

The $\Delta W \in \mathbb{R}^{d \times d}$ can be replaced by two low-rank matrices: $B \in \mathbb{R}^{d \times r}$ and $A \in \mathbb{R}^{r \times d}$ which obtained by gradient descent optimization. Due to $r \ll d$, the update computation for each linear layer in the model involves only the $2 \times r \times d$ parameters of the Low-Rank Adapter, which is significantly fewer than the $d \times d$ parameters otherwise required. This substantially improves parameter efficiency, enabling rapid training even with limited computational resources. In our experiments, we replaced all query, key, and value layers with the MMtuning framework, which combines multiple LoRAs into a Multi-Adapter for dynamically extracting multimodal skill inventory.

3.1.3 The Multimodal Skill Allocation Matrix

In MoE architectures, a robust routing matrix (allocation Matrix) plays a pivotal role in controlling the outputs of multiple adapters to provide appropriate information for downstream tasks [83]. This is because an effective routing matrix in the MoE architecture can dynamically assign multiple experts (i.e., Multi-Adapter block) at different depths of the model, thus achieving optimal output distribution. As a result, when dealing with multimodal inputs, this mechanism can superior capture the common features (i.e., mutual information) [84] of each modality and integrate the complementary information of multiple modalities, thus enabling

the fusion of information across modalities and significantly improving the overall performance of the model. Moreover, the LIMoE [85] and HPT [86] models underscore the significance of distinguishing different modality information in the token dimension and processing it in multimodal tasks.

Therefore, to bolster the adaptability in multimodal tasks, MMtuning V1 first adopts the strategy of separating the input information of different modalities along the token dimension. This approach aims to construct a robust multimodal skill allocation matrix W_{skill} to fine-grained extract the optimal subset of multimodal skill sets from the Multi-Adapter block that is highly tailored to the input. The core goal of this matrix is to enable the fusion of information across modalities to enhance the model's understanding of the input x_i and thereby achieve optimal skill sets. In the following section, the principles and algorithmic details of constructing the multimodal skill allocation matrix in detailed MMtuning V1 are elaborated.

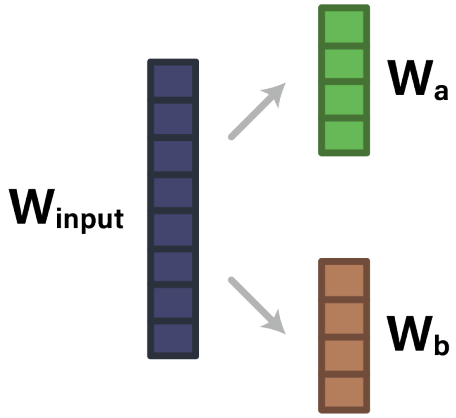


Figure 3.1 A schematic drawing of separating input information according to different modalities into W_a and W_b in the token dimension.

In this study, I explore a novel approach to process the multimodal input x for constructing the skill allocation matrix, which is realized by utilizing well-designed segmentation and interaction algorithms for the processing of input information from different modalities. Specifically, as shown in Figure 3.1, the input information $W_{\text{input}} = \{x_1, x_2, \ldots, x_{token}\} \in (\mathbb{R}^{dk})^{token}$ will be split to distinct routing matrices W_a and W_b, according to different modalities. W_a and W_b represent the modality-specific routing matrices respectively, with dimensions as shown in

Equation3.3 ($i + j = token$). Through this method, we give a fixed location for the different modality inputs at the token level, which can define the boundary between modalities. The dimensionality of this space is designed to allow MMtuning to segregate the input information according to different routing strategies, which in turn lays the foundation for subsequent modality information integration. Subsequently, I designed the Interaction Enhancement Block (IEB) and the Modality Weights Block (MWB) process, which provide in-depth processing and integration of modality-specific routing matrices. The essential goal of these two blocks is to enable the model to learn both the similarity and heterogeneity among modality information. Eventually, the multimodal skill allocation matrix W_{skill} constructed based on these two modality-specific routing matrices, will optimally align with the multimodal skill sets from the Multi-Adapter Block derived from input x_i. The following sections comprehensively overview IEB and MWB principles and functions.

$$W_a = \begin{bmatrix} \mathbf{w}_{a,1}^\top \\ \mathbf{w}_{a,2}^\top \\ \vdots \\ \mathbf{w}_{a,i}^\top \end{bmatrix} \in \mathbb{R}^{i \times dk} \quad W_b = \begin{bmatrix} \mathbf{w}_{b,1}^\top \\ \mathbf{w}_{b,2}^\top \\ \vdots \\ \mathbf{w}_{b,j}^\top \end{bmatrix} \in \mathbb{R}^{j \times dk} \tag{3.3}$$

3.1.4 Interaction Enhancement Block

Implementation of attention mechanisms significantly enhances the ability of models to discern critical information and extends its applicability to integrating information across diverse domains and tasks [87–91]. In the MMtuning framework to superior align information from different modalities, the Interaction Enhancement Block (IEB) utilizes a pattern similar to the Attention Mechanism to facilitate the model of learning the similarities between modality-specific routing matrices. This block enables each modality-specific routing matrix to selectively acquire and integrate significant information or expertise from other routing matrices, thereby enriching their respective original information.

Specifically, as shown in Equation 3.4, each modality-specific routing matrix ($W_a$and W_b) captures the inter-modal similarity or correlation by calculating the inner product between it and the target modality-specific routing matrix

(e.g., $\sum_{m=1}^{dk} \mathbf{w}_{a,k}^{m} \mathbf{w}_{b,l}^{m}$). Then the matrix after the inner product and the target modality-specific routing matrix are calculated again to get the information after interaction(e.g.,$\Delta W_a = \sum_{l=1}^{j} \sum_{m=1}^{dk} \mathbf{w}_{a,k}^{m} \mathbf{w}_{b,l}^{m} \mathbf{w}_{b,li}$), which is ΔW_a and ΔW_b ($\Delta W_a \in \mathbb{R}^{i \times dk}$ and $\Delta W_b \in \mathbb{R}^{j \times dk}$). Subsequently, the matrix after the interaction ($\Delta W_a$and ΔW_b) will be added to the original matrix ($W_a$and W_b) to form $W_{a_{\text{updated}}}$ and $W_{b_{\text{updated}}}$ ($W_{a_{\text{updated}}} \in \mathbb{R}^{i \times dk}$ and $W_{b_{\text{updated}}} \in \mathbb{R}^{j \times dk}$).

$$
\begin{aligned}
W_{a_{\text{updated}},ki} &= W_{a,ki} + \sum_{l=1}^{j} \left(\sum_{m=1}^{dk} \mathbf{w}_{a,k}^{m} \mathbf{w}_{b,l}^{m} \right) \mathbf{w}_{b,li} \\
W_{b_{\text{updated}},li} &= W_{b,li} + \sum_{k=1}^{i} \left(\sum_{m=1}^{dk} \mathbf{w}_{b,l}^{m} \mathbf{w}_{a,k}^{m} \right) \mathbf{w}_{a,ki}
\end{aligned}
\tag{3.4}
$$

The essential purpose of the IEB design is to calculate and integrate the mutual information between various modalities [92], that is, to assess the shared information among different modalities when processing the same task. As shown in Figure 3.2, this process allows each modality-specific routing matrix to contain information unique to that modality and incorporate mutual information that is important and relevant to other modalities. This information fusion design strategy enriches the task-related information contained in the original modality-specific routing matrix, thereby enabling a more precise distribution of weights when constructing a multimodal skill allocation matrix, to achieve optimal information processing results. Ultimately, the IEB achieves cross-modality information fusion and knowledge sharing so that the model can effectively capture the similarities and align the modality-specific routing matrices from different modality inputs.

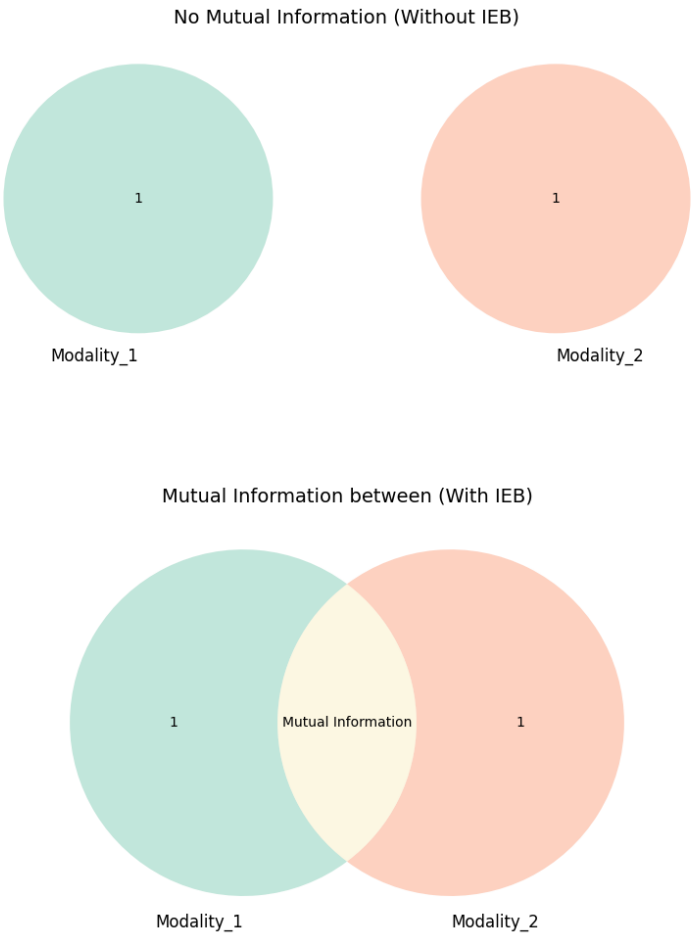


Figure 3.2 Mutual Information Schematic: Demonstrates the computation and interaction process of sharing information between different modalities through IEB (Interaction Enhancement Block).

3.1.5 Modality Weights Block

In a multimodal input, each modality brings affluent, task-specific mutual information. However, the unique data structure of each modality leads to the fact that those different modality inputs are heterogeneous in nature and are expressed through their independent encoding. As a result, a certain amount of heterogeneity between different modalities is inevitable. Based on this analysis, given that $W_{a_{\text{updated}}}$ and $W_{b_{\text{updated}}}$ inherently contain distinct numerical distributions due to

different modalities, the Modality Weights Block (MWB) utilizes this difference in inter-modal information to formulate a rule for building the final multimodal skill allocation matrix W_{skill}.

First, to reduce the complexity of the model computation and improve the comprehensibility of input samples from diverse modalities, I opted to eliminate the token dimension for $W_{a_{\text{updated}}}$ and $W_{b_{\text{updated}}}$. Consequently, MMtuning will exclusively regulate the Multi-Adapter Block's output, contingent upon sample-level information. Specifically, I get $W_{a_{\text{avg}}}$ and $W_{b_{\text{avg}}}$ by averaging the token dimension, as shown in Equation 3.5 ($W_{a_{\text{avg}}} \in \mathbb{R}^{dk}$ and $W_{b_{\text{avg}}} \in \mathbb{R}^{dk}$).

$$W^m_{A_{\text{avg}}} = \frac{1}{i}\sum_{p=1}^{i} W^m_{a_{\text{updated}},p} \quad W^m_{B_{\text{avg}}} = \frac{1}{j}\sum_{l=1}^{j} W^m_{b_{\text{updated}},l} \tag{3.5}$$

I hypothesize that the amount of information contained in each modality-specific routing matrix is represented by the degree of numerical discretization, and accordingly, different modality-specific routing matrices should provide a disproportionate contribution to the construction of the final multimodal skill allocation matrix W_{skill} based on their amount of information. As shown in the Equation 3.6, we take the standard deviation of the last dimension of $W_{a_{\text{avg}}}$ and $W_{b_{\text{avg}}}$ as a measure of the information content. Then the scalar weights (K_i) of the modality-specific routing matrices under different modalities are obtained by Equation 3.7. This represents the relative contribution of each modality-specific routing matrix in the final multimodal skill allocation matrix W_{skill}. With this approach, the MWB identifies heterogeneity between modalities by evaluating the differences in the distribution of information contained in each routing matrix.

$$\begin{aligned} \mu_a = \frac{1}{dk}\sum_{m=1}^{dk} W^m_{a_{\text{avg}}} \quad \mu_b = \frac{1}{dk}\sum_{m=1}^{dk} W^m_{b_{\text{avg}}} \\ W_{a_{\text{std}}} = \sqrt{\frac{1}{dk}\sum_{m=1}^{dk}\left(W^m_{a_{\text{avg}}} - \mu_a\right)^2} \\ W_{b_{\text{std}}} = \sqrt{\frac{1}{dk}\sum_{m=1}^{dk}\left(W^m_{b_{\text{avg}}} - \mu_b\right)^2} \end{aligned} \tag{3.6}$$

A temperature coefficient t [93] is incorporated into the weight output function

3.7 to mitigate the risk of the multimodal skill allocation matrix overly relying on a single modality during early training. This approach ensures balanced attention to the routing matrices of different modalities, preventing excessive focus on any single modality-specific routing matrix results.

$$\begin{aligned} \text{std}_{concat} &= [W_{a_{\text{std}}}, W_{b_{\text{std}}})] \\ K_i &= \frac{\exp\left(\frac{\text{std}_{concat,l}}{t}\right)}{\sum_{j=1}^{2} \exp\left(\frac{\text{std}_{concat,j}}{t}\right)}, \quad i \in \{A, B\} \end{aligned} \tag{3.7}$$

Subsequently, as shown in the Equation 3.8, firstly map the last dimension of each modality-specific routing matrix to the number of adapters ($dk \rightarrow N_{adapters}$) by applying two learnable matrices M_A and M_B ($M_i \in \mathbb{R}^{dk \times n = N_{\text{adapters}}}, i \in \{A, B\}$). Then, a weighted sum with the scalar weights K_A and K_B ($K_A = K_1, K_B = K_2$) is performed to obtain the final multimodal skill allocation matrix $W_{\text{skill}} \in \mathbb{R}^{n=N_{\text{adapters}}}$.

$$W_{\text{skill},n} = K_A \cdot \left(\sum_{m=1}^{dk} M_{A,n}^{m} W_{a_{\text{avg}}}^{m} \right) + K_B \cdot \left(\sum_{m=1}^{dk} M_{B,n}^{m} W_{b_{\text{avg}}}^{m} \right) \tag{3.8}$$

Through MWB, MMtuning can dynamically construct an appropriate multimodal skill allocation matrix according to the input of two modality-specific routing matrices, which contain information about different modalities split in the W_{input} token dimension. This approach enables MMtuning to optimally distribute attention across modalities and select the most pertinent skill set from the shared multimodal skill inventory Φ, improving its robustness and generalization across various multimodal task processing. Finally, the multimodal skill allocation matrix W_{skill} is used to adjust the coarse-grained skill sets to the fine-grained multimodal skill set from the Multi-Adapter Block $A_i(W_{\text{input}})$, as shown in Equation3.9, which enables concurrent learning between adapters and allocation matrices. In addition, the approach described above for obtaining the multimodal skill allocation matrix is not limited to the case of inputs with only two modalities in the methodology. Theoretically, the framework can be extended to the task of three or more modalities of input information.

$$y = \sum_{i=1}^{n} (W_{\text{skill},i} \cdot A_i(W_{\text{input}})) \tag{3.9}$$

3.2. MMtuning V2

Although MMtuning V1 shows prominent advantages in the Encoder-Decoder architecture for multimodal large language models (MM-LLMs), which is effective in achieving modality alignment, information fusion, and optimal output in resource-limited situations verified by experiment analysis in Chapter 4, the framework suffers from a notable limitation: it is only applicable to encoder-decoder Language model architectures.

The core of MMtuning V1's mechanism is constructing an optimal multimodal skill allocation matrix W_{skill} based on input x_i for acquiring an optimal multimodal skill set from skill inventory Φ. In the encoder part of the LLMs, MMtuning V1 can efficiently handle the modal alignment problem, achieve information fusion, and produce optimal output results. However, for LLMs that contain Decoder-Only, the model inputs the final predicted labels along with the multimodal input information during processing. This results in MMtuning V1 being able to anticipate future predictions when processing the input information, which may affect the model's outputting accuracy. To address the limitations of MMtuning V1 and broaden its applicability to a wider range of multimodal large language models, including purely decoder-based architectures, we have conducted a comprehensive investigation and devised an innovative approach. This refined framework, MMtuning V2, aims to enhance the usability and utility of MMtuning in both research and practical applications within the field of multimodal large language modeling.

3.2.1 Mask mechanism of Decoder-only LLMs

In the decoder-only Transformer LLMs architecture, the masking mechanism is the pivot technique to achieve regressive prediction [88]. This architecture is mainly used in language models, such as the GPT family [21] and OPT [94], where the core goal is to predict the next possible occurrence of a word or token

based on a given sequence of inputs. In the training phase, the input to the model consists of a sequence of tokens, which are converted into embedding vectors by an embedding layer. At the same time, the label of the model is fed to the model at the same time along with the input sequence. The output of the model is the probability distribution of the next token, which is usually compared with the real next token of the next label to compute the loss and optimize the weight parameters of the original model by backpropagation. The role of the masking mechanism in the decoder-only Transformer is to prevent future tokens from being seen when predicting the current token. This is accomplished by creating a mask matrix that adds a very large negative number (e.g., $-1e9$) to the positions of the label markers that represent future information during the current prediction step to ensure that the attention weights at these positions are nearly zero before computing the softmax function. This design ensures that the model does not see future information when predicting each label tag, but only relies on previous tags, thus realizing the regressive property.

Only the decoder model exhibits different behavioral patterns in the training and inference phases. The training phase allows the model to process the entire sequence in parallel since the prediction of each token depends on the information of all previous tokens in the sequence. In other words, the training is done by predicting the entire token of the label in one operation, which also makes the training process more efficient by paralleling the computation. However, in the inference phase, the model must generate the markers serially because only previously generated markers can be observed at a time. This regressive inference process is serial and cannot be parallelized. At the initial stage of inference, the input to the model consists of the entire context sequence as well as a special starting token (e.g., $< bos >$). When predicting a second word, the input to the model will include the word expected in the first round as well as the original sequence. As the inference process proceeds, newly predicted words will be successively added to the input sequence to predict subsequent words. This regressive inference mechanism ensures that the model can construct the output sequence incrementally but it also limits its ability to generate outputs in parallel.

3.2.2 Mask mechanism of MMtuning V2

To realize the efficient fine-tuning of the MMtuning framework in the multimodal large language model composed of Decoder-Only LLMs, and to adapt to its special input and processing mechanisms, I have conducted meticulous research and experiments. Finally, a suitable masking technique for the MMtuning architecture was designed, leading to the creation of MMtuning V2. MMtuning V2 was designed to introduce a suitable masking technique while maintaining the same modular and architectural compatibility as MMtuning V1. This technique effectively masks access to future prediction information during processing, thus ensuring the accuracy and validity of model predictions.

Specifically, I initially transformed multimodal input information into a standardized format suitable for decoder-only large language models (LLMs). **Length-varying inputs V_1, e.g., text**: Given that the length of input information varies significantly across samples, the length of textual questioning and descriptive information accompanying questions for different images may also show inconsistency. For this reason, we adopt a normalization strategy to standardize the length of all input texts to a fixed number of tokens, denoted as T_1. For inputs with lengths less than T_1, we fill them with special tokens $< pad >$ to ensure that their lengths are up to T_1, while inputs with lengths exceeding T_1 are truncated to ensure that their lengths are not exceeding T_1. T_1 is determined based on a statistical analysis of the lengths of all sample texts in the input dataset, and the lengths of all sample texts in the input dataset are determined based on a statistical analysis of the lengths of all sample texts. The upper quartile of the text length was selected as the value of T_1 based on a statistical analysis of the text lengths of all samples. This approach is designed to ensure that the value of T_1 remains stable and is not unduly influenced by a small number of excessively long texts while maintaining consistency and efficiency in model input processing. **Length-constant inputs V_2, e.g., image**, leaving the original input unchanged without any modification and recording its corresponding length as T_2. **Label V_3**: Let them as they are without any padding or truncation, regardless of whether or not the data is mutable. Eventually, we will splice all inputs in the order of V_1, V_2, V_3 to form a complete multimodal input $W_{\text{input}} = \{x_1, \ldots, x_{T_1}, \ldots, x_{T_2}, \ldots, x_{T_3}\} \in (\mathbb{R}^{dk})^{T_1+T_2+T_3}$ to be fed to the Decoder-

only language model for processing. After determining the position of the individual modalities and labels in the input, the masking operation can be performed more explicitly. The mask mechanism schematic diagram of MMtuning V2 is shown below 3.3.

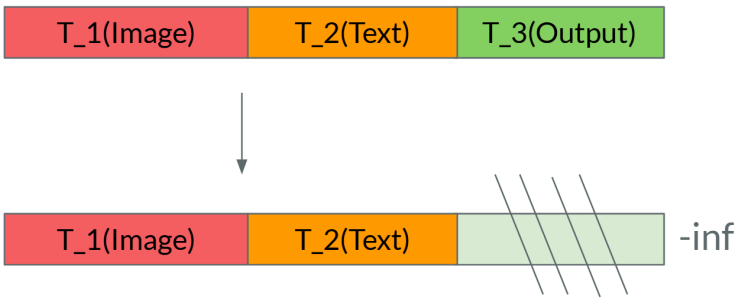


Figure 3.3 A schematic diagram of the masking mechanism in MMtuning V2 is shown, where T_1, T_2, T_3 correspond to the token positions of the image, text, and output labels, respectively.

In the MMtuning V2 framework, the masking mechanism is a curial innovation that centers on two main components. Specifically, as shown in the figure 3.4, within the Input Encoding Block (IEB), the vector values of all tokens where the label is located in the input vector are reset to zero ($W_{\text{label}} = \{x_{T_2+1}, \ldots, x_{T_3}\} = 0$), based on the exact individual modalities and the position of the label in the input. This strategy effectively eliminates the label position's attention weight when performing the attention computation of the dot product. Then, the routing matrices specialized for each modality cannot obtain future information on the position of the label for other modalities.

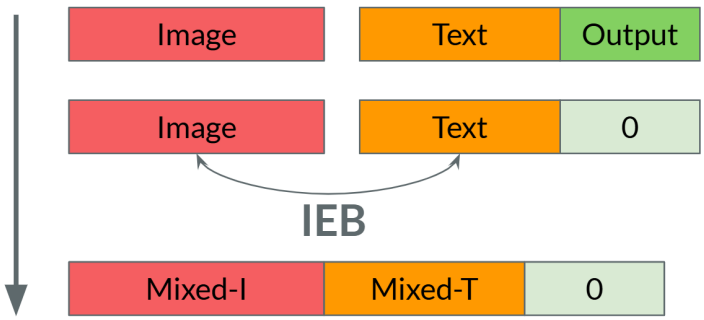


Figure 3.4 A masking mechanism schematic diagram of the IEB in MMtuning V2.

Subsequently, in the MWB, we use a one-dimensional mean pooling layer with dimensions $T_2 + T_2$ to replace the original dimension reduction operation, as illustrated in the figure 3.5. This design ensures that the fused modality-specific routing matrix no longer contains future information at the location of the label token, thus achieving effective information masking. In addition, during the inference process, the model is generated using multimodal input information only in the initial prediction phase. Therefore, the application of the MMtuning framework in the inference phase of the Decoder-only language model is limited to the first round of predictions. From the second round onwards, the model will follow the steps and methods of its intrinsic generation and will no longer rely on the MMtuning framework.

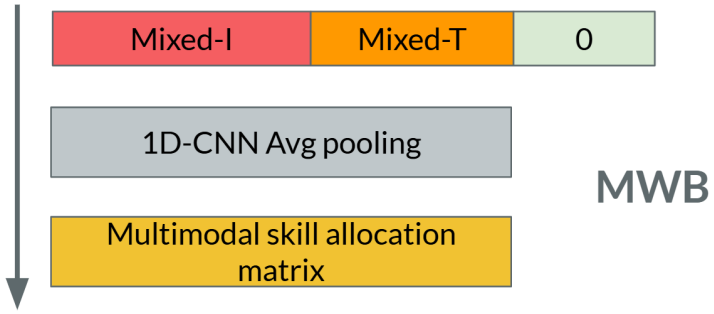


Figure 3.5 A masking mechanism schematic diagram of the MWB in MMtuning V2.

MMtuning V2 effectively solves the problem of information leakage during fine-tuning of Decoder-only multimodal large language models by introducing an innovative masking technique, while maintaining the same modularity and architectural compatibility as MMtuning V1. This strongly enhances the usability of MMtuning's parameter-efficient fine-tuning framework on a wider range of multimodal large language models.

3.3. MMtuning V3

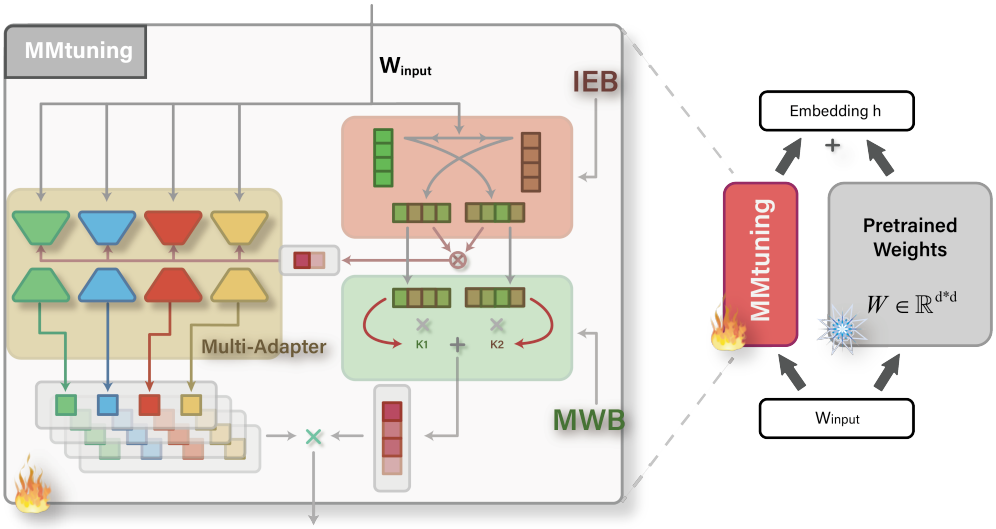


Figure 3.6 An overview of the MMtuning framework. In this framework, the left side features the Multi-Adapter block, which is illustrated in the figure as four LoRA adapters for extracting the Skill Inventory. The Interaction Enhancement Block (IEB) and the Modality Weights Block (MWB) are positioned on the right side and are responsible for processing modality-specific routing matrices. In the center, two red squares of different shades represent the Shared-Skill matrix. At the bottom, the four red squares in varying shades represent the multimodal skill allocation matrix.

In multimodal tasks, it is essential to emphasize that similar inputs should share analogous skills within the Skill Inventory, thereby enabling a more fine-grained

capture of differences across sample dimensions. As illustrated in figure 3.6, compared to previous versions of MMtuning, MMtuning V3 introduces the innovative Shared-Skill Matrix, which leverages a low-rank gating mechanism for LoRA Adapters to achieve fine-grained extraction of multimodal skills. The detailed methodology is outlined in the accompanying equations 3.10 and 3.11. The low-rank gating system first generates the rebuilt matrix X_{rebuild} by concatenating two updated weight matrices $W_{a_{\text{updated}}}$ and $W_{b_{\text{updated}}}$, and then computes the average X_{avg} for subsequent weighted operations. The gating mechanism then applies a softmax function to the weighted sum of the low-rank matrix $M_{\text{rank},r}$ and X_{avg}, adjusting the gating weights of the input Ax which from Lora Adapter's A matrix, and ultimately producing the gated output matrix $W_{\text{Gated\_Ax}} \in \mathbb{R}^{\text{Adapter\_num}\times\text{token}\times r}$. This process enhances the flow of information within tokens by selectively amplifying the shared skills between similar inputs.

$$\begin{aligned} \mathbf{X}_{\text{rebuild}} &= \text{concat}\left(W_{a_{\text{updated}}}, W_{b_{\text{updated}}}\right), \\ \mathbf{X}_{\text{avg}} &= \frac{1}{i+j}\sum_{k=1}^{i+j}\mathbf{X}_{\text{rebuild},k}, \end{aligned} \tag{3.10}$$

$$\begin{aligned} W_{\text{Gated\_Ax}} &= \mathbf{Ax} + \text{SoftMax}\left(\sum_{m=1}^{dk} M_{\text{rank},r}^{m}\cdot X_{\text{avg}}^{m}\right)\cdot \mathbf{Ax}, \\ \mathbf{Ax} &\in \mathbb{R}^{\text{Adapter\_num}\times\text{token}\times r}, \\ W_{\text{Gated\_Ax}} &\in \mathbb{R}^{\text{Adapter\_num}\times\text{token}\times r} \end{aligned} \tag{3.11}$$

Gating mechanisms, widely used in architectures such as Transformer [95, 96], RNN [97, 98], and CNN [99, 100] variants, are effective in filtering intra-token information within input matrices. The incorporation of the Shared-Skill Matrix further enhances the ability of the Multi-Adapter to strengthen internal skill information through a low-rank gating approach. Furthermore, due to its low-rank nature, this approach significantly reduces computational complexity compared to traditional gating mechanisms in the hidden dimension, making it well-suited for low-cost fine-tuning scenarios in PEFT. As shown in equation 3.11, within the Skill Inventory extracted by the Multi-Adapter, the Shared-Skill Matrix is shared across all adapters. In contrast, the multimodal skill allocation matrix is custom-designed for each adapter.

3.4. MMtuning Web Platform

In the field of contemporary artificial intelligence, the popularization of technology and the ease of operation are the key factors that promote the wide application of related models. To promote the widespread adoption of vision-language multimodal models, I have designed and developed a specialized web platform built upon the MMtuning framework. This platform is tailored specifically for fine-tuning vision-language multimodal models. Compared with traditional codes or specialized software, the platform makes it easy for users to fine-tune and acquire models without a programming background by simplifying the operation process. This design principle of intuitiveness and simplicity not only reduces the cognitive burden of users but also improves operational efficiency, thus enhancing the usability of the MMtuning fine-tuning platform.

Specifically, the platform is designed in the following areas:

- **Data upload guidance**: Through clear guidance and restriction prompts, it provides users with a clear path for data upload operation and necessary feedback on data format, which reduces users' uncertainty when uploading datasets.

- **Diversified model selection**: The platform provides diversified model selection functions to meet the specific needs of different user groups. Users can select appropriate visual and language models according to their needs to achieve personalized deep-learning model solutions. The platform also provides default visual and language model recommendations for users who are not deep learning professionals based on experimental results.

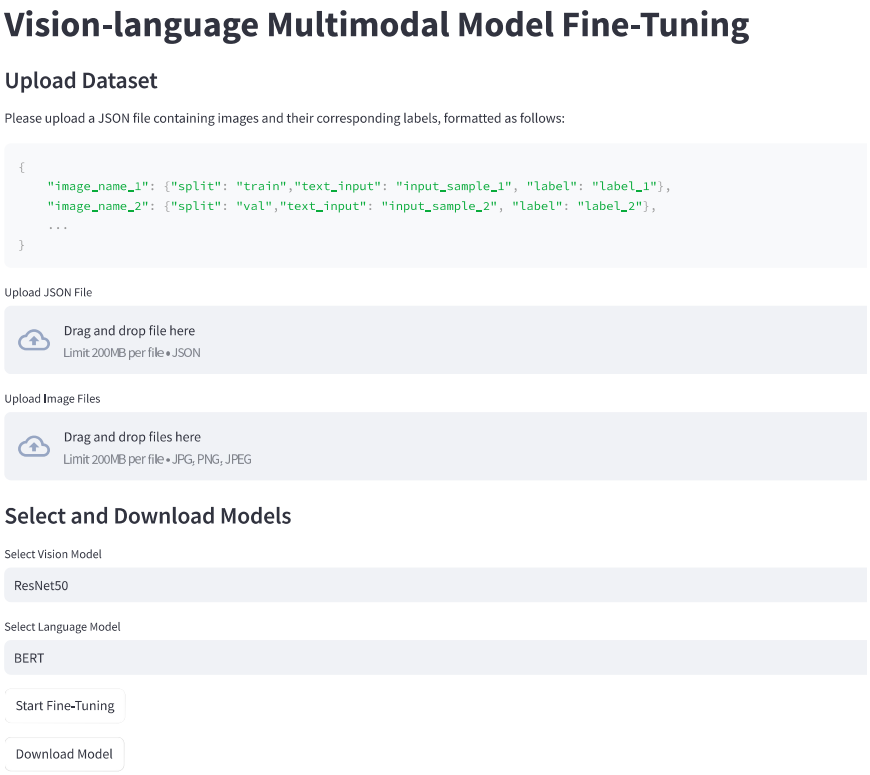
- **Automated fine-tuning process**: The parameter-efficient fine-tuning (re-training) process of models is simplified through automated design. Users can complete the fine-tuning with just one-click operation, without the need for complex parameter settings. This design reduces the users' requirements for deep learning or AI expertise, lowering the time cost and technical threshold for model fine-tuning.

- **Visualize the fine-tuning process**: Through the form of progress bars, the platform provides real-time feedback on the progress of fine-tuning, en-

abling users to monitor the training process intuitively. In addition, the platform will assist users in evaluating and understanding the capability of the fine-tuned model by displaying an exhaustive presentation of the parameters of the model's performance, such as loss and accuracy.

Based on the above design principles and guidelines, the web platform shown in Figure 3.7 is designed so that users can easily get started without deep knowledge of AI. In the data upload module, users can upload JSON files containing images, text, and labels (the output they want), and the platform helps users standardize the upload operation through formatting hints and supports batch uploading, which improves the efficiency of data preparation. The model selection module provides a variety of visual and language model options and recommends default models to meet different user needs. The fine-tuning training module simplifies the training process and allows users to intuitively monitor the training progress through the one-click start design and real-time progress bar visualization. Upon completion of training, users can directly access the fine-tuned model files through the download module. The whole process is convenient and efficient, which significantly reduces the user's technical threshold and improves operation efficiency and user experience.

Figure 3.7 MMtuning Web Platform: Provides a one-stop fine-tuning process for multimodal models in visual language, which is easy to operate and suitable for users with no programming and Deep-Learning background.



In summary, the support of the MMtuning framework ensures the efficiency of the fine-tuning process and the high quality of the models of the web platform within limited resources. Meanwhile, based on the above design and interaction principles, the platform embodies user-friendliness and highlights its important role in promoting the popularization of deep learning models and lowering the technical threshold. Specifically, by providing an easily accessible platform for fine-tuning and acquiring Vision-language multimodal models, it opens the door for users from non-technical backgrounds to enter the field of deep learning. Subsequently, bridges the lack of literacy that general designers and engineers making it difficult to access multimodal models in the era of AI and deep learning.

Chapter 4
Experiment

4.1. Experiment Setup of the MMtuning Framework

MM-LLMs framework To evaluate the effectiveness of the MMtuning framework, I constructed a visual-language multimodal large language pre-training model based on BLIP-2 architecture. The BLIP-2 architecture [2] ingeniously utilizes a two-stage pre-trained Querying Transformer (Q-Former) module to bridge the gap between different modalities, thereby achieving the goal of bootstrapping an unimodal pre-trained model to build a multimodal language model. In my experiments, for verifying the performance of the MMtuning V1, we employed the Transformer-based Encoder architecture google/vit-large-patch16-224 [101] as the visual pretraining model, and the Transformer-based Encoder-Decoder architecture T5-3B [102], adopting as the language pre-training mode. The visual and language models are connected using the pre-trained Q-Former module. In the MMtuning V1 experiment, we apply the MMtuning V1 framework to the query, key, and value components of the T5 Encoder, while freezing the visual model, language model, and Q-Former. To validate the effectiveness of the MMtuning V2 framework, I built a visual-language multimodal large language model based on the BLIP2 architecture. Different from the implementation of MMtuning V1, I replace the original large language model with a Decoder-only language model facebook/opt-2.7b [94], while keeping the visual model as google/vit-large-patch16-224 and the Q-former connecting the visual and language modules module fixed. In this study, the MMtuning V2 architecture is applied to the query, key, and value components of the OPT model. In addition, the visual, language, and Q-former modules were fixed during training, and only the MMtuning V2

component was trained to explore its impact on model performance in terms of fine-tuning. In the above experiments, in addition to the MMtuning V1 and V2 frameworks, trainable parameters include the two fully connected network layers connecting the visual model to the Q-Former and the Q-Former to the language model respectively.

Datasets I evaluated two publicly available Visual Question-Answer datasets: ScienceQA and Visual7W. ScienceQA [103] is a dataset containing multimodal contexts across different scientific fields, which is particularly well suited for multimodal question-answering tasks. For ScienceQA, we extracted only a subset of this dataset containing multiple choice questions of images and contexts for training and testing the quality of the generated text and the accuracy of the corresponding scientific topics. On the other hand, Visual7W [104] is a dataset focused on deep image understanding for language models and consists of 7W visual QA pairs divided into two distinct sections: Telling (dealing with textual responses) and Pointing (dealing with coordinates-based image annotation). For our purposes, we only used the Telling part as the VQA task, which is to extract visual multiple-choice question items containing six distinct types, for training and testing the quality of the generated text and the accuracy of the respective question types.

Baselines I validate the performance of the MMtuning framework by comparing three popular Parameter-Efficient Fine-Tuning (PEFT) methods – LoRA [8], Poly [12], and MoLoRA [11] – in the context of a multimodal QA task. All three methods have comparable parameter scales. I apply all PEFT methods uniformly to the query, key, and value matrices of the Encoder component of the T5 model in the experiment of MMtuning V1 framework. In the validation of the MMtuning V2 framework, I uniformly apply PEFT methods to the query, key, and value matrices of the OPT model. In all experiments, similar to PEFT methods based on the MoE architecture, all adapters in MMtuning are based on the low-rank matrices originating from LoRA, with parallel adapters count of four. In addition, in the MMtuning framework, the MWB employs a temperature coefficient of four ($t = 4$) as its weight output function.

Training Details Due to the relatively small scale of the ScienceQA dataset, ensuring satisfactory fine-tuning results, we conducted training for 3 epochs on ScienceQA and only one epoch on Visual7W. Recognizing the sensitivity of MoE architectures to hyperparameters [11,72,105], we referred to previous findings and utilized smaller batch sizes and lower learning rates to mitigate the risk of the MoE architecture tilting the output towards a single expert. Specifically, our training hyperparameters take a batch size of 8 and a learning rate of $1e^{-5}$ and utilize the AdamW optimizer [106] with a weight decay value set at $1e^{-8}$. In addition, we adopted a warm-up cosine linear decay strategy as the learning rate scheduler, with the warm-up steps set to 4000. All experiments were performed on a single NVIDIA GeForce RTX 4090 graphics card, and the parameter precision of the language model is bfloat16.

4.2. Main Results and Discussion

4.2.1 Evaluation of Skill Allocation Matrix Effectiveness in Multimodal Framework

In Table 4.1, we present a comparative analysis of the impact of distinct approaches to acquiring skill allocation matrices on conventional MoE-based PEFT methods. The experimental results indicate that in the Multi-Adapter architectures for multimodal tasks, the skill allocation matrix obtained by separating input information from distinct modalities leads to a marked enhancement in model performance compared to the conventional approach of processing input information. The conventional approach derives the skill allocation matrix directly from the entirety of the input information. In contrast, we postulate that, at the dataset level, the clustering outcomes produced by the model for different modalities may diverge from the ground-truth labels. At the sample level, to fulfill the downstream task requirements more effectively, individual samples may require different proportions of modality-specific routing matrices to optimally select the multimodal skill subset from the Multi-Adapter Block. Therefore, it is important to separate the input information of different modalities to form

Table 4.1 A comparative analysis of different PEFT methods for obtaining the skill allocation matrix in the Multi-Adapter architecture, evaluated on the ScienceQA and Visual7W datasets. The strategies denoted by Multi-MoLoRA and Multi-Poly utilize a multimodal skill allocation matrix. In the Multi-Adapter block, the setting for LoRA rank was uniformly set to $4 * 4$. The base model comprises the google/vit-large-patch16-224 as the visual backbone and the T5-3B model as the language backbone within the BLIP2 framework. We report the average Rouge1, RougeL, and multiple-choice Accuracy for all samples in the validation set, with higher scores for these metrics denoting superior performance.

Datasets	PEFT Methods	Metrics		
		Rouge1	**RougeL**	**ACC(%)**
ScienceQA	MoLoRA	65.30	65.04	54.41
	Multi-MoLoRA	**67.13**	**66.96**	**57.18**
	Poly	68.64	68.30	57.27
	Multi-Poly	**70.50**	**70.28**	**59.74**
Visual7W	MoLoRA	38.50	38.49	31.07
	Multi-MoLoRA	**40.13**	**40.11**	**33.61**
	Poly	40.02	40.02	32.82
	Multi-Poly	**40.31**	**40.30**	**33.89**

the final allocation matrix W_{skill}. For this reason, in the MMtuning framework, we attempt to explicitly segregate input information from diverse modalities and construct a multimodal skill allocation matrix tailored to a Multi-Adapter setup.

In the experiments, we separate the information outputted by the Q-Former (the first 32 tokens of input W_{input}) and the text data (the remaining tokens) directly inputted into the T5 model. Following dimension reduction through mean pooling in the token dimension and linear transformation operations, We derive two distinct modality-specific routing matrices, denoted as W_A and W_B, respectively. Ultimately, we obtain the overall multimodal skill allocation matrix, W_{skill}, by averaging the numerical value of these two modality-specific routing matrices, following Equation 4.1.

$$W_{\text{skill}} = \frac{1}{2}(W_A + W_B) \tag{4.1}$$

4.2.2 Empirical Evaluation of the Effectiveness of the MMtuning v1 and MMtuning V3 Framework

First, an in-depth comparative analysis of the generalizability of MMtuning is performed. Within the MMtuning framework, we not only establish a methodology for extracting individual modality-specific routing matrices from the input but also introduce the Interaction Enhancement Block (IEB) to fortify inter-modality information exchange and the Modality Weights Block (MWB) to facilitate the assignment of weights to modality-specific routing matrices. This design enables MMtuning to construct a multimodal skill allocation matrix W_{skill} tailored for multimodal input. Table 4.2 presents the performance of traditional PEFT methods alongside the MMtuning framework on the ScienceQA and Visual7W datasets. In these experiments, both Poly and MoLoRA employ a multimodal skill allocation matrix without the enhancements provided by IEB and MWB.

The results reveal two key insights:

(1) the MMtuning framework consistently surpasses the conventional PEFT methods on the ScienceQA and Visual7W datasets, even when using the same approach of separating input information based on different modalities. This superiority is attributed to the IEB and MWB in the MMtuning framework. The IEB facilitates cross-modality information fusion and knowledge-sharing by allowing modality-specific routing matrices to learn information in other routing matrices relevant to the original modality. Subsequently, the MWB provides an effective rule for dynamic modality weight allocation, enabling MMtuning V1 to analyze the important information in the input from different modalities to construct an optimal multimodal skill allocation matrix. Within the above two blocks, the robustness of the multimodal skill allocation matrix can be significantly improved relative to the conventional PEFT method, which speeds up the learning process of the model and enhances the adaptability to multimodal tasks. Ablation experiments also substantiate the contributions of the IEB and MWB to the importance and performance of the MMtuning framework.

(2) Moreover, through the synergistic interplay of the Shared-Skill matrix and the multimodal skill allocation matrix, MMtuning V3 significantly improves robustness in handling complex multimodal tasks, particularly in scenarios involving multiple subdomains. The incorporation of the Shared-Skill matrix in the MMtuning framework further emphasizes the importance of leveraging similarities in input x to effectively retrieve analogous multimodal skills from the Skill Inventory. Additionally, the multimodal skill allocation matrix refines the extracted skill combinations along the sample dimension, selectively extracting the most relevant subset of skills required for the current input. This synergy enables a more fine-grained regulation of the outputs generated by multiple adapters, facilitating the identification of the optimal skill combination for a given task.

(3) MMtuning demonstrates variable performance improvements between different types of multimodal QA tasks. Notably, it demonstrated a particularly pronounced improvement in the Visual7W dataset with only image information, compared to the ScienceQA dataset which integrates image and contextual information. This discrepancy can be attributed to the stronger inherent ability of the language model to extract information from direct text input. However, the Visual7W dataset relies heavily on information input from images. Due to the lack of descriptive context, the fine-tuning stage of conventional PEFT methods on the Visual7W dataset encountered a huge challenge. However, the information fusion mechanism within IEB significantly improves the ability of the language model to comprehend critical information in images that are relevant to the question of input. Ablation studies also provide compelling evidence. In summary, the IEB enables MMtuning to, in the absence of extensive contextual information, accurately identify the input-relevant content through a multimodal skill allocation matrix W_{skill}, thereby extracting the fine-grained multimodal expertise-relevant skills subset from the Multiple-Adapter block and improving performance in downstream multimodal tasks.

Table 4.2 Based on the BLIP2 framework, we employed the google/vit-large-patch16-224 model as the visual backbone and the T5-3b model as the language backbone for evaluation on the ScienceQA and Visual7W datasets. In the Multi-Adapter block the setting for LoRA rank was uniformly set to $4 * 4$, while the rank for individual LoRA was set to 16. We report the average Rouge1, RougeL, and multiple-choice Accuracy for all samples in the validation set. Higher values are preferred for all metrics.

Datasets	PEFT Methods	Metrics		
		Rouge1	**RougeL**	**Accuracy(%)**
ScienceQA	LoRA	59.48	59.22	46.59
	Multi-MoLoRA	67.13	66.96	57.18
	Multi-Poly	70.50	70.28	59.74
	MMtuning(IEB only)	72.44	72.27	63.23
	MMtuning V1	**73.08**	**72.95**	**64.62**
	MMtuning V3	**75.14**	**74.96**	**65.43**
Visual7W	LoRA	39.13	39.11	31.79
	Multi-MoLoRA	40.13	40.11	33.61
	Multi-Poly	40.31	40.30	33.89
	MMtuning(IEB only)	59.99	59.98	55.54
	MMtuning V1	**60.43**	**60.42**	**56.14**
	MMtuning V3	**63.03**	**62.94**	**58.89**

4.2.3 Empirical Evaluation of the Effectiveness of the MMtuning v2 Framework

Subsequently, I conducted exhaustive comparative experiments on the performance of the MMtuning V2 framework for fine-tuning the Decoder-only large language model. In the provided Table 4.3, the performance of MMtuning V2 on large language models (MM-LLMs) based on Decoder-only architecture is exhaustively demonstrated compared to other baseline PEFT methods. This experiment shows that similar to MMtuning V1 and V3, MMtuning V2 exhibits superior fine-

tuning performance in multimodal tasks. This result further confirms the significant advantages of MMtuning in the fine-tuning process of MM-LLMs in novel multimodal downstream tasks.

Table 4.3 Based on the BLIP2 framework, we employed the google/vit-large-patch16-224 model as the visual backbone and the opt-2.7b model as the language backbone for evaluation on the ScienceQA and Visual7W datasets. In the Multi-Adapter block the setting for LoRA rank was uniformly set to $4 * 4$, while the rank for individual LoRA was set to 16. We report the average Rouge1, RougeL, and multiple-choice Accuracy for all samples in the validation set. Higher values are preferred for all metrics.

Datasets	PEFT Methods	Metrics		
		Rouge1	**RougeL**	**Accuracy(%)**
ScienceQA	LoRA	45.79	45.62	39.20
	MoLoRA	45.32	45.20	35.48
	Multi-MoLoRA	49.18	46.51	39.34
	Poly	46.65	46.35	37.77
	Multi-Poly	49.46	49.18	38.91
	MMtuning V2	**57.36**	**57.27**	**53.61**

4.2.4 Parameter Efficiency Analysis

Figure 4.1 presents the comparative performance of diverse PEFT (Parameter-efficient Fine-Tuning) techniques when subjected to varying proportions of trainable parameters while preserving a constant number of training epochs and utilizing the same underlying base model on the ScienceQA dataset. Experiments were conducted across three distinct scales ($LoRA_{rank} = 8, 16, 32$), with each adapter scale set to $LoRA_{rank}/4$ to ensure comparable amounts of trainable parameters. The MMtuning framework realizes PEFT through the application of low-rank LoRA Adapters and simple multimodal router learning matrices. MMtuning outperforms all other methods across every scale and even outperforms other meth-

ods with a larger parameter count. Consequently, we assert that the MMtuning framework stands out as the most suitable parameter-efficient fine-tuning method for handling multimodal tasks, particularly for large-scale multimodal language models (MM-LLMs).

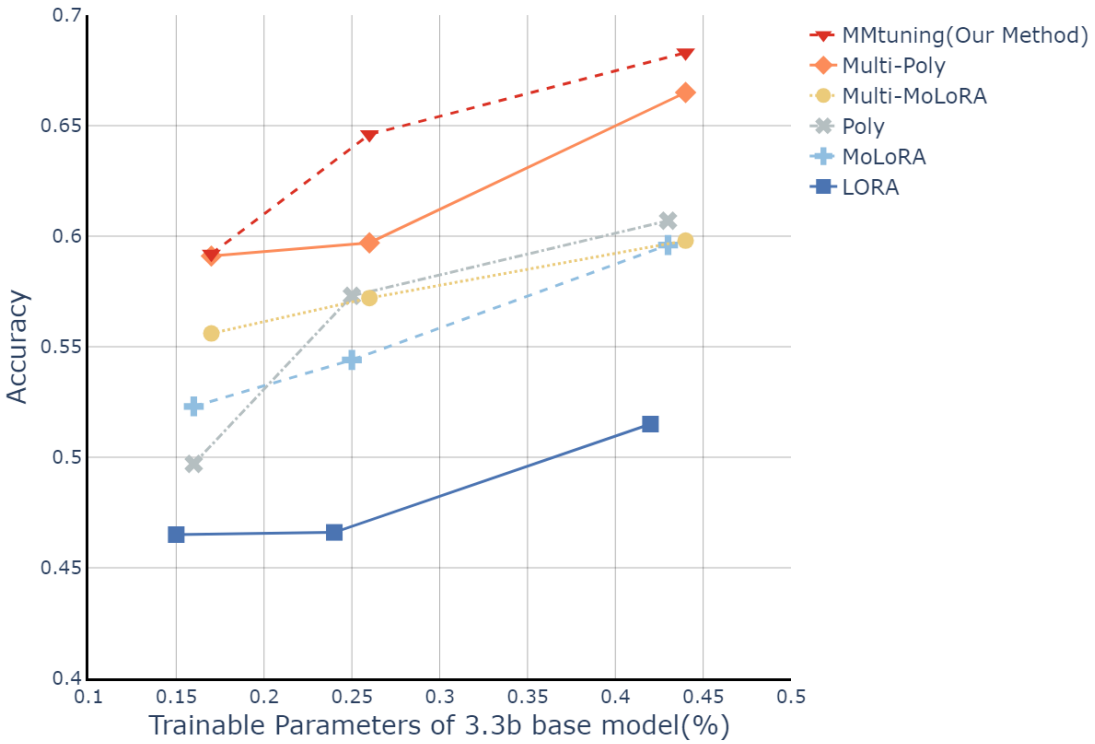


Figure 4.1 On the ScienceQA dataset, we evaluated the accuracy of various PEFT methods. Here, Multi-MoLoRA and Multi-Poly respectively represent the multi-modal skill allocation matrix strategies based on the original MoLoRA and Poly methods. The X-axis represents the percentage of trainable parameters relative to the 3.3B parameters of the base model.

4.3. In-depth Exploration of the MMtuning Framework

4.3.1 Enhanced Knowledge-Sharing through Modality Integration

The Interaction Enhancement Block (IEB) constructs a knowledge-sharing mechanism between different modalities through attention-based cross-modality information fusion, which effectively enhances the recognition of similarities across different modalities for the model. To validate whether this understanding of different modality input information by the IEB can facilitate the model distinguishing between different types of inputs x_i, we conducted experimental analysis on the ScienceQA dataset. Specifically, we performed a clustering analysis of the skill allocation matrix W_{skill} at specific layers of the Encoder of the T5 model to evaluate whether the multimodal skill allocation matrices have learned both differences and commonalities across various topics. These matrices of different topics were sampled at fixed intervals on all layers of the model encoder with Softmax normalization at the time of sampling. The results of the hierarchical clustering are shown in Figure 4.2, and this clustering reveals differences in the skill allocation among different topics and commonalities between similar topics. Similar to the differences in the methods employed by different journals to classify articles into disciplines [107], we heuristically acknowledge that the clustering cognition of one specific dataset for the model may not align perfectly with human classification schemes. Therefore, despite some discrepancies between the clustering results of the model and the ground truth labels of human classification for distinct scientific topics in the ScienceQA dataset, the figure can confirm that the IEB enables models to possess the capability to accurately identify differences and commonalities between different domains' inputs x_i through the cross-modality knowledge-sharing mechanism.

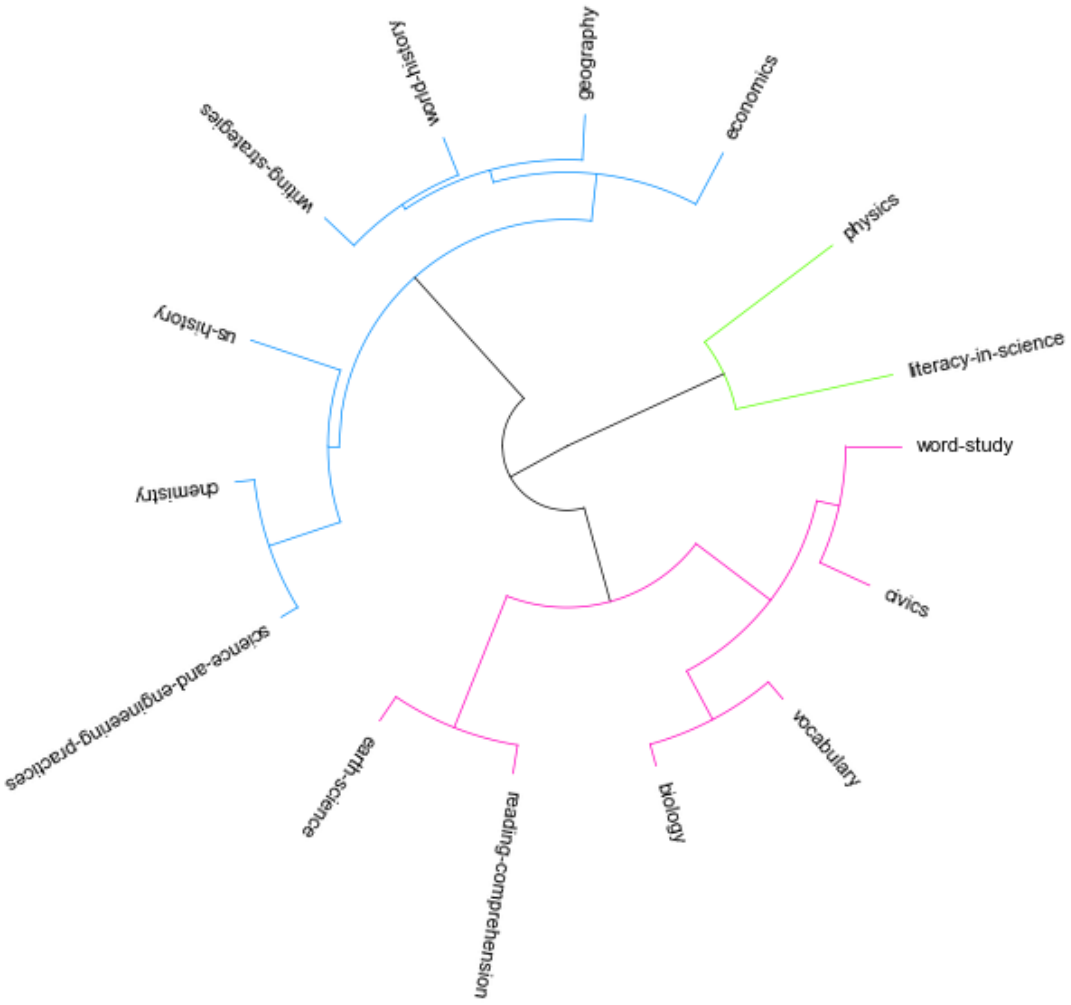


Figure 4.2 In the experiment of the ScienceQA dataset, we inquire about the topic clustering dendrogram of the multimodal skill allocation matrix W_{skill} within the MMtuning framework. This hierarchical clustering aims to classify scientific topics into common categories based on shared subsets of the allocation matrix.

4.3.2 Explicit modality segmentation enhances skill learning

As discussed in Chapter 3.1, a robust multimodal skill allocation matrix W_{skill} plays a crucial role in facilitating the model to acquire fine-grained skill subsets from the Multi-Adapter block. Figures 4.3, and 4.4 show the variation of the multimodal skill allocation matrix W_{skill} to varying depths of the model concerning the model depth when only the Interaction Enhancement Block (IEB) is employed (i.e., the skill allocation matrix is the average of the routing matrices of each modality). The results indicate that in the shallow layers of the encoder of the language model, the distribution of the final skill multimodal allocation matrix is closer to the numerical distribution of the text routing matrix. As shown in Figure 4.3, the numerical range of the text routing matrix significantly exceeds the range of values of the Q-former routing matrix. In contrast, at a deeper level, the model tends to allow the Q-former routing matrix to determine the numerical distribution of the final skill allocation matrix. Based on these findings, we infer that the language model attempts to learn information from different modality-specific routing matrices at varying depths and assigns corresponding importance weights. This suggests that the rule designed in the Modality Weights Block (MWB) is indeed rational in enabling the model to assign differing weights to the modality-specific routing matrices dynamically. Therefore, we propose that through the MWB mechanism, the MMtuning framework facilitates language models in actively extracting the relevant multimodal skill information required from various input modalities.

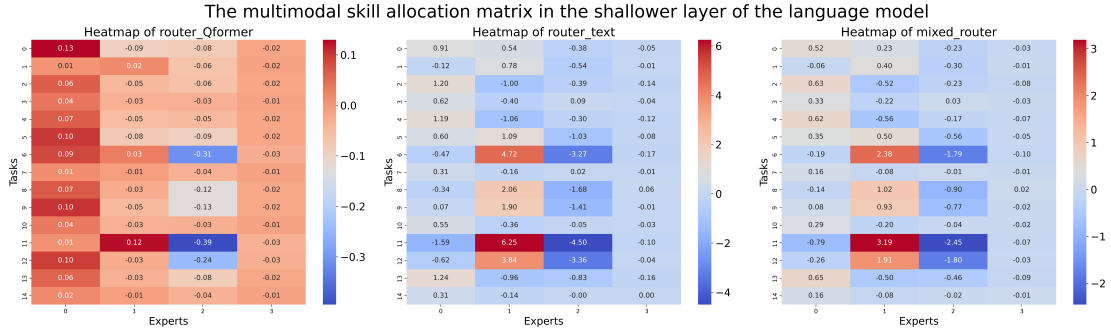


Figure 4.3 The visualization of individual routing matrices from each modality within the selected shallower Encoder layer of T5-3B, along with the multimodal skill allocation matrix obtained through average, after training on ScienceQA.

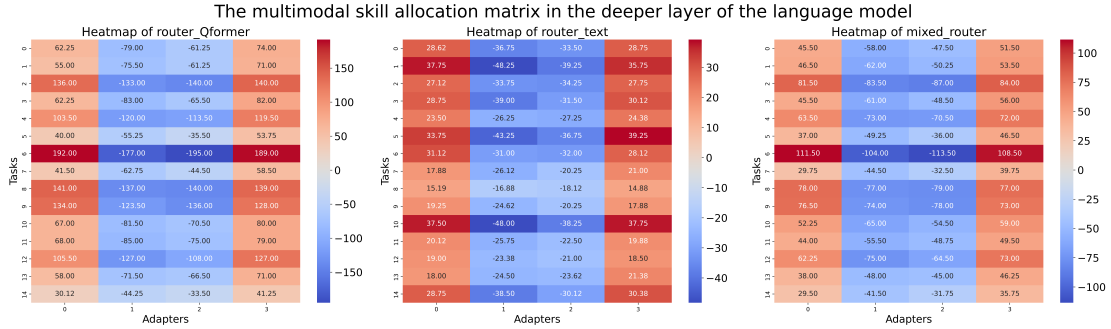


Figure 4.4 The visualization of individual routing matrices from each modality within the selected deeper Encoder layer of T5-3B, along with the multimodal skill allocation matrix obtained through average, after training on ScienceQA.

4.4. Experiments of the MMtuning Web Platform

This research is dedicated to the exploration of deep learning fine-tuning algorithms and focuses on the application of the research results through user-friendly interaction design, thus reinforcing the key role of the MMtuning fine-tuning

method in lowering the technological threshold and promoting the popularity of deep learning models. It aims to realize the vision of making deep learning models or AI models more accessible and applicable in the era of artificial intelligence. Specifically, the MMtuning web platform serves as an easy-to-access platform that enables users with non-technical backgrounds, especially the designer community, to easily fine-tune and access visual-language deep learning multimodal models that meet their creative needs. In addition, the platform's design principles and interactions aim to reduce the workload and cognitive burden on users, increase the efficiency of accessing multimodal models, and enhance the usability of MM-tuning techniques in solving real-world tasks.

4.4.1 Target Users

This web platform targets professional designers specializing in media design, visual arts, and interactive experiences, who possess rich creative expertise but lack technical proficiency in AI and programming. While they are highly interested in exploring deep learning-powered tools for visual and verbal creativity, they face significant challenges due to resource constraints and technical barriers in fine-tuning traditional deep learning models.

The designer user can be further segmented based on their specific needs and professional roles, including UI/UX designers, multimedia and content creators, AI application explorers, and multimodal information exploiters:

- Focused on creating engaging and intuitive interfaces and experiences, UI/UX designers seek multimodal models that recognize and respond to visual and textual inputs to deliver innovative interactive experiences.
- Multimedia and content creators, on the other hand, utilize visual and textual elements to communicate ideas. They show a keen interest in features such as automated authoring, smart quizzes, and AI assistants.
- AI application explorers are eager to integrate cutting-edge multimodal AI technologies into their design workflows without facing technical barriers.
- Multimodal information exploiters, on the other hand, are looking to leverage visual and textual data to get a more accurate picture of the user and

design needs to provide more precise solutions.

In a nutshell, these designer-user groups seek to leverage visual-language multimodal models to create more efficient, innovative, and personalized design solutions. As creative professionals from non-technical backgrounds, they urgently require accessible and practical fine-tuning technologies that simplify the integration of advanced multimodal models into their design processes while minimizing workload.

Based on the above analysis, I have identified the target user group for the MMtuning Web platform and verified the capability of the MMtuning Web Platform to reduce the workload of designers in acquiring visual-language multimodal models by comparing it with other fine-tuning applications.

4.4.2 Main Results

In this experiment, we aimed to evaluate the effectiveness of the MMtuning Web Platform in reducing the workload of designers in acquiring visual-language multimodal models, and the experiment compared three different model fine-tuning scenarios. The subjects of the experiment were 21 professionals working in design, mainly from the fields of media design, visual arts, and interactive experience design. They worked on developing AI applications using visual-language models to realize many of their interesting IDEAS. these designers typically have deep expertise in media design, visual arts, and interactive experience design, but lack the necessary technical background in AI, especially deep learning fine-tuning, as well as programming.

Experimental Scenario Design

Scenario 1 - Traditional PyTorch-based fine-tuning approach: In Scenario 1, participants use the PyTorch framework to directly perform fine-tuning of visual-language models. This process requires a high technical background and requires participants to manually write and run the code, which involves complex steps such as parameter setting, GPU management, and model optimization. Due to the high technical threshold, many designers need to spend a lot of time learning and find it difficult and error-prone to operate the process.

Scenario 2 - Stable Diffusion Web UI: Scenario 2 was tested using the Stable Diffusion Web UI software, which only supports fine-tuning of image generation models, it cannot handle visual-language multimodal tasks, so its applicability is limited. Participants were required to understand and manipulate the complex user interface to fine-tune the image generation model using LoRA [8] technology. And free parameter tuning and model selection are not possible.
Scenario 3 - MMtuning Web Platform: Scenario 3 shows the MMtuning Web Platform, a multimodal model fine-tuning platform developed in this study with a more user-friendly interface. The platform allows users to upload datasets by simple drag-and-drop and select the appropriate language and visual processing model according to the task requirements and cost, which is easy to operate and allows personalized fine-tuning of visual-language multimodal models without writing code. However, this web platform is unable to set many parameters in the model fine-tuning process, which may lead to the problem of the fine-tuned models not performing as expected.

Task Setting

Participants were tasked with fine-tuning and acquiring a visual language model model for a day (3 hours) in each scenario. This task was designed to assess the ease of use of different acquisition methods in practice and the workload imposed on the user.

Evaluation Scale

To evaluate the participants' workload in different scenarios, this study used the NASA Task Load Index (NASA-TLX) [15, 108], a widely recognized subjective workload assessment scale that was originally designed to measure the burden that an individual endures while performing a specific task. Designed by the National Aeronautics and Space Administration (NASA), the tool provides a careful quantitative analysis of task load through six core dimensions. These dimensions include psychological demands, physical demands, time demands, task performance, effort, and frustration. By considering these dimensions together, researchers can assess the psychological and physical demands and other subjective experiences of users during task performance. By quantifying the data from these dimensions, it

is possible to gain deeper insights into the workload conditions of users in different task scenarios.

Experimental Process

In the experiment, 21 participants, in three scenarios (each scenario for one day, 2 hours), performed the fine-tuning task described above. On the first day, model fine-tuning was performed by a PyTorch-based approach, which requires participants to have some knowledge of programming and deep learning, and has a high technical threshold, thus allowing users to use conversational AI such as ChatGPT [21]. Afterward, on the second day, they performed model fine-tuning by using the Stable Diffusion Web UI, an image generation model using the LoRA as a fine-tuning method that also has a relatively high operational complexity. On the final day, participants use the MMtuning Web Platform, which enables users to fine-tune and acquire models without a programming background through a simplified operational process.

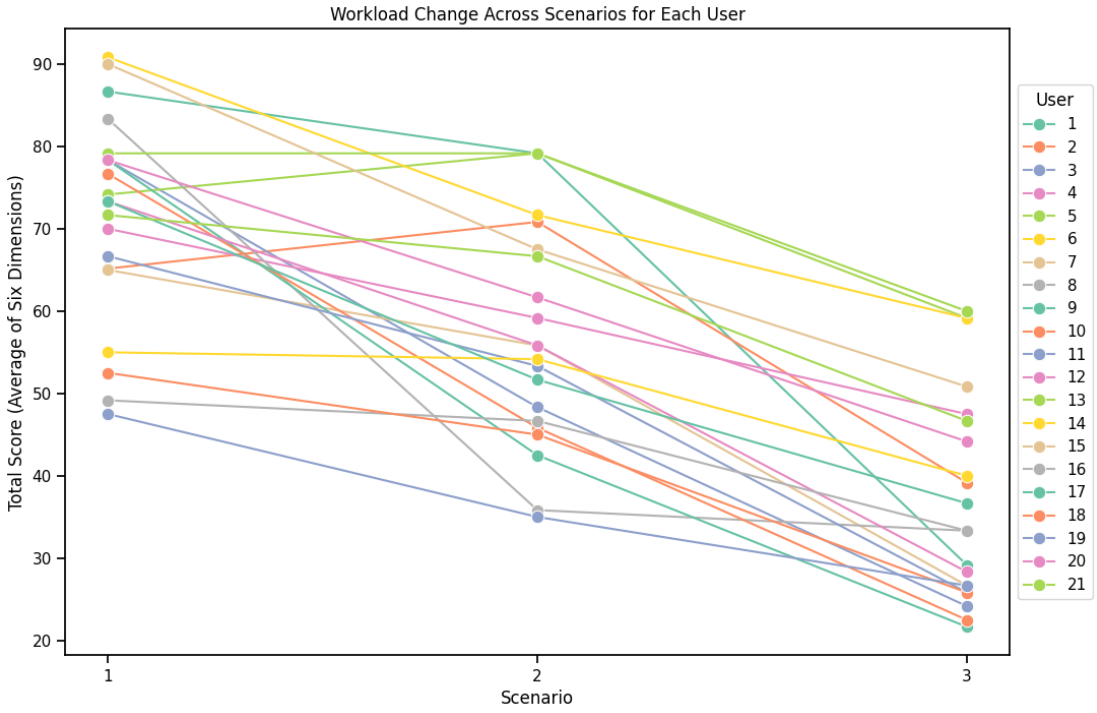


Figure 4.5 This line plot illustrates the changes in workload for each participant across different scenarios (Total Score). Each line represents a participant's total score in three experimental scenarios (Scenario 1, Scenario 2, and Scenario 3). It is observed that participants generally reported a higher workload in Scenario 1, while their scores significantly decreased in Scenario 3, indicating a lighter workload in that scenario. Different colors of the lines and markers represent individual participants, highlighting their score trends across the scenarios.

Experimental results and analysis

Based on the individual workload variations for each participant shown in Figure 4.5 and the overall workload distribution for the different scenarios presented in Figure 4.6, it can be observed that the participants' workloads were generally at a high level in Scenario 1. Further analysis reveals that the participants' workload decreased in Scenario 2, while in Scenario 3, the workload decreased significantly. Figure 4.6 further reveals the distribution of scores, means, and medians of par-

ticipants' workload in different scenarios. The results show that in Scenario 3, the mean and median workload of the participants were significantly lower than in the other scenarios. Scenario 1 had a higher technical threshold, resulting in participants facing greater challenges in the PyTorch fine-tuning process, especially in terms of code writing and environment configuration. In contrast, the Stable Diffusion Web UI of Scenario 2, while providing an intuitive interface, was limited in that it was only applicable to the image generation model and the complexity of the interface was not conducive to efficient operation by users with non-technical backgrounds.

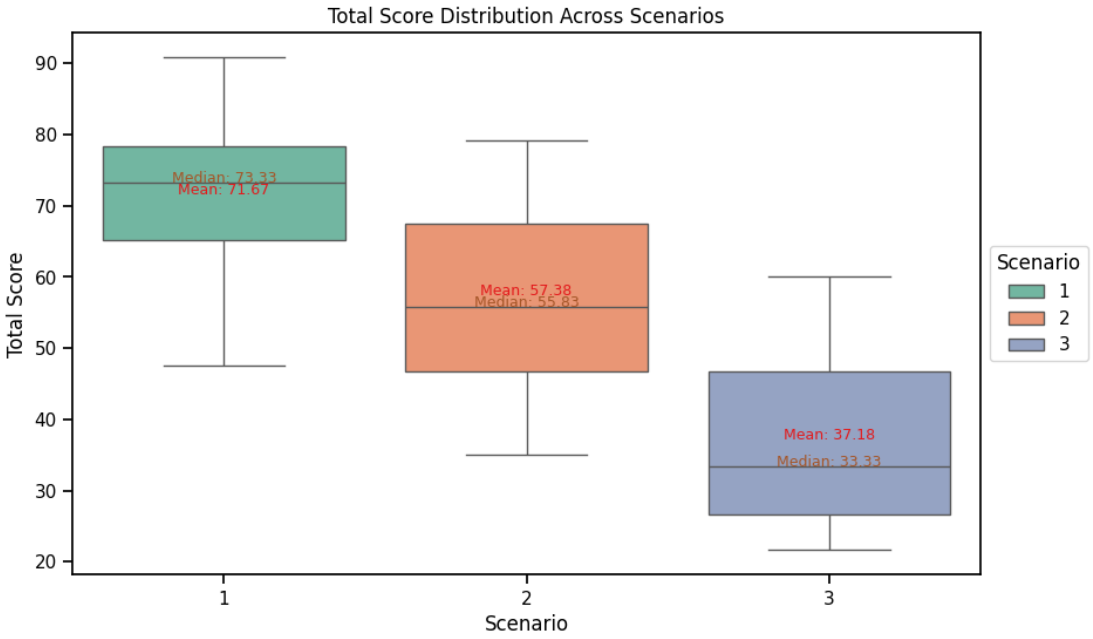


Figure 4.6 Box plot displaying the distribution of total scores across three scenarios (1, 2, and 3). Each box represents the interquartile range (IQR) of scores, with the median indicated by a line inside the box. The mean scores (shown in red) and median scores (shown in brown) are annotated for each scenario. The results demonstrate that the mean and median total scores are highest in Scenario 1, followed by Scenario 2, with Scenario 3 showing the lowest scores, indicating a significant reduction in workload across the scenarios.

Subsequently, further statistical analysis in Table 4.4 showed that the workload difference between Scene 1 and Scene 3 was significant, with a t-statistic of 10.996 and a p-value of 1×10^{-9}; and between Scene 2 and Scene 3, with a t-statistic of 9.371 and a p-value of 9×10^{-9}. These results indicate that the MMtuning Web platform reduces user workload and effectively lowers the technical barriers for participants in fine-tuning visual-language multimodal models.

Table 4.4 T-test results comparing workload across different scenarios. Higher t-statistics and lower p-values indicate stronger evidence of a significant difference between scenarios.

Comparison	**t-statistic**	**p-value**	**Result**
Scenario 1 vs 3	10.996	1×10^{-9}	Scenario 3 has a significantly lower workload than Scenario 1
Scenario 2 vs 3	9.371	9×10^{-9}	Scenario 3 has a significantly lower workload than Scenario 2

The experimental results confirm that the MMtuning Web Platform significantly reduces user workloads compared to the PyTorch-based approach and the Stable Diffusion Web UI. The MMtuning Web Platform substantially reduces the workload and technical threshold by simplifying the operational process and providing an intuitive user interface that allows designers from non-technical backgrounds to fine-tune and be more accessible visual-language multimodal models effortlessly.

Chapter 5
Conclusion

5.1. Conclusion

This study introduces MMtuning, a Parameter-Efficient Fine-tuning (PEFT) framework tailored for Multimodal Large Language Models (MM-LLMs). This framework effectively dynamically integrates various multimodal expertise in the Multi-Adapter block based on the input by employing an innovative multimodal skill allocation matrix. In MMtuning, we developed the Interaction Enhancement Block (IEB) and the Modality Weights Block (MWB). These two blocks aim to enhance knowledge-sharing across modalities by identifying the similarity and heterogeneity among all input modalities and optimizing the learning ability of modality-specific routing matrices. Consequently, they facilitate the construction of the most suitable shared-skill matrix and the multimodal skill allocation matrix at the sample level, instantiating the fine-grained extraction of the optimal multimodal skill subset from the task-relevant skill inventory. By incorporating low-rank LoRA adapters into the adapter modules of MMtuning, designed to extract task-relevant skill inventory, efficient fine-tuning of multimodal tasks has been successfully achieved within limited computational resources. In addition, we also conducted an in-depth investigation into the effectiveness of the MMtuning framework with a comprehensive comparative analysis and ablation study. At the same time, we also accomplish some interpretability analysis for the MMtuning framework. Finally, the experimental results reveal that irrespective of the scale of fine-tuning parameters, MMtuning displays exceptional proficiency in handling multimodal tasks, outperforms existing PEFT baselines, and attains state-of-the-art performance. These findings underscore the significance and transformative potential of our MMtuning framework for parameter-efficient multimodal downstream tasks within MM-LLMs.

In addition, comparative user experiments provide direct evidence that Web platforms built on the MMtuning framework can significantly lower the technical barriers faced by users from non-AI backgrounds in fine-tuning multimodal models to fit their specific needs. The platform's intuitive interface and simplified workflow enable users such as designers and software engineers with limited deep-learning expertise to bypass the complexity of traditional coding-based approaches. By allowing users to upload datasets, select appropriate models, and fine-tune them with minimal technical effort, the MMtuning-based Web platform significantly reduces user workload.

5.2. Contribution of the MMtuning Framework

In this study, my essential contributions for the Fine-tuning field are outlined below:

- We have extended the concepts of "skill inventory" and "skill allocation matrix" from multi-task PEFT to multimodal PEFT and introduced a more fine-grained mechanism for extracting optimal multimodal skill subsets at the sample level. This method enhances the model's adaptability and performance across various multimodal tasks by constructing an elaborated multimodal skill allocation mechanism that dynamically selects the most relevant multimodal skills from the skill inventory based on the input.

- We demonstrate that in multimodal scenarios, separating the input information from different modalities into individual modality-specific routing matrices is more effective in obtaining an optimal multimodal skill allocation matrix according to sample-level input than traditional approaches of processing all input information.

- Within the MMtuning framework, We introduce IEB and MWB. IEB is dedicated to enhancing the information exchange and knowledge sharing of different modality-specific routing matrices, thus improving the learning efficiency of each routing matrix. On the other hand, MWB assigns specific weights to each modality-specific routing matrix through a dynamic

weight allocation mechanism, to ensure the stability of the multimodal skill allocation matrix and achieve optimal decisions.

- We have explored the multimodal skill allocation matrix in detail. By analyzing each routing matrix, we empirically demonstrate how the language model of MM-LLMs progressively refines and integrates the distribution of different multimodal skill subsets in different model depths.
- We performed ablation experiments to validate the performance of IEB and MWB in the MMtuning framework. A series of comparative experiments establishes that MMtuning is the current state-of-the-art approach (SOTA) in the field of PEFT for MM-LLMs.

5.3. Contributions of MMtuning's Application

- We have released the MMtuning series code on GitHub (link: https://github.com/qiaoliamor/MMtuning), packaged as a user-friendly and well-documented Python library. This open-source release provides developers and researchers with a convenient tool for applying and extending the MMtuning framework in multimodal fine-tuning tasks. This initiative aims to promote the broader adoption and continued innovation of parameter-efficient fine-tuning in multimodal large language models.
- Based on the MMtuning framework, we build a user-friendly web platform that enables non-deep learning users to access multimodal models more conveniently. The scale results of NASA-TLX indicate that, compared to other popular methods, the platform significantly reduces the workload and technical barriers of the users when accessing visual-language multimodal models. Therefore, we expect that the platform will play a significant role in the broader application and popularization of visual-language multimodal models.

5.4. Limitations of the MMtuning Framework

Several limitations must be explicitly recognized. First, our experiments are based entirely on the Q-former architecture in BLIP2. Despite our significant achievements, the effectiveness of the MMtuning framework in other MM-LLMs frameworks remains to be verified. Second, MM-LLMs are often implemented in multi-task scenarios. However, the performance of MMtuning in multimodal multitask scenarios has not been comprehensively experimentally verified. Third, the MMtuning framework heavily relies on pre-trained bridge connectors to effectively align inputs from different modalities, thereby enhancing the performance of multimodal information processing. However, these pre-trained bridge connectors are typically designed for specific modality pairs, which significantly constrains the adaptability and scalability of MMtuning when applied to a broader range of multimodal tasks. Furthermore, MMtuning depends on the powerful encoding capabilities of large language models (LLMs) to extract and integrate multimodal information. Nevertheless, not all multimodal tasks require or are well-suited for LLMs, potentially leading to inefficient resource utilization.

5.5. Limitations of the MMtuning Web Platform

Additionally, while the MMtuning Web Platform lowers technical barriers by providing a user-friendly interface, from feedback many users without an AI background may still struggle to effectively integrate multimodal models into their design or software applications. This highlights the need for further research on making multimodal models more accessible in practical design and application scenarios.

5.6. Future Work

These limitations above will be prioritized as key directions for future work. Additionally, our current research focuses on constructing the multimodal skill allocation matrix. However, adapting adapters for extracting domain-specific knowledge more efficiently in a multimodal context needs further research. Moreover,

it is necessary to explore how the multimodal skill allocation matrix and adapter parameters in the MMtuning framework synergistically affect the performance and behavior of the model and conduct related interpretability research. Furthermore, future research will focus on further iterating and optimizing the MMtuning framework to eliminate reliance on pre-trained bridge connectors. The goal is to enhance its adaptability to various high-performance pre-trained unimodal models and multimodal tasks.

Furthermore, in terms of applications, future work will focus on developing more cost-effective and user-friendly multimodal AI applications to further lower the technological barriers of multimodal models. Specifically, we will explore more intuitive interactive software to help non-expert users seamlessly integrate multimodal models into their design and application workflows.

References

[1] Luca Zhou, Daniele Solombrino, Donato Crisostomi, Maria Sofia Bucarelli, Fabrizio Silvestri, and Emanuele Rodolà. Atm: Improving model merging by alternating tuning and merging, 2024. URL: https://arxiv.org/abs/2411.03055, arXiv:2411.03055.

[2] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, pages 19730–19742, 2023. URL: https://proceedings.mlr.press/v202/li23q.html.

[3] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS)*, volume 36, pages 34892–34916, 2023. URL: https://proceedings.neurips.cc/paper_files/paper/2023/file/6dcf277ea32ce3288914faf369fe6de0-Paper-Conference.pdf.

[4] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Peng Tan, Sinan andb Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. arXiv preprint arXiv:2308.12966 [cs.CV], 2023.

[5] Du-Zhen Zhang, Yahan Yu, Chenxing Li, Jiahua Dong, Dan Su, Chenhui Chu, and Dong Yu. Mm-llms: Recent advances in multimodal large language models. arXiv preprint arXiv:2401.13601v4 [cs.CL], 2024.

[6] Nicholas Botzer, Sameera Horawalavithana, Tim Weninger, and Svitlana Volkova. Lessons from developing multimodal models with code and developer interactions. In *I Can't Believe It's Not Better Workshop: Un-*

derstanding Deep Learning Through Empirical Falsification, 2022. URL: https://openreview.net/forum?id=iUUvNqUzJX2.

[7] Qidong Liu, Jiaxi Hu, Yutian Xiao, Xiangyu Zhao, Jingtong Gao, Wanyu Wang, Qing Li, and Jiliang Tang. Multimodal recommender systems: A survey, 2024. URL: https://arxiv.org/abs/2302.03883, arXiv:2302.03883.

[8] Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large lan-guage models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022.

[9] Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In *Proceedings of the 36th Conference on Neural Information Processing Systems (NeurIPS)*, volume 35, pages 1950–1965, 2022. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/0cde695b83bd186c1fd456302888454c-Paper-Conference.pdf.

[10] Nam Hyeon-Woo, Moon Ye-Bin, and Tae-Hyun Oh. Fedpara: Low-rank hadamard product for communication-efficient federated learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022.

[11] Ted Zadouri, Ahmet Üst Ün, Arash Ahmadian, Beyza Ermis, Acyr Locatelli, and Sara Hooker. Pushing mixture of experts to the limit: Extremely parameter efficient moe for instruction tuning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024.

[12] Edoardo M. Ponti, Alessandro Sordoni, Yoshua Bengio, and Siva Reddy. Combining modular skills in multitask learning, 2022. URL: https://arxiv.org/abs/2202.13914, arXiv:2202.13914.

[13] Haowen Wang, Tao Sun, Congyun Jin, Yingbo Wang, Yibo Fan, Yunqi Xu, Yuliang Du, and Cong Fan. Customizable combination of parameter-

efficient modules for multi-task learning. In *Proceedings of the Twelfth International Conference on Learning Representations (ICLR)*, 2024.

[14] Haowen Wang, Tao Sun, Kaixiang Ji, Jian Wang, Cong Fan, and Jinjie Gu. Orchmoe: Efficient multi-adapter learning with task-skill synergy, 2024.

[15] Ernesto Bustamante and Randall Spain. Measurement invariance of the nasa tlx. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 52:1522–1526, 09 2008. doi:10.1177/154193120805201946.

[16] Tadas Baltrusaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: A survey and taxonomy. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(2):423–443, feb 2019. URL: https://doi.org/10.1109/TPAMI.2018.2798607, doi:10.1109/TPAMI.2018.2798607.

[17] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y. Ng. Multimodal deep learning. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, page 689–696, Madison, WI, USA, 2011. Omnipress.

[18] Wanli Ouyang, Xiao Chu, and Xiaogang Wang. Multi-source deep learning for human pose estimation. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2337–2344, 2014. doi:10.1109/CVPR.2014.299.

[19] Anubhav Jangra, Sourajit Mukherjee, Adam Jatowt, Sriparna Saha, and Mohammad Hasanuzzaman. A survey on multi-modal summarization. *ACM Comput. Surv.*, 55(13s), jul 2023. URL: https://doi.org/10.1145/3584700, doi:10.1145/3584700.

[20] J. Wu, W. Gan, Z. Chen, S. Wan, and P. S. Yu. Multimodal large language models: A survey. In *2023 IEEE International Conference on Big Data (BigData)*, pages 2247–2256, Los Alamitos, CA, USA, dec 2023. IEEE Computer Society. URL: https://doi.ieeecomputersociety.org/10.1109/BigData59044.2023.10386743, doi:10.1109/BigData59044.2023.10386743.

[21] OpenAI. Gpt-4 technical report. arXiv:2303.08774 [cs.CL], 2023.

[22] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models, 2023. URL: https://arxiv.org/abs/2304.10592, arXiv:2304.10592.

[23] Yi-Kai Zhang, Shiyin Lu, Yang Li, YanQing Ma, Qing-Guo Chen, Zhao Xu, Weihua Luo, Kaifu Zhang, De-Chuan Zhan, and Han-Jia Ye. Wings: Learning multimodal LLMs without text-only forgetting. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL: https://openreview.net/forum?id=nqWaya7hiX.

[24] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh Sahand, Sharifzadeh Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning. In *Proceedings of the 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[25] B. P. Yuhas, M. H. Goldstein, and T. J. Sejnowski. Integration of acoustic and visual speech signals using neural networks. *Comm. Mag.*, 27(11):65–71, nov 1989. URL: https://doi.org/10.1109/35.41402, doi:10.1109/35.41402.

[26] Mihai Gurban, Jean-Philippe Thiran, Thomas Drugman, and Thierry Dutoit. Dynamic modality weighting for multi-stream hmms inaudio-visual speech recognition. In *Proceedings of the 10th International Conference on Multimodal Interfaces*, ICMI '08, page 237–240, New York, NY, USA, 2008. Association for Computing Machinery. URL: https://doi.org/10.1145/1452392.1452442, doi:10.1145/1452392.1452442.

[27] Triantafyllos Afouras, Joon Son Chung, Andrew Senior, Oriol Vinyals, and Andrew Zisserman. Deep audio-visual speech recognition. *Proceedings of*

the IEEE Transactions on Pattern Analysis and Machine Intelligence, PP:1–1, 12 2018. doi:10.1109/TPAMI.2018.2889052.

[28] Ringki Das and Thoudam Doren Singh. Multimodal sentiment analysis: A survey of methods, trends, and challenges. *ACM Comput. Surv.*, 55(13s), jul 2023. URL: https://doi.org/10.1145/3586075, doi:10.1145/3586075.

[29] Songning Lai, Xifeng Hu, Haoxuan Xu, Zhaoxia Ren, and Zhi Liu. Multimodal sentiment analysis: A survey, 2023. URL: https://arxiv.org/abs/2305.07611, arXiv:2305.07611.

[30] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. Mmbench: Is your multi-modal model an all-around player? In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part VI*, page 216–233, Berlin, Heidelberg, 2024. Springer-Verlag. URL: https://doi.org/10.1007/978-3-031-72658-3_13, doi:10.1007/978-3-031-72658-3_13.

[31] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine Mcleavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 28492–28518. PMLR, 23–29 Jul 2023. URL: https://proceedings.mlr.press/v202/radford23a.html.

[32] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

[33] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014,*

Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings, 2014. arXiv:http://arxiv.org/abs/1312.6114v10.

[34] stable-diffusion webui. Stable diffusion web ui: A web interface for stable diffusion, implemented using gradio library. URL: https://github.com/AUTOMATIC1111/stable-diffusion-webui.

[35] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022. URL: https://arxiv.org/abs/2204.06125, arXiv:2204.06125.

[36] Hugging Face. The ai community building the future. the platform where the machine learning community collaborates on models, datasets, and applications. URL: https://huggingface.co.

[37] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf.

[38] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: a system for large-scale machine learning. In *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, OSDI'16, page 265–283, USA, 2016. USENIX Association.

[39] Xu Han, Zhengyan Zhang, Ning Ding, Yuxian Gu, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Yuan Yao, Ao Zhang, Liang Zhang, Wentao Han,

Minlie Huang, Qin Jin, Yanyan Lan, Yang Liu, Zhiyuan Liu, Zhiwu Lu, Xipeng Qiu, Ruihua Song, Jie Tang, Ji-Rong Wen, Jinhui Yuan, Wayne Xin Zhao, and Jun Zhu. Pre-trained models: Past, present and future. *AI Open*, 2:225–250, 2021. URL: https://www.sciencedirect.com/science/article/pii/S2666651021000231, doi:https://doi.org/10.1016/j.aiopen.2021.08.002.

[40] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. URL: https://aclanthology.org/N19-1423, doi:10.18653/v1/N19-1423.

[41] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[42] Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, LILI YU, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. LIMA: Less is more for alignment. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL: https://openreview.net/forum?id=KBMOKmX2he.

[43] Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V. Le, Barret Zoph, Jason Wei, and Adam Roberts. The flan collection: designing data and methods for effective instruction tuning. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[44] Hamish Ivison, Akshita Bhagia, Yizhong Wang, Hannaneh Hajishirzi, and Matthew E. Peters. Hint: Hypernetwork instruction tuning for efficient zero- and few-shot generalisation. In *Annual Meeting of the Association for Computational Linguistics*, 2022. URL: `https://api.semanticscholar.org/CorpusID:254877064`.

[45] Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*, 2022. URL: `https://openreview.net/forum?id=9Vrb9D0WI4`.

[46] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2022. URL: `https://openreview.net/forum?id=gEZrGCozdqR`.

[47] Xiaohu Jiang, Yixiao Ge, Yuying Ge, Dachuan Shi, Chun Yuan, and Ying Shan. Supervised fine-tuning in turn improves visual foundation models, 2024. URL: `https://arxiv.org/abs/2401.10222`, `arXiv:2401.10222`.

[48] Xiaohu Jiang, Yixiao Ge, Yuying Ge, Chun Yuan, and Ying Shan.

Supervised fine-tuning in turn improves visual foundation models. *ArXiv*, abs/2401.10222, 2024. URL: https://api.semanticscholar.org/CorpusID:267035230.

[49] Jie Chen, Xintian Han, Yu Ma, Xun Zhou, and Liang Xiang. Unlock the correlation between supervised fine-tuning and reinforcement learning in training code large language models, 2024. URL: https://arxiv.org/abs/2406.10305, arXiv:2406.10305.

[50] Yash Goyal, Tejas Khot, Aishwarya Agrawal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. *Int. J. Comput. Vision*, 127(4):398–414, April 2019. URL: https://doi.org/10.1007/s11263-018-1116-0, doi:10.1007/s11263-018-1116-0.

[51] Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. Prismatic vlms: Investigating the design space of visually-conditioned language models, 2024. URL: https://arxiv.org/abs/2402.07865, arXiv:2402.07865.

[52] Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. Instruction tuning for large language models: A survey, 2024. URL: https://arxiv.org/abs/2308.10792, arXiv:2308.10792.

[53] Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. Instruction tuning for large language models: A survey, 2024. URL: https://arxiv.org/abs/2308.10792, arXiv:2308.10792.

[54] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. InstructBLIP: Towards general-purpose vision-language models with instruction tuning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL: https://openreview.net/forum?id=vvoWPYqZJA.

[55] Xi Chen, Xiao Wang, Lucas Beyer, Alexander Kolesnikov, Jialin Wu, Paul Voigtlaender, Basil Mustafa, Sebastian Goodman, Ibrahim Alabdulmohsin, Piotr Padlewski, Daniel Salz, Xi Xiong, Daniel Vlasic, Filip Pavetic, Keran Rong, Tianli Yu, Daniel Keysers, Xiaohua Zhai, and Radu Soricut. Pali-3 vision language models: Smaller, faster, stronger, 2023. URL: https://arxiv.org/abs/2310.09199, arXiv:2310.09199.

[56] Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection, 2024. URL: https://arxiv.org/abs/2311.10122, arXiv:2311.10122.

[57] Hanrong Ye, De-An Huang, Yao Lu, Zhiding Yu, Wei Ping, Andrew Tao, Jan Kautz, Song Han, Dan Xu, Pavlo Molchanov, and Hongxu Yin. X-vila: Cross-modality alignment for large language model, 2024. URL: https://arxiv.org/abs/2405.19335, arXiv:2405.19335.

[58] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 2790–2799, 2019. URL: https://proceedings.mlr.press/v97/houlsby19a.html.

[59] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3045–3059, 2021. URL: https://aclanthology.org/2021.emnlp-main.243, doi:10.18653/v1/2021.emnlp-main.243.

[60] Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. Gpt understands, too. arXiv preprint arXiv:2103.10385v2 [cs.CL], 2023.

[61] Lisa Xiang, Li, and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the*

Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL/IJCNLP), pages 4582–4597, 2021. URL: https://aclanthology.org/2021.acl-long.353, doi: 10.18653/v1/2021.acl-long.353.

[62] Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, volume 02, pages 61–68, 2022. URL: https://aclanthology.org/2022.acl-short.8, doi: 10.18653/v1/2022.acl-short.8.

[63] Yiming Wang, Yu Lin, Xiaodong Zeng, and Guannan Zhang. Multilora: Democratizing lora for better multi-task learning. arXiv preprint arXiv:2311.11501v1 [cs.LG], 2023.

[64] Lucas Caccia, Edoardo Ponti, Zhan Su, Matheus Pereira, Nicolas Le Roux, and Alessandro Sordoni. Multi-head adapter routing for cross-task generalization. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL: https://openreview.net/forum?id=qcQhBli5Ho.

[65] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 12888–12900. PMLR, 17–23 Jul 2022. URL: https://proceedings.mlr.press/v162/li22n.html.

[66] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume

139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 2021. URL: http://proceedings.mlr.press/v139/radford21a.html.

[67] Alex Jinpeng Wang, Yixiao Ge, Rui Yan, Ge Yuying, Xudong Lin, Guanyu Cai, Jianping Wu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. All in one: Exploring unified video-language pre-training. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[68] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L. Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learning via sparse sampling. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7327–7337, 2021. URL: https://api.semanticscholar.org/CorpusID:231880022.

[69] Haixin Wang, Xinlong Yang, Jianlong Chang, Dian Jin, Jinan Sun, Shikun Zhang, Xiao Luo, and Qi Tian. Parameter-efficient tuning of large-scale multimodal foundation model. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2024. Curran Associates Inc.

[70] Haoyu Lu, Yuqi Huo, Guoxing Yang, Zhiwu Lu, Wei Zhan, Masayoshi Tomizuka, and Mingyu Ding. Uniadapter: Unified parameter-efficient transfer learning for cross-modal modeling, 2023. URL: https://arxiv.org/abs/2302.06605, arXiv:2302.06605.

[71] Muhammad Uzair khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[72] William Fedus, Noam Shazeer, and Alexander Clark. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23:1–39, 2022.

[73] Noam M. Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and J. Dean. Outrageously large neu-

ral networks: The sparsely-gated mixture-of-experts layer. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[74] Carlos Riquelme, Google Brain, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton Google, Brain André, Susano Pinto, and Neil Houlsby. Scaling vision with sparse mixture of experts. In *Proceedings of the 35th Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

[75] Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, Barret Zoph, Liam Fedus, Maarten P Bosma, Zongwei Zhou, Tao Wang, Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, Kathleen Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc Le, Yonghui Wu, Zhifeng Chen, and Claire Cui. GLaM: Efficient scaling of language models with mixture-of-experts. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pages 5547–5569, 2022. URL: `https://proceedings.mlr.press/v162/du22c.html`.

[76] Yuxuan Lou, Fuzhao Xue, Zangwei Zheng, and Yang You. Cross-token modeling with conditional computation. arXiv preprint arXiv:2109.02008v3 [cs.LG], 2022.

[77] Dmitry Lepikhin, Dehao Chen, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with condi-tional computation and automatic sharding. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[78] Simiao Zuo, Xiaodong Liu, Jian Jiao, Young Jin Kim, Hany Hassan, Ruofei Zhang, Tuo Zhao, and Jianfeng Gao. Taming sparsely activated transformer with stochastic experts. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022.

[79] David. Eigen, Marc ' Aurelio Ranzato, and Ilya Sutskever. Learning factored representations in a deep mixture of experts. arXiv preprint arXiv:1312.4314v3 [cs.LG], 2014.

[80] Run Luo, Haonan Zhang, Longze Chen, Ting-En Lin, Xiong Liu, Yuchuan Wu, Min Yang, Minzheng Wang, Pengpeng Zeng, Lianli Gao, Heng Tao Shen, Yunshui Li, Xiaobo Xia, Fei Huang, Jingkuan Song, and Yongbin Li. Mmevol: Empowering multimodal large language models with evol-instruct, 2024. URL: https://arxiv.org/abs/2409.05840, arXiv:2409.05840.

[81] Mojtaba Valipour, Mehdi Rezagholizadeh, Ivan Kobyzev, and Ali Ghodsi. DyLoRA: Parameter-efficient tuning of pre-trained models using dynamic search-free low-rank adaptation. In Andreas Vlachos and Isabelle Augenstein, editors, *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3274–3287, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. URL: https://aclanthology.org/2023.eacl-main.239, doi:10.18653/v1/2023.eacl-main.239.

[82] Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adalora: Adaptive budget allocation for parameter-efficient fine-tuning. In *Proceedings of the 11th International Conference on Learning Representations (ICLR)*, 2023.

[83] Tianlin Liu, Mathieu Blondel, Carlos Riquelme Ruiz, and Joan Puigcerver. Routers in vision mixture of experts: An empirical study. *Transactions on Machine Learning Research*, 2024. URL: https://openreview.net/forum?id=aHk3vctnf1.

[84] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2019. URL: https://arxiv.org/abs/1807.03748, arXiv:1807.03748.

[85] Basil Mustafa, C. Riquelme, J. Puigcerver, Rodolphe Jenatton, and N. Houlsby. Multimodal contrastive learning with limoe: the language-image mixture of experts. In *Proceedings of the 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[86] Jialiang Zhao Kaiming He Lirui Wang, Xinlei Chen. Scaling proprioceptive-visual learning with heterogeneous pre-trained transformers. In *Neurips*, 2024.

[87] Gianni Brauwers and Flavius Frasincar. A general survey on attention mechanisms in deep learning. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, PP:1–1, 11 2021. doi:10.1109/TKDE.2021.3126456.

[88] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st Conference on Neural Information Processing System (NeurIPS)*, page 6000–6010, 2017. URL: https://api.semanticscholar.org/CorpusID:13756489.

[89] Tianzhu Ye, Li Dong, Yuqing Xia, Yutao Sun, Yi Zhu, Gao Huang, and Furu Wei. Differential transformer, 2024. URL: https://arxiv.org/abs/2410.05258, arXiv:2410.05258.

[90] Anonymous. Selective attention improves transformer. In *Submitted to The Thirteenth International Conference on Learning Representations*, 2024. under review. URL: https://openreview.net/forum?id=v0FzmPCd1e.

[91] Weijie Tu, Weijian Deng, and Tom Gedeon. A closer look at the robustness of contrastive language-image pre-training (CLIP). In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL: https://openreview.net/forum?id=wMNpMe0vp3.

[92] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI*, page 776–794. Springer-Verlag, 2020. URL: https://doi.org/10.1007/978-3-030-58621-8_45, doi:10.1007/978-3-030-58621-8_45.

[93] Hinton Geoffrey, Vinyals Oriol, and Dean Jeffrey. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531v1 [stat.ML], 2015.

[94] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer.

Opt: Open pre-trained transformer language models, 2022. URL: https://arxiv.org/abs/2205.01068, arXiv:2205.01068.

[95] Yifei Wang, Yuyang Wu, Zeming Wei, Stefanie Jegelka, and Yisen Wang. A theoretical understanding of self-correction through in-context alignment. In *ICML 2024 Workshop on In-Context Learning*, 2024. URL: https://openreview.net/forum?id=XHP3t1AUp3.

[96] R. Liu, Young Jin Kim, Alexandre Muzio, Barzan Mozafari, and Hany Hassan Awadalla. Gating dropout: Communication-efficient regularization for sparsely activated transformers. In *International Conference on Machine Learning*, 2022. URL: https://api.semanticscholar.org/CorpusID:249192179.

[97] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*, 2014.

[98] Chandrashekar Lakshminarayanan and Amit Vikram Singh. Neural path features and neural path kernel: understanding the role of gates in deep learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[99] Jianfeng Wang and Xiaolin Hu. Convolutional neural networks with gated recurrent connections. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3421–3435, 2022. doi:10.1109/TPAMI.2021.3054614.

[100] Towaki Takikawa, David Acuna, V. Jampani, and Sanja Fidler. Gated-scnn: Gated shape cnns for semantic segmentation. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5228–5237, 2019. URL: https://api.semanticscholar.org/CorpusID:196471056.

[101] Dosovitskiy Alexey, Beyer Lucas, Kolesnikov Alexander, Weissenborn Dirk, Zhai Xiaohua, Unterthiner Thomas, Dehghani Mostafa, Minderer Matthias, Heigold Georg, Gelly Sylvain, Uszkoreit Jakob, and Houlsby Neil. An image

is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of The 7th International Conference on Learning Representations (ICLR)*, 2021.

[102] Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL: `http://jmlr.org/papers/v21/20-074.html`.

[103] Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *Proceedings of the 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[104] Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. Visual7w: Grounded question answering in images. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pages 4995–5004, 2016.

[105] Shen Sheng, Hou Le, Zhou Yanqi, Du Nan, Longpre Shayne, Wei Jason, Chung Hyung Won, Zoph Barret, Fedus William, Chen Xinyun, Vu Tu, Wu Yuexin, Chen Wuyang, Webson Albert, Li Yunxuan, Zhao Vincent Y, Yu Hongkun, Keutzer Kurt, Darrell Trevor, and Zhou Denny. Mixture-of-experts meets instruction tuning: A winning combination for large language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024.

[106] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, 2019.

[107] Linda Sīle, Raf Guns, Frédéric Vandermoere, Gunnar Sivertsen, and Tim C. E. Engels. Tracing the context in disciplinary classifications: A bibliometric pairwise comparison of five classifications of journals in the social sciences and humanities. *Quantitative Science Studies*, 2(1):65–88, 04

2021. URL: https://doi.org/10.1162/qss_a_00110, doi:10.1162/qss_a_00110.

[108] Sadiq Said, Malgorzata Gozdzik, Tadzio Raoul Roche, Julia Braun, Julian Rössler, Alexander Kaserer, Donat R Spahn, Christoph B Nöthiger, and David Werner Tscholl. Validation of the raw national aeronautics and space administration task load index (NASA-TLX) questionnaire to assess perceived workload in patient monitoring tasks: Pooled analysis study using mixed models. *J. Med. Internet Res.*, 22(9):e19472, September 2020.