SPECTATE-GPT: A SENSIBLE LARGE VISION-LANGUAGE MODEL ASSISTING IN COMPREHENDING THE IMAGE CONTENT

Project Submitted to the
SRM University AP, Andhra Pradesh
for the partial fulfillment of the requirements to award the degree of

Bachelor of Technology
in
Computer Science & Engineering
School of Engineering & Sciences

submitted by

Tulasi Sai Tharun Peram(AP20110010801)

L Sree Nidhi(AP20110010804)

Hashmmath Shaik(AP20110010809)
Taathvika Morampudi(AP20110010833)

Under the Guidance of
Dr. Rajiv Senapati

Department of Computer Science & Engineering
SRM University-AP
Neerukonda, Mangalgiri, Guntur
Andhra Pradesh - 522 240
May 2024

DECLARATION

I undersigned hereby declare that the project report **Spectate-GPT: A Sensible Large Vision-Language Model Assisting in Comprehending the Image Content** submitted for partial fulfillment of the requirements for the award of degree of Bachelor of Technology in the Computer Science & Engineering, SRM University-AP, is a bonafide work done by me under supervision of Dr. Rajiv Senapati. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree of any other University.

Place	:	Date	: May 11, 2024
Name of student	: Tulasi Sai Tharun Peram	Signature	: .....
Name of student	: L Sree Nidhi	Signature	: .....
Name of student	: Hashmmath Shaik	Signature	: .....
Name of student	: Taathvika Morampudi	Signature	: .....

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SRM University-AP

Neerukonda, Mangalgiri, Guntur

Andhra Pradesh - 522 240

CERTIFICATE

This is to certify that the report entitled **Spectate-GPT: A Sensible Large Vision-Language Model Assisting in Comprehending the Image Content** submitted by **Tulasi Sai Tharun Peram, L Sree Nidhi, Hashmmath Shaik, Taathvika Morampudi** to the SRM University-AP in partial fulfillment of the requirements for the award of the Degree of Master of Technology in in is a bonafide record of the project work carried out under my/our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Project Guide	Head of Department
Name : Dr. Rajiv Senapati	Name : Prof. Niraj Upadhayaya
Signature:	Signature: .....

ACKNOWLEDGMENT

I wish to record my indebtedness and thankfulness to all who helped me prepare this Project Report titled **Spectate-GPT: A Sensible Large Vision-Language Model Assisting in Comprehending the Image Content** and present it satisfactorily.

I am especially thankful for my guide and supervisor Dr. Rajiv Senapati in the Department of Computer Science & Engineering for giving me valuable suggestions and critical inputs in the preparation of this report. I am also thankful to Prof. Niraj Upadhayaya, Head of Department of Computer Science & Engineering for encouragement.

My friends in my class have always been helpful and I am grateful to them for patiently listening to my presentations on my work related to the Project.

Tulasi Sai Tharun Peram, L Sree Nidhi, Hashmmath Shaik, Taathvika Morampudi
(Reg. No. AP20110010801, AP20110010804, AP20110010809, AP20110010833)
B. Tech.
Department of Computer Science & Engineering
SRM University-AP

ABSTRACT

There has been many advancements in Generative AI field in every aspect of the real world scenario and their applications for example it is used in generating human-like content for having conversation using Large Language Models (LLMs) and Generative Pre-Trained Transformers (GPT). They are also used in Generating images, music, video content using Generative Adversarial Networks and Transformers, and even suggesting ideas based on the topic or the query using the above technologies. This clearly indicates that it can be further developed, and after all the research done on the advancements on this field we were amazed by the fact that they were improved in the name of Large Vision-Language Models where the models are provided with vision to read and understand the content present in the images that are provided by the user and assist them accordingly.

This gave the idea to our model Spectate-GPT which unlike the other Large Vision-Language Models has been given a bit of common sense to act like a human being and assist like a human being and as the name suggests ”spectate” the model takes image input from the user, then it assists in briefing the contents of the image in a sensible way and answering the queries of user like a chat-bot using LLM's and GPT. This was achieved with the fine tuning of Mixture of Experts which addresses the issue of performance degradation and computational costs, which builds to be a sparse Large Vision-Language Model. The baseline of our model is compared with other Large Vision-Language Models to test it's efficiency.

Keywords: Generative-AI, Large Language Models (LLMs), Generative Pre-Trained Transformers (GPT), Generative Adversarial Networks (GAN), Large Vision-Language Models (LVLMs), Mixture of Experts (MoE).

CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1
INTRODUCTION

Generative AI, also called generative modelling in other words, is capable of producing a variety of different types of content,which includes text, images, audio, and synthetic data. It is one type of AI technology. The primary purpose of generative AI is to produce new data that closely resembles the training set. The outputs we received from the generative AI models can be identical to the human-generated content.

Generative AI uses the patterns it's learned from data to dream up new content that feels real. Generative AI is mainly focused on creating entirely new content. Generative AI models are built on quite large datasets of existing data. Once the generative AI model is trained, it can predict what elements are likely to come next in a sequence. This allows it to generate entirely new content that follows those same patterns.

Now let's discuss in detail the real-world applications in the field of generative AI.

- It can assist artists,writers and musicians by creating drafts, variations, or inspiring new ideas.
- It helps in generating variations in the existing product designs, exploring possibilities, and optimizing them for functionality.
- It helps in creating synthetic data and also helps to train other AI models more effectively.

Generative AI models, also known as foundation models, are designed to generate different types of content, namely text and chat, images, code, and videos. Some of the models are generative adversarial networks (advanced deep learning architecture), diffusion models, variational autoencoders, and flow models.

Generative AI has adverse effects in numerous fields. Some of the fields include art and creativity, design and fashion, healthcare activities, financial activities, manufacturing and engineering, gaming and entertainment, language and communication, climate and environmental sciences, educational purposes, security, and defense. It is also mainly useful in the medical field, very prominently. Some of the medical field areas include medical imaging, drug discovery and development, personalized medicine,

medical simulation and training, healthcare optimization, medical research, and innovation.

Large Language Models (LLM) are machine learning models that are useful in recognizing and generating text. LLMs are trained/instructed on huge sets of data. It is a type of AI Program. LLM uses machine learning in such a way that it helps in understanding how characters, words, and sentences function together.It uses a machine learning technique called deep learning, which involves the deep analysis of unstructured data. LLMs can also be trained to do a greater number of tasks efficiently.

Perhaps one of the best-known uses of LLms is their ability to work as generative AI. Here, the user provides some prompts or asks a question, and AI is able to generate text as a reply or answer. For example, LLM ChatGpt is freely available online and thus can generate essays, poems, and some others in textual form in response to input from the user. Additionally, other uses include sentiment analysis, DNA research, customer service, chatbots, and online search. Some examples of real-life available LLMs include ChatGPT by OpenAI, Bard by Google, Llama by Meta, and Bing Chat by Microsoft. Additionally, GitHub's Copilot is also worth mentioning.

Generative Adversal Networks (GANs) are the framework for training generative models. These are a class of artificial intelligence algorithms.These models are trained to provide fresh data samples that bear similarities to a particular training dataset. GANs involves two neural networks

- Generator (Responsible for creating synthesized data samples)
- Discriminator (tries to distinguish between real data samples and fake samples generated by the generator)

GANs have a lot of applications across various domains, like image generation, anomaly detection, and so on.

- Image generation (generating realistic images of animals, landscapes, etc.)
- Image-to-Image Translation (Transforming images from one domain to another)
- Text-to-Image Synthesis (Generating images from the textual description)
- Anomaly Detection (Identifying anomalies in datasets)
- Super-resolution (enhancing the resolution and quality of images)

Vision Transformers (ViT) is an image classification model that gives us a transformer-like architecture over the provided patches of image. This output for each image is a sequence of fixed-size patches that are linearly embedded with position embeddings and run through a standard TRANSFORMER encoder. The sequence of vectors is then used for classification by including an extra learnable "classification token" in the sequence.

The applications for the Vision Transformers are really vast, it is possible to carry out almost all the tasks in computer vision with Vision Transformers, and the results will be very good and, in some cases, state-of-the-art. Some examples are: image classification, detection of the object, image segmentation, anomaly detection, synthesis of the image, cluster analysis, and autonomous driving.

Vision Transformer-based algorithms, such as DINO and no-label self-distribution, have already demonstrated expected properties on biological datasets. For example, on cell painting images and generated images, DINO vector representations were generated afterward by clustering them and exploring the morphological profiles in the feature space. The main versions of the Vision Transformers architecture have been implemented in Pytorch.

Large Vision Language Models (LVLMs) are a recent advancement in the artificial intelligence field. It combines the capabilities of both computer vision (CV) and natural language processing (NLP) within a single model. These LVLMs are designed to understand and generate both images and text content, enabling them to perform tasks such as image comprehension and captioning, visual question answering, and image generation from textual descriptions.

The architecture of LVLMs is typically based on the fusion of transformer models, which has shown us remarkable success in both computer vision and NLP tasks. Some examples of LVLMs are OpenAI's CLIP (Contrastive Language Image Pre-Training), Facebook's UNITER (Universal Image Text Representation), and Google's LXMERT (Learning Cross-Modality Encoder Representations from Transformers).

LVLM Models like LLaVA and MiniGPT-4, which use an image encoder and multiple visual projection layers, have already given encouraging findings in improving the LLM's visual interpretation capabilities. Other such models are BERT, which is bi-directional and transformer-based, and it knows the meaning of a word based on context, meaning that BERT can comprehend a word based on its preceding and succeeding words. GPT-if, is another one but uses the Transformer architecture – in this case, it is a collection of generative language models that can be pre-educated on enormous texts. Transformer models also have utilization of the self-interest mechanism.

Feed Forward Neural Network (FNN) is a fundamental type of artificial neural network where the information flows in one direction, i.e., in the forward direction.It flows from input layer through one or more hidden layers to the output layer. The network is called "feed-forward" because there are no cycles/loops in the connections between neurons. This may also imply that there are no feedback connections. These feedforward neural networks serve as the building blocks for a more complex architecture.

The Years of advancement in the field of feed-forward neural network date from 1958 and the advancements are happening up to date and upgrading based on the present scenarios

- Frank Rosenblatt first described a layered network of perceptrons in his book Perceptron in 1958. This network is made up of three layers: an input layer, a hidden layer, and an output layer.
- In 1965, first deep-learning feed-forward network was published by Alexey Ivakhnenko and valentin Lapa
- In 1967, Deep-learning network using stochastic gradient descent was published.
- In 1970, modern backpropogation method was published by Finnish researcher Seppo Linnainmaa.
- In 1990s, a support vector machine approach was developed by Vlasimir Vapnik and his colleagues.
- In 2003, Backpropogation networks is applied to language modelling by Yoshua BEnhio and co-authors
- In 2017, modern transformer architecture were introduced.

A wide range of machine learning tasks employ feedforward neural networks,which include identification of patterns, tasks involving classification, analysis of regression, recognition of images, and forecasting time series.These are able to represent intricate relationships in data and have served as the basis for more intricate neural network structures.

Multi-Layer Perceptron (MLP) is a type of feedforward artificial neural network characterized by having one or more layers of perceptrons which are organised in a series of interconnected layers. These are some of the most common and fundamental architectures in deep learning capable of learning complex nonlinear relationships. At the same time, although each neuron in an MLP only does a simple calculation, it receives inputs from all the neurons in the previous layer, each weighted with its connection strength, sums them all up, and then applies a non-linear honesty function. This honesty function is what allows MLPs to model nonlinear functions. Otherwise,

they would be completely linear models since, no matter how many layers of linear operations you stack, they would remain linear. The most popular non-linear functions are the sigmoid, the ReLU function, and the tanh.

The breakdown of MLPs include Architecture, Activation Functions and Training. These breakdowns are discussed in detail now.

- In architecture,there are input layers, hidden layers, and output layers, and in each layer, neurons are connected to neurons in subsequent layers, forming a feedforward network.
- In activation functions, neurons typically use nonlinear activation functions like sigmoid, tabh, ReLU, or softmax, and enable the network to learn complex mappings between inputs and outputs.
- To minimize a loss function, these MLPs are trained through the use of optimization algorithms like gradient descent. In addition, gradients are computed and the weights and biases of the network are updated via backpropagation.

MLPs are adaptable and powerful machine learning techniques that have a variety of different types of applications. Their capacity to recognise complex patterns facilitates activities ranging from picture categorization to financial forecasting. This MLPs also performs well in classification, regression, and pattern recognition, allowing systems to categorise, predict and interpret complex data.

Mixture of Experts(MoE) is a machine learning technique where multiple expert networks are used to divide a problem space into homogeneous regions. It is also a neural network architecture which is designed to improve the performance and flexibility of traditional neural networks. They use a gating mechanism. MoE enables the model to be maintained with far less computation.With the same compute budget as a dense model, we can significantly increase the model or dataset size. There are two primary components to MoE.

- Sparse MoE layers(This layer is used instead of dense feed-forward networks.)
- Router or Gate Networks (this determines which tokens are sent to which expert).

Now lets see some of the benefits and applications of MoE models MoE models are highly flexible and can adapt to a wide range of tasks by adjusting the number. MoE models can achieve superior performance compared to traditional neural networks and the modular structure of MoE

models allows for greater interpretability. The MoE models have large applications as it are successfully applied in various domains, including natural language processing (language modeling and machine translation), computer vision (object detection and image classification), reinforcement learning (robotics), and game playing. Overall, we can say that MoE models represent a powerful and versatile approach to neural network, offering improved performance, flexibility, and interpretability compared to traditional architectures.

Although most of the previous work has explored MoEs as a pretraining mechanism, the intrinsic motivation setting of interest is not necessarily confined to pretraining. In reality, MoEs' benefits are perhaps particularly well-matched to an instruction fine-tuning scenario. The data is frequently intentionally designed to simulate a variety of duties in an attempt to replace multi-task fine-tuning.

One of the main drawbacks of the MoE paradigm is that it introduces an extreme number of total parameters. Thus,we concentrate on the more practical situation of daily practitioners: even if MoEs can be efficiently applied to PEFT methods such as or , which modify a much smaller number of parameters , a major difficulty arises. Indeed, our variance in expectations is small, and we continue to operate in an increasingly constrained space characterized by several other optimization challenges in MoEs.

MoE tuning is the process through which one gets the parameters of the MoE model to be optimal so that the model works well for the given task condition. It is tuning the parameters of the expert network and gating mechanism so that, when combined, the model works the best. This implies that MoE tuning is necessary for MoE development since it is a process where the model is ensured to work the best for the given task conditions. This process course involves finking the expert network parameters and gating mechanisms, optimizing the hyper-parameters, regularization, and language model methods, and evaluating the test data.

Now lets discuss some of the large language models. In that, the most important one is the Mistral large language model. It is a groundbreaking development in the artificial intelligence field. It is a generative text model with 7 billion parameters.It is a decoder-based language model that employs several innovative architectural choices. These may include sliding window attention, group query attention (GQA), and byte-fallback BPE tokenizers. It is a specific generative AI model developed by a French company called Mistral AI. Mistral AI offers open-source access to its Mistral 7B model through Hugging Face. This allows developers and researchers to easily integrate the model into their projects and contribute to its development. Overall, Mistral LLM is a notable player in the LLM landscape, offering a powerful, multilingual option with open access for exploration and development.

The contributions of our work are discussed below, which made the Spectate GPT model come into life for the public to access and revolutionize the field of generative AI:

- For the Spectate-GPT model, we have taken an open-source large vision language model as the base model [1], which uses visual transformers, feed-forward neural networks, and multiple-layer perceptrons for image classification purposes. The reason we took this model as our base model is because it has been fine-tuned with a number of experts and has no interface for the public to access.
- Taking that as an advantage, our team decided to study the model and make an interface out of it so that everyone can use and benefit from the capabilities of the base model.
- The interface that we have provided is comprised of important components from the base model that can access the functionalities of the base model, like image uploading, image question-answering, and general question-answering.
- Our model's interface gives the user the option to upload the image and allows for seamless interaction with the chat interface in order to clear their queries.
- In addition to that, we have also provided options such as upvote, downvote, flag, regenerate, and clear history to our interface so that the user can provide feedback to the model, which will help in the training of the model.

Chapter 2
MOTIVATION

The main motivation for our project is the announcement of developing phases of the GPT-5 Vision, which is under the development of Open-AI, which is an AI research and deployment company which has given models like Chat-GPT and DALL-E to the public to utilise the Large Language Models (LLMs) and Generative-AI features for solving the tasks of everyday life with useful and meaningful suggestions. Not only Open-AI, Google has also stepped into the field of Generative AI (G-AI) by building their own Large Language Model called as Google Gemini, which has more features than the Chat-GPT free version and also said to be more accurate than the Chat-GPT.

Google is another heavyweight to keep an eye on this generative AI world, with its own Gemini model. Gemini, representing Google's leap into advanced text generation, is a language model in the mold of Chat-GPT but one that far surpasses that rival. It is distinguished for features such as enhanced accuracy and powerful multi functionality, now the third main brand in AI. To this end, weathering the great tribulations of natural language processing and machine learning, Google nurtured Gemini. Its success in text-based content production and understanding world events of all sorts owes everything to these sectors.

In the area of everyday use, these developments underscore both the potential and growing importance of generative AI models. These models are not only used for natural language tasks but are likewise capable, for example, of creating images and code from which it was the first time people were able to generate their own text Thai from an idea (GPT-7). As research continues and these models grow more complex, they open up new possibilities in various domains–from customer service to the arts—and promise a future world where AI systems conspicuously make it easier to work creatively.

An autonomous system where visual and language capabilities are seamlessly integrated is the new vision represented by GPT-5 Vision.In addition to mastering and generating text-based instructions, the advanced model is also able to interpret visual cues in real time and act upon them.At the same time, equipped with sophisticated sensors and linked to a robotic platform, the GPT-5 Vision can perceive the world visually, understand spoken commands, and perform tasks in an integrated manner, bridging slots

for artificial intelligence to physically interact.This advanced approach takes full advantage of the multi-modal capabilities of GPT-5 Vision.

By merging its vision analysis with natural language processing, the model can receive complex instructions from users in spoken language, see the context immediately through its visual environment, and answer back both verbally and mechanically. This means that users can talk to the machine in their own language, and the model can understand—even anticipating what they will say. Furthermore, at work, it can perform one task while carrying on conversations with other individuals in the area; this means that its multitasking reflects human cognition itself.

GPT-5 Vision's autonomous learning and adaptation are yet another factor that differentiates it from the rest. As the model interacts with its environment and with users, it improves its ability to understand tasks and produce intelligent solutions. This cyclical learning process then raises an individual robot's efficiency and versatility rather than merely counting the number of tasks that it completes over time. Thus, we find that GPT-5 Vision as a quantum leap in AI-driven robotics, leading to more intelligent and better integrated human-machine relationships.

With this much advancement if the Generative-AI field where it can be integrated to devices to perform physical tasks with the instructions generated using Large Language Models and GPT's, these information are then processed to the main brain of the robot and decode the instructions to perform tasks as instructed. This model uses live capturing of the surroundings and updates itself accordingly which made us think that this model can be very much useful for the blind people assisting them while crossing roads, performing daily tasks, try to understand what is going on in-front of them with a simple "What is going in-front of me?" question and get cleared instantly.

Not only for the blind but also help people who are physically challenged and staying in their homes these robots can assist them by performing tasks that they can't by just giving an instruction. This model if developed properly and surpassing all the challenges that will be faced in-front of it while launching the product for public use then this will be a game-changer in the tech industry where the latest advancements of software is meeting with the latest advancements of the hardware for a better and comfortable future.

While discussing all these points with my fellow teammates then we got an idea of "why don't we build something like this?" later when we were in deep discussion we finalized that we are in our undergraduate and domain is computer science engineering lets build something similar to the software of the model and not going into the hardware of the model i.e.,

building a robot to perform tasks and take inputs in the form audio like a human. Leaving that to the ground reality we finalized our project idea and came up with our model i.e., Spectate-GPT.

Our model basically works on the idea of GPT-5 Vision only concentrating on the area of Visual Questioning Answering and Image Comprehension with basic input format that is the image upload option, where the user has access to provide the image and can perform tasks accordingly, such as asking for a brief comprehension of the image, asking queries related to the image, and also clarifying their general doubts in a particular field.

Of course, there are many large Vision-Language Models out there and every model has it's unique ability to perform for example one model can only generate content that is present in the image but cannot be used as a visual question answering model, where as other model can perform as a question and answering model but cannot perform as an vision infused question and answer model which leads us to this problem where it can be solved only when both the models have to be mixed and work as the desired model which was discussed earlier.

This model is almost imposes a slight similarity with the GPT-4 Turbo version, where the user can perform the same actions like visual questioning answering and image comprehension, but the limitation with the GPT-4 Turbo is that it is not free and not an open-source model for everyone to use, and one must pay the amount that will be billed in order to use the model, which is not really that useful for just giving input in text format and getting output in text format. So, we thought we would build a slightly similar model with some inspiration from GPT-5 Vision by only using open source large vision-language models which are free and trained accordingly.

Now, our team has discussed on the drawback of the GPT-4 Turbo model, i.e., its pricing and the fact that not everyone can use it. So, we came up with the idea of making our model available to the public with no extra cost and completely free, as this is just part of something big and doesn't deserve pricing for its usage. So, to make this happen, we have found a free open source large vision language model and to that end, our group has designed a user-friendly interface for the public to use by connecting the model and the interface with extreme python programming skills and web development skills to make it more interactive and beginner-friendly so that even school-going child and old people can use it hassle-free. This is the main motivation of our project and the process of building it is further discussed in the chapters below.

Chapter 3
LITERATURE SURVEY

The groundwork of Generative AI rested upon the introduction of Generative Adversarial Networks (GANs) [2] and Variational Autoencoders (VAEs) [3] in the mid-2010s. GANs utilize two neural networks: the generator, which produces synthetic data samples, and the discriminator, which differentiates between genuine and generated samples. Similarly, VAEs acquire how to encode data into a deceptively discernible space, a latent space, and then decode the data back to generate new data samples. These developments set the stage for future improvements in Generative AI.

Over the past few years, the increased utilization of large language models such as GPT-3 [4] and later versions has revolutionized the field of generative AI. These transformers have been trained using significant volumes of textual data, enabling them to generate coherent and fluent human-like text. Despite being mainly designed for text creation, LLMs models have been fine-tuned for image generation, often utilizing prompt engineering and diffusion models to achieve these goals.

Generative AI has also enabled several new improvements in image classification, such as improved data augmentation and domain adaptation. Diffusion models [5], for example, gained the most prominence due to their premise: generating high-quality images from noise. These models enhance the robustness and generalizability of classification systems by generating a variety of novel and simulating images to train a classification system particularly useful in cases where a shortage of data is critical.

The GANS technique that has primarily been adopted is image-to-image translation [6], where the model translates pictures from one domain to another while maintaining key features. This technique is common in image classification, where rare data is generated to improve precision.

LLMs have improved image classification since they use natural language prompts [7] to produce clear and coherent textual descriptions of the images which can be further scrutinized to determine whether the image has fulfilled the given attributes.

The concept of LLMs has become particularly popular after the introduction of transformer models [8] and the subsequent pre-training of very large models like BERT [9] and GPT [10]. Both models have demonstrated exceptional capabilities in understanding and generating human-like text

because of their training on large amounts of text data. In particular, the release of GPT-3 [4] in 2020 by OpenAI sparked interest in large-scale language models, as this LLM model had more than 175 billion parameters, achieving superior results on various text generation tasks.

Nowadays, LLMs have been developed and explored for their applications to various domains, including image classification. One popular approach is prompt engineering [7], where the generation process of LLM is guided by natural language prompts. By providing an appropriate prompt, LLM can thus generate a textual description or caption of an image, which can be used as another model's input for image classification, likely improving its performance and interpretability.

In addition, some LLMs have also been explored for their applications in the field of multimodal models [11], which combine both language and vision capabilities. The capacity of such models to process the textual and visual data together allows for more efficient image understanding and classification, as the textual data can be leveraged for training LLM and then integrated with images [12]. However, LLMs face several challenges that continue to be addressed, such as the computational requirement, potential biased behavior, as well as the responsible development and deployment.

Generative Pre-Trained Transformer has undoubtedly been a game-changer in the field of natural language processing. It has disrupted the conventional understanding of how machines can understand and communicate with languages in astonishing ways. GPT is primarily architected on a transformer, a deep neural network that revolutionized natural language processing. Recalling the rapid draw of the transformer model, the transformer became a de facto for researchers and in the industrial setting because the transformer model demonstrated impressive natural language processing attributes through interaction, particularly while conversing [13, 14, 15].

As earlier stated, a key feature of GPT's architecture is the employment of self-attention mechanisms. The self-attention mechanism enables the processor to process an input sequence, irrespective of its length. It is designed for the following tasks: language models, classification tasks, and text generation [13], [15].

Naturally, because of the training process, where a model predicts the probability of the next word in a particular sequence of words given the prior words, the task is popularly referred to as language modeling. It enables the model to acquire a general language representation that can be specified for any particular job or specialization. The applications of GPT are multidisciplinary. For example, in the healthcare field, it was utilized to build artificial notes in EHRs that improved patient outcome prediction [14]. In research, GPT pre-training was used to raise data for research by generating

more data [16]. In various sectors, such as business and psychology, product classification and tone definition were applied. It has vast potential for multidimensional applications such as image classification.

Although GPT's applicability to image classification is less direct, it informs transformers' application, including their development, in picture classification [17].

Generative Adversarial Networks (GANs) has been under intense research in deep learning since their discovery in 2014 by Ian Goodfellow. The main purpose of GANs is to identify mistakes in monitored data, which has resulted in their rising popularity across various domains. GANs are founded on a two-network architecture that includes a generator and a discerner. The generator generates synthetic data that is similar to the authentic data, while the discerner investigates that synthetic information, which yields feedback to the generator to enhance its operation [18].

Several breakthroughs have been made in the evolution of GANs in recent years. For example, in 2017 [21] developed Wasserstein GAN to address the issue of training of GANs by employing the Wasserstein distance as the distance metric between the genuine and the synthetic data, which generated relevant loss value, thus enhancing the output of the model. Notable, GANs utilize the binary cross-entropy loss function. Thus, in 2017 [20] created Least Squares Generative Adversarial Networks which utilized the least squares loss function instead for greater training stability and distinct prospective applications [19].

GANs have been employed in various applications and are still growing. In medical imaging application, GANs have been utilized mainly in image creation, image to image interpretation, and image modification. In brief, [19] demonstrated the image creation capability of the algorithm in 2015. In scenarios studies, GANs were broadly and more experienced in series of time analysis uses such as forcasting time series, and time series exception examination. In 2020, discovered that uses of GANs were already explored and debuted a classification of discrete circumference GANs and constant circumference GANs.

Image interpretation obtains the uses of GANs and is contriving for evolution in the field of image classification. GANs are being utilized as a solution for data augmentation in image interpretation and for generating synthetic data to develop classification versions. In 2017, [22] explained the training technique for the GANs. The techniques are all trends in the use of GANs and have enhanced the field of image classification.

In recent years, large vision language models, which combine advanced natural language processing with computer vision techniques have

proven to be a valuable asset. A notable work in this domain is Vision-Language Pre-training via Masked Token Prediction by [23], which introduced VL-BERT, a vision-language model pretrained on large-scale corpora. This work represented a significant step forward in joint vision-language understanding, based on the masked token prediction task, where the model is trained to predict masked words or tokens in vision and language at the same time.

Subsequent models, such as UNITER by [24], have enhanced performance through novel cross-modal learning tasks. For example, with diverse training data, including captioned images from Conceptual Captions and Visual Genome, UNITER vastly improves task performance, such as visual question answering and image-text retrieval. Aside from the two-stage paradigm, recent works in large vision language models have focused on model size and scaling. For example, Aligning Cross-Modal Spaces for Image and Text Retrieval by [25]. proposed image and text embeddings in a shared space for effective retrieval. This work demonstrates that large-scale pretraining and fine-tuning greatly improve performance, as shown in image captioning and other multimodal retrieval tasks.

With regard to model applications, large visiona language models have established a strong record in image classification. For example, in ViLBERT by [26], a task-agnostic visiolinguistic model improves performance on VQA and Visual Commonsense Reasoning over the previous model.

VisualBERT by [27], proposed a unified model that reasons about the image content and natural language question, attaining state-of-the-art performance. Lastly, new uses for vision language models have been proposed in the literature. LXMERT proposed in [28] have enabled diverse tasks like VQA, image retrieval, and referring expression comprehension.

Overall, recent advancements in vision language models have made impressive progress in joint vision-language understanding and multimodal tasks. These improvements rely on sophisticated pretraining, large-scale models, and new cross-modal learning paradigms to achieve outstanding performance on diverse tasks. Further, vision language models continue to evolve, offering promising advances in multimodal AI research.

Mixture of Experts is a versatile machine learning framework that combines the outputs of several "experts" models to achieve superior predictive power when combined. A seminal paper in this field is "Mixture of Experts Networks" by [29], which proposed an implementable architecture for MoE in neural networks. This work allowed for the use of MoE in complex tasks where data is dynamically routed to one of several experts using learned gating mechanisms, simultaneously improving the framework's efficiency and prediction accuracy.

More recent advancements have expanded the capabilities of MoE. For example, Switch Transformer: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity by [30] proposed a variant of MoE within the standard transformer network, allowing for the training of the largest models to date using similar computational resources. This study adds to the previous despite the difference in architecture, emphasizing the versatility of MoE use across different neural network configurations.

In the context of image classification, MoE has a broad range of applications [31], used a MoE to combine deep convolutional neural networks of various depths, each with a specialization in different image features to improve classification accuracy.

Recently published research further expands the use of MoE in this domain, with [32]. finding a new MoE architecture using shallow embedding layers to achieve the state-of-the-art performance on image classification benchmarks. This recent endeavor illustrates the potential of using MoE for incorporating hierarchical representation into a model for improved performance. An example from large-scale image recognition is [33], who introduced an adaptive MoE which dynamically activates and combines relevance experts based on the input characteristics; this enhances the scalability in dealing with large datasets.

Other uses of MoE in image-based models include addressing network robustness and interpretability, as in the case of [34]. To summarize, Mixture of Experts represents a broad concept in machine learning that allows for improved performance across many problem areas, including image classification.

MoE is most effectively expressed in models using neural networks, with recent research showing MoE's capacity to increase model precision and performance. Additionally, it has been extensively used in image recognition to accomplish diverse expert-based image classification and hierarchy representation.

Instruction-tuning datasets are critical to the progression of the field of natural language processing [35], introduced a novel instruction-tuning dataset that researchers can use to pre-tune their models or to fine-tune their language models for language understanding tasks.

The instructions are drawn from various domains, supporting pre-trained models to perform better on specific tasks after fine-tuning [36], also conceptualized a benchmark dataset that consists of instructional texts and the tasks they associate with, and this dataset is specially designed for instruction-based reasoning and inference [37], contributed to the establishment of instruction-tuning bodies by presenting a dataset that focuses on

knowledge-based instruction understanding. The dataset is multimodal, that is, it consists of textual instructions and visual representations, making the tasks more challenging as they require models to perform instructions based on both modalities.

Human-computer interaction is highly reliant on the competence of the models to understand and respond to the instructions given by humans [38], constructed and curated an instruction-tuning dataset that consists of instructions from real-life situations [39], were keen on creating an instruction-tuning dataset that they labeled an iterative continuous dataset curation.

All the instruction-elect datasets have played a critical role in ensuring a systematic and meaningful evolution of the field of computer science.

A related study by [40] proposed an innovative strategy to improve the training efficiency of deep neural networks, a novel approach to machine learning involving curriculum learning found that training efficiency can be increased by sampling gradually complex samples during the training process, which eventually improves model convergence and generalization. Focusing the difficulty of the training set, curriculum learning is an essential means of optimizing training and fastening the training process.

In a similar vein, [41] explored the application of meta-learning to optimize training in limited sample few-shot learning conditions. Their approach is based on meta-learning, in which the optimization of the model structure and learning is done depending on the training conditions and data. Meta-learning is instrumental in optimizing training in resource constraints settings.

Besides, [40] also emphasized the role of adaptive learning rate sh scheduling in training enormous neural networks. They propose a novel adaptive learning rate formulation that depends on the parameter gradient norm. Adaptive learning improves training quite dramatically, especially with the enormous neural networks associated with cutting-edge machine learning applications.

In another context, [41] looked into the regularization for optimizing training methods. They checked and demonstrated that regularization using L1 or L2 regularization and drop-out training enhanced generalization and model complexity control.

These examples of studies by [40] and [41] display the diversity of approaches and tools used in optimizing training methodologies. Curriculum learning, meta-learning, adaptive learning, and regularization are all part of training that improves performance, art-to-converge speed, and generalization.

Enhancing the resolution of images is a critical area in computer vision that spans many applications, including medical imaging and satellite imagery analysis. Recent work has addressed new ways to improve image quality by utilizing primarily deep learning techniques [42], introduced a new single-image super-resolution framework based on deep convolutional neural networks which implemented a multi-scale architecture with residual connections to efficiently train high-frequency details and rebuild a image with high-resolution given an input with low-resolution. By incorporating perceptual loss features and feature matching schemes were able to preserve the optimal quality of images.

In a similar approach, [43] investigated the ability to employ generative adversarial networks in image super-resolution tasks incorporated adversarial training objectives and perceptual loss functions to demonstrate that the proposed scheme allows for high-fidelity image resolution enhancement and detail preservation. The studies conducted in this suggest that GANs may serve as a primary method for increasing the resolution of images.

[44] is further advanced in the field by integrating attention and self-supervised learning into the deep learning image super-resolution process. Using self-attention mechanisms, the authors are able to capture the long-range dependencies within the image and maturely allocate resources in the high-resolution process using self-attention mechanisms to produce better results than traditional methods.

Furthermore, the research also discussed a novel perceptual loss that included feature matching optimized for image properties. All these advanced deep learning techniques achieved improved results and helped to achieve better resolution quality.

The second area of research in computer vision is image encoders' enhancements, which are conducted to improve models' performance and capabilities in image classification, object detection, and image generation tasks. Recently published works by [45] and [46] present unique ways of enhancing the effectiveness and efficiency of image encoders.

Specifically, [45] developed a novel approach to enhance image encoders through attention mechanisms and self-attention modules. The researchers focused on improving the structure of convolutional neural networks by incorporating self-attention mechanisms that help them to learn long-range dependencies and neighboring relations in images. They demonstrated that enhanced self-attention mechanisms in image encoders improve the ability to "attend" to important features and relations across separate spatial locations, which significantly improves the performance of fine-grained image understanding tasks.

Another work is [46], which introduce methods to enhance image encoders through multi-modal learning and cross-model representations. The authors suggest that image encoders have to learn features from a diverse set of information sources, e.g., text or audio data. Such multi-modal learning allows the image encoders to be more robust and adaptable to diverse and complex real-world situations. Recent advancements in encoding architectures also provide robust improvements in the image encoding performance through the application of unsupervised or self-supervised learning strategies.

[45] have proposed a self-supervised pretraining approach to image encoders, which allows creating models on unlabeled large-scale data by learning meaningful representations without direct human supervision. Both studies by [45] and [46] showcase the effectiveness of attention mechanisms, multi-modal learning, and self-supervised training in image encoder capabilities.

These improvements ensure that modern encoding systems are more sophisticated and flexible when addressing different input sources and task types in real-world applications.

Projection layers have become an essential component of numerous deep learning architecture, especially in neural networks such as transformers and models based on natural language and image processing tasks. The recent works of [47], [48], [49], [50] and [51]. have significantly contributed to a better understanding of the projection layers for the neural networks.

For example, [47] used a novel systematic design to integrate the projection layers within the transformers. The research focused on the structural models of the projection layer to systematically capture the dependencies existing between sequences. Particularly, application of different projection types helped to optimize the computation intensity of the projection layer while enhancing performance. Hence, as illustrated in their study, the model made significant progress in language understanding tasks.

Likewise, [48] studied the networks within the scope of CNN, which utilize the projection layers. Their focus on the special projection layer design to improve feature extraction and representation led to the network's increased accuracy in tasks such as classification and recognition.

More so, [50] worked with a reinforcement learning model with the adaptive projection layers in the policy networks. The research optimized the weight of the projection depending on different reward systems, making the policy generation more effective. As indicated by the improved learning performance, such networks can form the basis for developing large adaptive systems.

Multi-region understanding is an essential element in computer vision which involves studying how multiple regions interact and depend on each other within visual data.

[52] proposed a complete solution to multi-region understanding that utilizes graph-based modeling to capture interactions between multiple diverse regions in an image. By considering contextual dependencies and semantic connections between regions, this approach makes visual understanding systems less ambiguous and more resilient, enabling them to study intricate scenes and objects.

[53] expanded the definition of multi-region understanding to multi-modal data fusion, in which they examined how to mix it with visual and textual data to enhance region-based representations. Through multi-modal learning, found out that incorporating multiple data sources aided them better understand and interpret without context representations.

Simultaneously, [54] also furthered multi-region understanding by studying scalable modalities for efficient region-based analysis and feature extraction. Focusing on the scalability and applicability of this process, facilitated the development of such multi-modality analyses for real-world applications like image retrieval. This method helps in resolving how to work with large- scale and complex datasets.

Pixel-wise grounding, on the other hand, is the localization and identification of an object instance or semantical concept of interest at the pixel level within images. [55] studied pixel-wise grounding using deep learning and structured prediction processes. They argued that for tasks such as instance and semantic segmentation, pixel-level annotation and localization are crucial due to the need for pinpoint accurate pixel data.

Furthermore, [56] captured pixel-wise grounding from an interactive image editing scene understanding perspective. They employed interaction tools and interfaces to enable users to work at the pixel level based on their understanding of semantical concept analysis in achieving a comfortable and practical pass-through correlation.

Hard routers are an integral part of network infrastructure and have received considerable attention in recent research activities. Several studies by [57], [58], [59], [60], and [61]. Were found as seminal works that discussed the various aspects of hard routers, including architecture, performance optimization, security, and emerging trends. The present review aims to provide an overview of these studies and synthesize their findings to present the advancements and challenges in the area.

Most of the findings were found as highly applicable and relevant to the current trends related to network infrastructure and relevant technologies. For instance, [57] analyzed the architecture of hard routers and argued

that efficient packet processing and forwarding mechanisms were missing; the provided solution helped increased throughput while reducing latency to meet the demands of high-speed networking.

[58] followed with finding the best hardware-software co-design paradigm that would allow for the implementation of flexible and scalable hardware units to be utilized in various networking layouts using specialized processing units integrated with programmable blocks.

[59] examined performance optimization techniques for hard routers, focusing on traffic and congestion control by suggesting that intelligent routing and dynamic resource allocation may be beneficial for improved efficiency.

This was followed by [60] who provided power-efficient routing strategies to mitigate environmental threats and reduce the waste of energy using hardware optimized for power consumption and a power-aware scheduling algorithm.

Given the importance of security in the network infrastructure design, [61] undertook a detailed analysis of security threats and vulnerabilities unique to hard routers.

Key findings of their study include several potential attack vectors against hardware components and corresponding mitigation strategies, such as hardware-based IDS and secure boot. Strengthening the boundary of the router against potential malicious external attacks contributed to enabling network infrastructure to be more resistant to potential cyber threats.

Additionally, the works provide valuable insights into emerging trends in the hard router development. [59] authorized the potential of artificial intelligence and machine learning for smart traffic management, whereas [58] recognized the potential of programmable data planes for dynamic network function virtualization.

Finally, is the works explore the potential of development of routers using advanced materials, integrated photonics, and silicon photonics, which may further enhance scalability and bandwidth.

Works of [57], [58], [59], [60], and [61]. advanced the state-of-the-art in hard router development through investigation of architecture, performance optimization, security considerations, and emerging trends, which provide a foundation for further research and innovation in this critical area of network infrastructure. Continued growing of the need in advanced, high-performance, and secure, energy-efficient networking infrastructure would make their findings integral in future hard router development.

Task-specific Mixture of experts (MoEs) models have emerged as a powerful framework in machine learning, offering increased performance and efficiency across various tasks. Prior studies by [62], [63], [64], and [65] have significantly advanced the forefront of the development of MoEs by examining new MoE architectures, optimization approaches, and real-world applications.

The current review aims to summarize their works, distilling the main insights about the recent progresses and barriers in the realm of task-specific MoEs. To begin, the works by [62] and [63] present the architectures and design approaches of MoEs customized for specific tasks.

[62] proposed a novel architecture of a task-specific Mixture of Experts framework for natural language processing that utilizes hierarchical gating mechanisms to adaptively select expert models depending on the current input context. The architecture and design of this MoE framework enable flexibility and adaptation to various linguistic phenomena, yielding improved performance.

On a similar note, [63] focused on the design of a task-specific MoE framework for computer vision and introduced attention mechanisms and expert fusion approaches that help capture small-scale and detailed visual patterns. By incorporating task-specific expertise into the architecture, and achieved the best-in-the-world results in image classification and object detection in their paper.

Methodologies and findings next involved the necessary optimizations for developing and training effective MoE models, which are discussed by [64] and generalization techniques, which were analyzed by [65]. Specifically, [64] asymptotically studied optimization techniques for training MoE models customized for the given task. Here, the authors focused on working with large datasets and sophisticated model designs. Thus, it introduced methods that relied on parallel and distributed computation frameworks to train model experts simultaneously and accelerate convergence without sacrificing accuracy.

In turn, [65] studied regularization techniques that played a vital role in model generalization. In particular, it also developed methods for balancing model complexity and generalization by regularizing the model to improve real-world performance.

Beyond architecture and optimization, [62], [63], [64], and [65] investigate various applications and use cases. For example, [62] apply their architecture to sentiment analysis, machine translation, and named entity recognition, while [63] use their approach for image captioning and visual question answering. Such cases demonstrate the possibilities of using

task-specific MoE models for natural language understanding and visual analytics.

However, task-specific MoE models raise several challenges for the future. According to [62], [63], [64], and [65], the scalability of training algorithms, interpretability of predictions, and integration of context-dependent knowledge are the primary concerns. As a result, future work in this field will likely optimize training algorithms' performance, improve interpretability, and identify new fields of applications. Given the contributions of [62], [63], [64], and [65], future research on the task-specific MoE models will likely lead to innovative applications in machine learning and artificial intelligence.

Soft routers are an emerging approach to network routing that takes advantage of software-defined networking principles to improve flexibility and scalability in network management [66], [67], [68], and [69] have made significant contributions to numerous aspects of the field, from architecture to performance and applications. This document seeks to compile the works of these authors and distill the advances, trends, and prospects arising from them.

Regarding architecture and design, [66] developed a soft router architecture based on neural network models that permits the routing to be made on a real-time basis. Their approach includes the notion of neural routing tables which learn and evolve based on the experience of interacting with the network.

On the other hand, [67] proposed distributed soft router architecture, which is designed for wide-scale deployment on data center networks. They recognize the significance of optimizations in the processing of the data plane and the coordination of the control plane in achieving low latency and high throughput.

[68] focused on research that studied performance optimization for soft routers, particularly resource allocation and load-balancing approaches. They proposed routing schemes that respond to network congestion and link use in real-time by altering the path taken by packet transmission opportunities.

Additionally, [69] covered research that used reinforcement learning techniques to optimize routing policies in soft robots. Likewise, this method also learns, but it learns by trial and error. Therefore this method seeks to learn the most suitable means of routing and adapt network changes as necessary. In terms of applications and use cases, research by [66], [67], [68], and [69] covered a variety of options. [66] showed how their neural routing can be used to solve fast adjustment in traffic engineering and load balancing problems.

Likewise, [67] used their distributed soft router architecture to demonstrate its benefits on datacenter networks. Finally, this work addresses the challenges and the way forward. The primary issues that need to be resolved include scaling to a considerable extent, the question of security of routing infrastructure's strength and weaknesses, and integrating this system with various networking concepts. Next-generation research is projected to focus on those issues and explore new areas of design and operation. These works suggested providing a significant rise in soft router development as a concept that may usher in a fresh phase of development in network routing management.

Text generation and text enhancing techniques have become prominent research frontiers in the field of machine learning and natural language processing. Among the rapidly evolving approaches are EVE [70], Enhanced Variational Encoder, and MoCLE [71], Model-based Contrastive Learning for Enhancing Text Generation.

EVE uses variational autoencoder framework infused with principles of rein men restoration and adversarial training for developed text generation while MoCLE emphasizes text generating quality and diversification acquired through contrastive learning methods. This paper reviews the literature created by those doing texts, reviewing the impact and implications of the two-generation innovations and advancements stimulated by EVE and MoCLE.

EVE, the enhanced Variational Encoder, is a text generation by [70] that combines VAE principles with reinforcement learning to improve text generation and adversarial training to achieve quality and real-life likeability of generations.

The other model, MoCLE developed by [71], proposes a model-based contrastive learning framework for enhanced text generation. It was designed to ensure quality and diversity in crunchier ways. MoCLE relies on learning representations on latent space text data through contrasting, represented as positive and disguised samples to generate contrast-rich and diverse texts.

Comparative analysis between EVE and MoCLE unveils the differences in the two methods that contribute differently to the text generation quality and diversity. By virtue of that, numerous text generation fronts are opening up due to the arising options that different parties bases, such as the EVE and MoCLE developers. We can imagine, informed thinking for the future might reflect a higher rate of exploration, and hybrid may-roll may be the next approach base on the different options, and other possible areas that you might wish to invest your time for the future such as multi semantic rolling texts, stories, and more which os clearly studied in [70] & [71].

Chapter 4
DESIGN AND METHODOLOGY

Our model Spectate-GPT consists of a vision encoder which is a method infused in to the visual transformers that are employed for performing vision tasks, introduced by google for image classification, which demonstrated to be more accurate than the famous ResNet model which is used for image classification tasks. These Visual Transformers surprisingly do not convolution which is the main rule when it comes to computer vision tasks. The encoder works by converting raw input into input embeddings, these embeddings are numeric representation of the input because the computer only understands language of numbers. These numbers are generated according to the context and these contexts are extracted using positional embedding, which gives more weightage for an embedding compared to other embeddings of the image context as seen in figure 4.1.

From the figure we can clearly understand the working of the encoder where the combined embeddings are forwarded to multi-head attention blocks. The attention blocks specifies more importance to the useful embeddings which is given to the encoder, and later these blocks are followed by normalization layer and skip connection. This is followed by a feed-forward layer which is then followed by combination of normalization layer and skip connection, and by the result of this whole process we get encoded embeddings.

Later on in our model once the task of the vision encoder is done then it is passed onto a visual projection layer (MLP), the visual projection layer is a deep learning tool which is useful in the customization of semantic word embeddings in text analytics task. This helps in representing planar projected images which is most often used to change the virtual world based on user's perspective. The visual project layer that has been used in a Multi Layer Perceptron (MLP), in which the neurons are trained with back propagation learning algorithm, these are designed to approximate any continuous function and solve problems which can't be linearly separable. MLP's are used in prediction, recognition, pattern classification and approximation, but for our model we will be using MLP for image classification.

The working of MLP in image classification process is derived as multiple layer of neurons with an activation function and threshold value, the MLP takes multiple inputs from its one or more input neurons, the multiple perceptron layers are segregated as a single input layer, 1 or more hidden

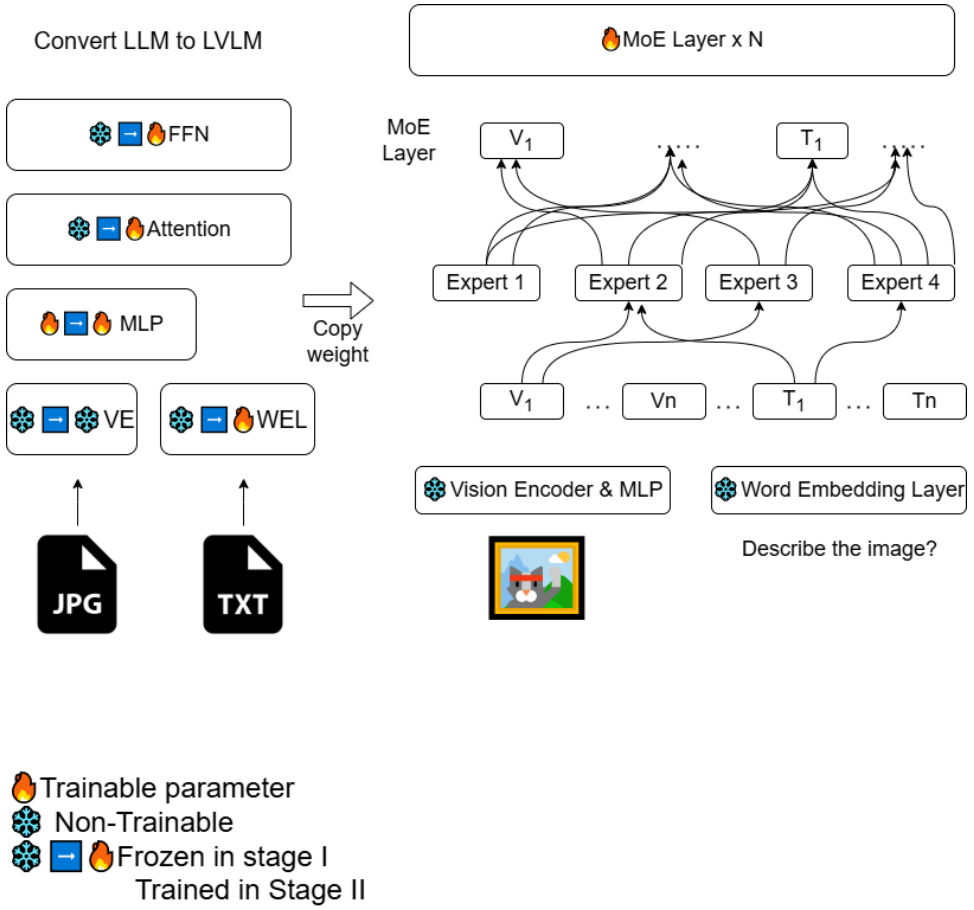

Figure 4.1: Working Interface of our Base Model

layers, and a single ouput layer of perceptrons, which flow in forward direction only. In this there is also a method known as Back-propagation where the MLP recieves feedback on the error in its results and adjusts its weights accordingly to make more accurate predictions in the coming future. This is used in many machine learning techniques like classification and regression, and in classification this has shown high accuracy.

Further in our model once the input is passed through the vision-encoder and MLP including those two comes the word embedding layer which maps word indices to vectors, this embedding is used in text analysis. It is a representation of a real-valued vector that encodes the meaning of the word in a way that the words are closed in vector space and expected to be similar in meaning. Later the model is introduced with multiple stacked Large Language Model (LLM) blocks which has the components like Self-attention, Add & Norm, FFN, and Add & Norm, which initially generates the image text caption, and when trained they generate response as seen figure 4.2, figure 4.3, and figure 4.4.

We will be now introducing the architecture of our model Spectate-

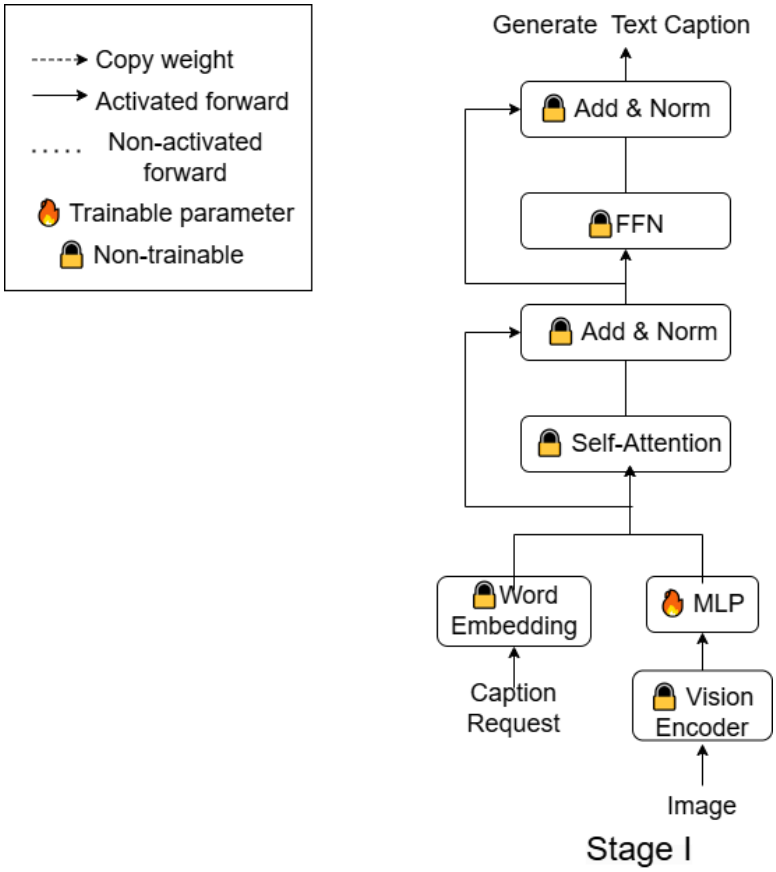

Figure 4.2: Stage-I Training of the Base Model

GPT in 3 stages in section 4.1, 4.2, and in section 4.3 we will be explaining on training the Spectate-GPT model, and in 4.4 we discuss on elaborating the training objectives of the Spectate-GPT model. These training framework strategy is completely explained in figure 4.5.

4.1 ARCHITECTURE OF SPECTATE-GPT

The Spectate-GPT model uses a base Large Language Vision Assistant (LLaVA) which is compared with the other models on the basis of the things that is discussed earlier, the FFN Factor is defined as the number of layers that are linear in the FFN part of the model which is another crucial component of the Transformer architecture, which is the fundamental component behind the majority of contemporary language models and vision-language models.

This factor determines the number of layers or the complexity of layers in this component. The number of linear layers stacked exerts the value of this factor. Smaller values denote fewer linear layers. Higher FFN Factor implies more linear layers are stacked. Higher values can be beneficial for

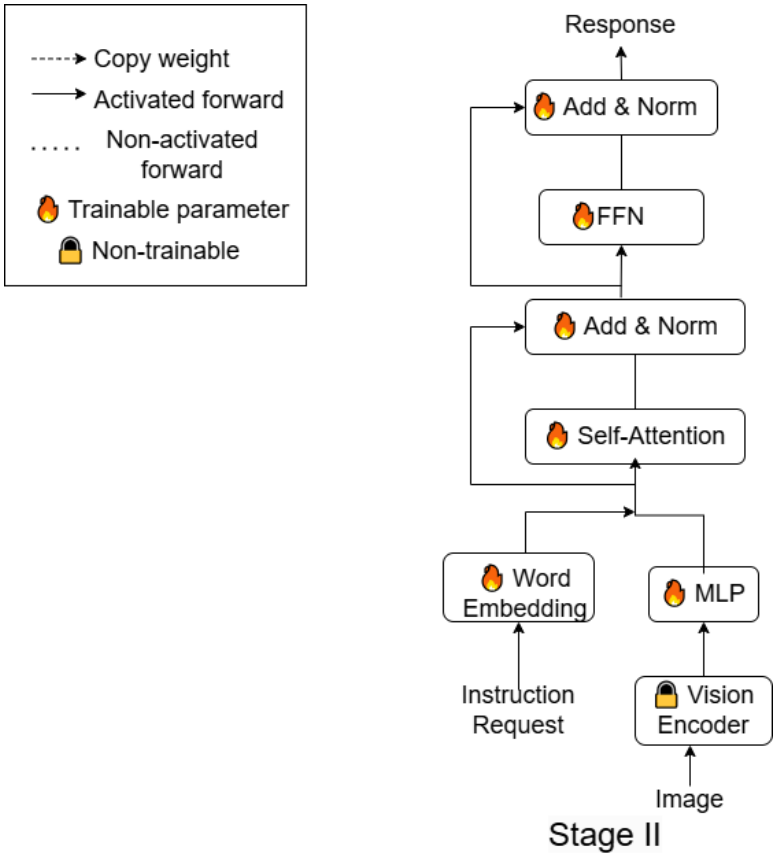

Figure 4.3: Stage-II Training of the Base Model

the model since they bring more expressive power to the model, which enables it to get more complex relevant data patterns and relationships. However, this also comes with the increase of computational time, due to increased complexity and memory consumption, potentially harming the performance and scalability.

Meanwhile, the 1.6B×4-Top2-version is a specific formulation of the dense foundation model which supports the sparse Mixture of Experts model, described in the article. In this context, the dense foundation model has around 1.6 billion parameters, which represents its initial capacity. The x4-Top2 part, however, represents the expert layer. The number 4 represents the total number of experts available in the expert network, while the "-Top2" part represents the number of active experts during the forward phase. The expert selection design is crucial for the success of the MoE architecture, which is the foundation of the sparse network.

By selecting only one active expert set, the model can fully leverage the computing resources to a set of robots, instead of splitting. Since a sparse network can host a high number of parameters, this reduction is possible. In aspect to language, and vision-language models, this design is ideal.

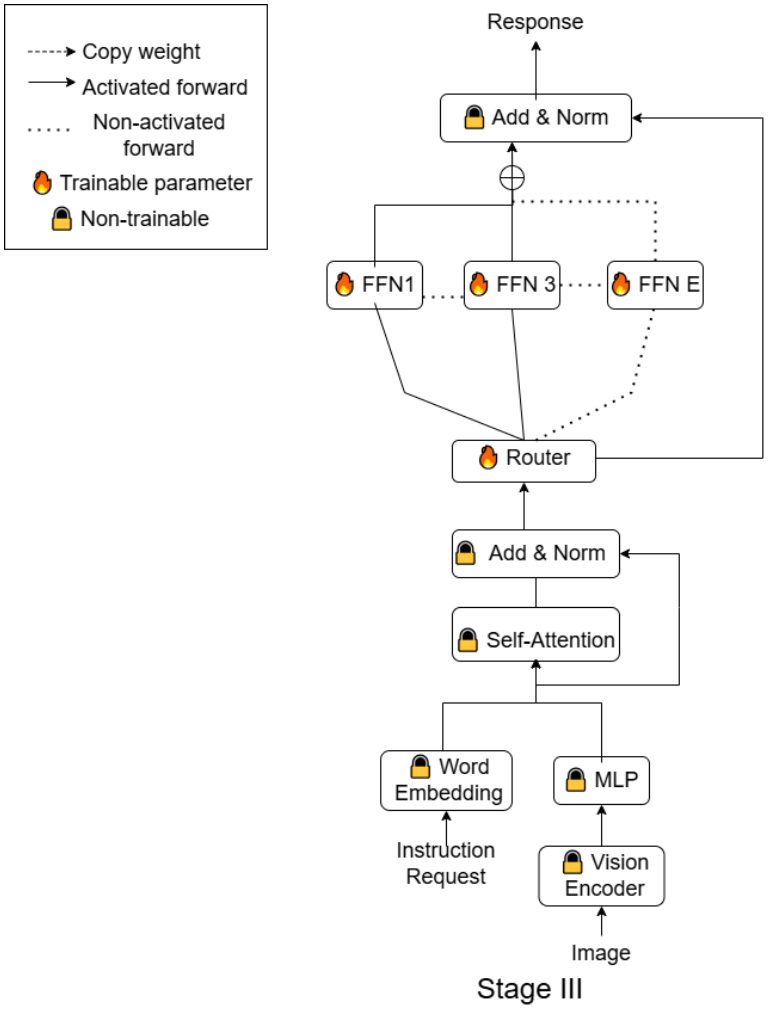

Figure 4.4: Stage-III Training of the Base Model

The top2 part of the configuration indicates the expert selection strategy, where the top two experts are selected for each representation or token. The expert assignment strategy is pivotal for the division of the labor concept to be preserved. This method assists the model by focusing its effort on understanding complex relationships in the representation data set, using insights delegated to different assigned robots as seen in table 4.1.

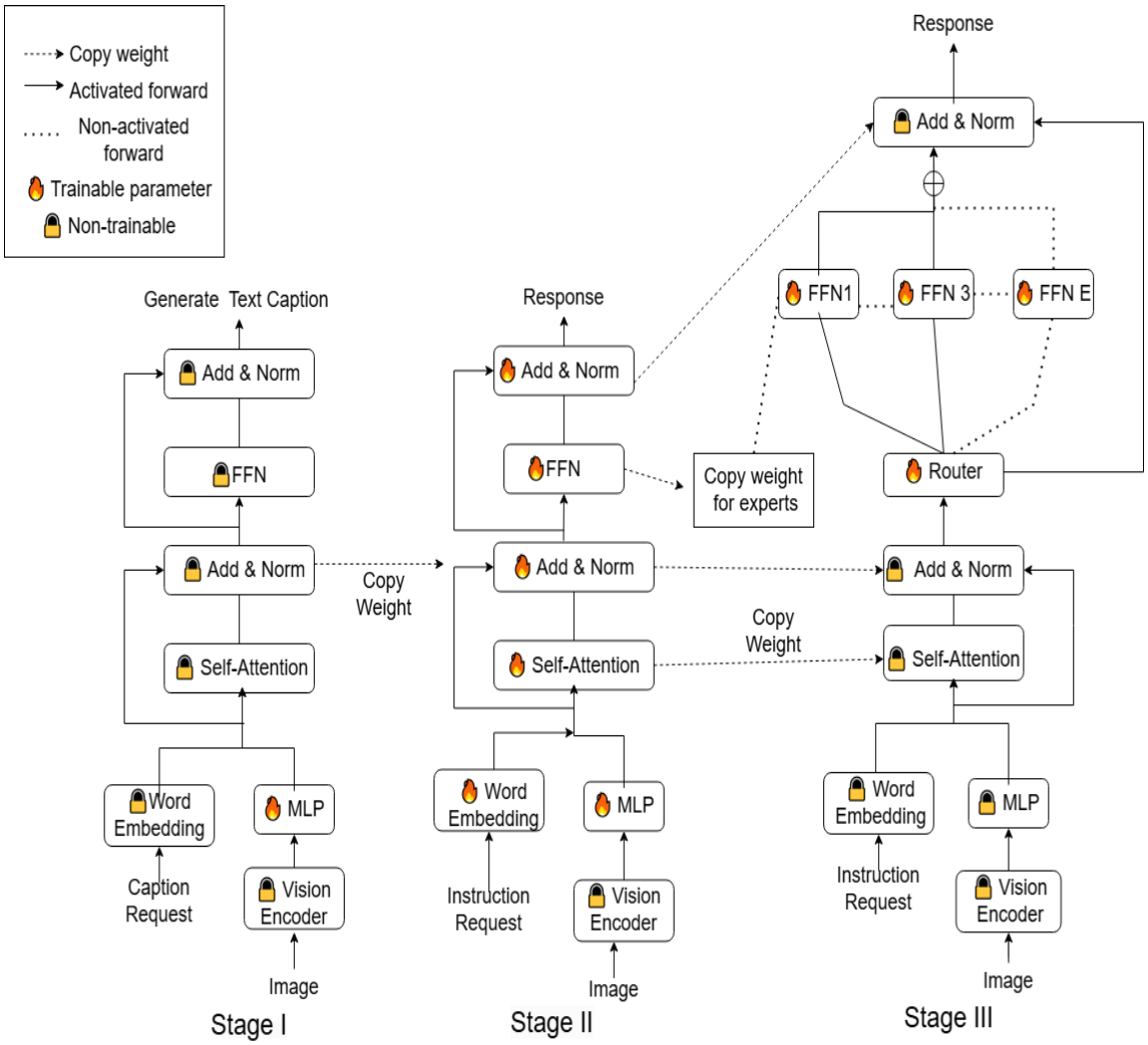

Figure 4.5: Complete Working Of The Interface

Table 4.1: Comparison of our Base Model Architecture with Existing Models

Name	Experts	Top-k	Experts Layers	Embedding	Width	Layers	FFN	FFN Factor	Heads	Activated Parameters	Total Parameters
StableLM-1.6B[72]	-	-	-	100352	2560	32	10240	2	32	1.6B	1.6B
1.8Bx4-Top2-Version	4	2	16	100352	2560	32	10240	2	32	2.0B	2.9B
Qwen-1.8B[73]	-	-	-	151936	2048	24	5504	3	16	1.8B	1.8B
1.8x4-Top2-Version	4	2	12	151936	2048	24	5504	3	16	2.2B	3.1B
Phi2-2.7B[74]	-	-	-	51200	2560	32	10240	2	32	2.7B	2.7B
1.8x4-Top2-Version	4	2	16	51200	2560	32	10240	2	32	3.6B	5.3B

Table 4.2: Some more Architecture Comparison of Our Base Model with Existing Models

Name	No. of Experts	Top-k	Experts Layers	Embedding	Width	Layers	FFN	FFN Factor	Heads	Activated Parameters	Total Parameters
StableLM-1.6B[72]	-	-	-	100352	2560	32	10240	2	32	1.6B	1.6B
1.6Bx4-Top2-Version	4	2	16	100352	2560	32	10240	2	32	2.0B	2.9B
1.6Bx4-Top2†-Version	4	2	32	100352	2560	32	10240	2	32	2.5B	4.1B
Qwen-1.8B[73])	-	-	-	151936	2048	24	5504	3	16	1.8B	1.8B
1.8Bx4-Top2-Version	4	2	12	151936	2048	24	5504	3	16	2.2B	3.1B
1.8Bx4-Top2†-Version	4	2	24	151936	2048	24	5504	3	16	2.6B	4.3B
Phi2-2.7B[74])	-	-	-	51200	2560	32	10240	2	32	2.7B	2.7B
2.7Bx4-Top2-Version	4	2	16	51200	2560	32	10240	2	32	3.6B	5.3B
2.7Bx4-Top2†-Version	4	2	32	51200	2560	32	10240	2	32	4.5B	7.8B
OpenChat-7B[75]	-	-	-	32000	4096*	32	14336	3	32	6.7B	6.7B
7Bx4-Top2-Version	4	2	16	32000	4096*	32	14336	3	32	9.6B	15.2B
7Bx4-Top2†-Version	4	2	32	32000	4096*	32	14336	3	32	12.4B	23.7B

In the table the base model is evaluated accordingly using no.of experts, top-k, Experts layers, Embedding, Width, Layers, FFN, FFN Factor, Heads, Activated Parameters, and Total Parameters. Here, the Experts are typically an individual neural networks or models which are trained on specific aspects or segments of the data,.

Next, comes the top-k which is used for avoiding non-sensible data our base model has been provided with top-k unlike the other existing models which refines the selection process by limiting the pool of next potential tokens to the most probable candidates. Now MoE layers, here MoE stands for Mixture of Experts which is a machine learning model containing collection of expert models and a gating network, each expert is specialized in different subsets of the data, which offers a more refined and specialized approach to problem-solving, in this the gating network component determines which expert to consult for a given input, which effectively directs traffic, ensuring most relevant experts to handle each query, this is the working Mixture of Experts.

Now, the embedding factor that has been evaluated is basically a high-dimensional vectors encoding semantic contexts and relationships of data tokens, which facilitates in nuanced comprehension by LLMs. It is also a way to store data of all types including images, audio files, text and documents, etc.) in the form of number arrays known as vectors. After the embedding now cones the Width evaluation is defined as the context size limit which represents the max sum (request+response).

The Layers evaluation is done on the basis of the number of layers that have been considered in the base model. Next, comes Feed Forward Neural Network (FFN) which is comprised of multiple fully connected layers that transform the input embeddings, by doing this the layers will enable the model to perform higher-level abstractions i.e., to understand the intent of user's text input. After comes the Head factor of the Large Language Model in which the idea is to fine-tune the additional layers on task-specific data to adapt the general language understanding capabilities of the LLM to match the requirements of the specific task in hand.

Now, comes the parameter factor which is comprised of two different classes they are 1) Activated Params, and 2) Total Params. Here, Activated params indicate the number of parameters that are activated during the process of training the base large language model where the parameters are comprised of different features of the large language model, like the ability of generating various types of text, ability to translate languages, and the ability to summarize text, and many more. It is always better to introduce more parameters to the model so that it can perform more complex tasks. The second params factor is the total params which is basically used for the comparison of the total number of parameters that are being taken by the

model and in that the total number of activated parameters are considered as activated params. These are the configurations that have been considered in our base large language model of Spectate-GPT.

Here, the image input type I that we are considering for our model is RGB image and is given by equation 4.1.

$$\mathbb{R}^{l \times b \times 3} \tag{4.1}$$

here, l and b are represented for origin resolution. Where the original resolution is the resolution of the image that has been uploaded by the user and it is taken by the model which is then processed with the framework of the model. Now, the Vision Encoder which process the input images will obtain a sequence of visual tokens represented as in equation 4.2.

$$K = [k_1, k_2, \ldots, k_N] \in \mathbb{R}^{N \times C} \tag{4.2}$$

where N represents the sequence length of the visual tokens and the N is given by

$$P = \frac{l \times b}{14^2} \tag{4.3}$$

The visual projection layer is utilized for mapping $Z \in \mathbb{R}^{M \times C}$ to $V \in \mathbb{R}^{M \times H}$, where H represents the size of the hidden layers of the Large Language Model (LLM), after this the text undergoes through a word embedding layer g and it is also projected to obtain a sequence of tokens $T = [u_1, u_2, \ldots, u_N] \in \mathbb{R}^{L \times H}$ in which the length of the sequence of the tokens is represented by L.

Subsequently, for the base large language model that we have considered for our main model, we perform concatenation for visual tokens and text tokens together and then they are fed into the large language model. Now, instead of that only training of the visual projection layer is done. Here, the Large Language Model consists a pile of Feed-Forward Neural Networks (FFN) and Multi-Head Self-Attention (MSA). Multi-Head Self-Attention is described as the key component of Large Language Models to enhance the ability of the models to capture complex relationships and dependencies within sequences of data, such as words in a sentence or tokens in a document.

Later, in the process Layer Normalization (LN) is done which is a crucial technique particularly in the context of transformers, to improve the training stability and accelerate convergence. This is a type of normalization technique which is applied to the activation's of each layer within a neural network. It also computes the mean and variance of the inputs across the features and normalizes each feature independently. The main intention behind layer normalization is to reduce the internal co-variate shift

within the network by normalizing the inputs of each layer. Now, along with the Layer Normalization another process is performed i.e., Residual Connections which is also known as skip connections, are a fundamental architectural component used in Large Language Models which facilitates in training many deep networks. These are introduced to address the problem of vanishing the gradients in deep neural networks and to make it easier to train deeper models effectively. This adds the original input layer to its output whose working is defined as the input passes through a block of layers and then the output is generated. These methods are applied within each block in [76, 77].

Which is displayed in equations 4.4, 4.5, 4.6, 4.7.

$$x_0 = [t_1, t_2, \ldots, t_N, \ldots, u_1, u_2, \ldots, u_N], \tag{4.4}$$

$$x'_l = \mathrm{MSA}(\ln(y_{n-1})) + y_{n-1}, \quad n = 1, \ldots, N, \tag{4.5}$$

$$x_l = \mathrm{MoE}(\ln(y'_n)) + y'_n, \quad n = 1, \ldots, N, \tag{4.6}$$

$$x = \ln(y_N) \tag{4.7}$$

4.2 MIXTURE OF EXPERTS FORWARD

Basically, Mixture of Experts (MoE) layer consists multiple Feed Forward Neural Networks (FFNs). The main concept of MoE is a sophisticated architectural approach used in large language models to improve models scalability and performance by utilising multiple specialized sub-models or ”experts” within a single framework. The MoE architectures are employed in various domains including Natural Language Processing (NLP) to handle complex tasks effectively. In this architecture, the model consists of multiple sub-models or experts which specialises in a particular aspect or subset of the input data. The experts are trained in such a way that they collectively contribute to the final prediction or output of the model through a gating mechanism that dynamically selects which expert (or combination of experts) to use for a given input.

During the initial phases, we replicated the Feed Forward Neural Networks from Stage II to form a group of experts $F = [f_1, f_2, \ldots, f_F]$. The Router, which is implemented as a linear transformation serves as an important component within the architecture of large language models. It functions a linear layer by using it to the input data which enables the prediction of probabilities associated with token assignments across various experts. The prediction mechanism is very much useful for the process of dynamically determining how each token should be allocated among various available

expert modules which are completely based on learned weights and activation's. With leveraging these probabilities routing strategy, the model optimizes the ability to distribute computational resources effectively and try to specialize in different linguistic aspects, which will enhance the overall performance and adaptability in complex language tasks. So, introducing a router that can predict the probability of the tokens which are being assigned to each and every expert that is in the model. Which is formulated as seen in equation 4.8.

$$S(x)_i = \frac{e^{f(x)_i}}{\sum_{j=1}^{F} e^{f(x)_j}} \tag{4.8}$$

The routers are operated with the generation of weight logits given by $f(x) = W \cdot x$, where $W \in \mathbb{R}^{D \times F}$ representing the trainable parameters linking the input x to the expert outputs. Each tokens distribution among experts is determined through softmax normalization of these logits, ensuring probabilistic assignment to E available expert modules. Consequently, the top-k experts with the highest probabilities are selected for further processing, facilitating a weighted aggregation based on softmax-derived probabilities. This strategic routing mechanism optimizes computational efficiency and allows specialization across diverse expert domains within large language and vision language models as seen in equation 4.9.

$$\text{EM(Experts-Mixture)}(x) = \sum_{i=1}^{K} S(x)_i \cdot F(x)_i \tag{4.9}$$

4.3 MIXTURE OF EXPERTS (MOE) - TUNING

The MoE-Tuning phase of the model is divided into three stages where at each and every stage the process that is being performed is explained in detail.

4.3.1 Stage-I

In this phase, the primary aim of our group is to integrate image tokens into the Large Language Model (LLM) framework, enabling the model to comprehend visual content within the context of understanding of the language. In-order to achieve this integration we will be using a Multilayer Perceptron (MLP) to map the tokens of image onto the input space of the LLM, which treats image patches as surrogate the tokens containing text. This transformation facilitates training the LLM to generate descriptive captions for images, increased utilization of its capability in language processing. Importantly, the use of Mixture of Experts (MoE) layers is deferred during the phase of adaptation within the LLM. This approach fosters the seamless integration of visual and textual modalities, which increases the

model's ability to interpret and describe the visual content effectively. The fine-tuning in stage-I is displayed in figure 4.2.

4.3.2 Stage-II

In stage-II, fine-tuning using multi-modal instruction data which will serve as pivotal strategy to argument the capabilities and fine-tune for the controlling of expansive models within the realm of large language and vision language models. This technique involves adapting pre-trained models by leveraging diverse datasets that combine textual and visual information, thereby enriching the model's understanding and enabling more nuanced control over its output. Then by incorporating multi-modal instructions during fine-tuning, as these models can do better integration and can process complex data representations, which will ultimately enhance the performance of the across various tasks and domains. This strategic adaptation enhances the model's adaptability, robustness, and effectiveness during the handling of intricate real-world scenarios involving both language and visual content. The fine-tuning of the Mixture of Experts in stage II is clearly depicted in the framework which is represented in figure 4.3.

4.3.3 Stage-III

In the initialization phase of the modelling approach, we have adopted a strategy which involves the replication of Feed-Forward Network (FFN) multiple times to establish the foundation for initializing the expert modules within the model. By introducing both image and text tokens into the Layers containing the combination of Experts, which is an important component known as the router computes the correlation of weights between each token and the expert modules. The weight calculation process is used to determine which expert is activated for the processing based on the tokens that are being provided. Specifically, the router facilitates the selection of the top-k experts that will exhibit the highest match weights with the input tokens, whereas the remaining experts remain silent or inactive as seen in figure 4.4.

Once the experts taken on Top-K factor it gets activated and identified, the outputs from the experts are aggregated using the summation of weight mechanism which is influenced by the weights and calculated by the router. This approach ensures that each input token is effectively processed by very closely related experts, which will contribute to final overall output of the model. With the implementation of dynamic routing and aggregation strategy, our model, termed Spectate-GPT, establishes a framework with infinitely possible different paths. The architecture of our base model allows for a broad aspect of capabilities, which will enable the model to effectively leverage and integrate information from both vision and language models.

The base models architecture is designed to use the power of Mixture of Experts while accommodating the specific requirements of language and vision applications. With strategic initialization of the expert modules taken from the Feed-Forward network, where the model lays a strong foundation for processing image and text tokens. The important role of the router cannot be overstated, as it dynamically calculates and assigns matching weights between each input token and available expert models. The weight based selection method ensures that only the Top-K experts, which will exhibit the highest correlation with the input tokens, that are activated for further processing, while the remaining experts remain inactive.

The activation process of the Top-K experts lead to the generation of the outputs which are collected through a weighted summation process based on the weights of the router that are computed. This selective activation and aggregation strategy will for sure optimize model's efficiency and effectiveness in handling complex multi-modal inputs, which will pave the way for sophisticated language and vision applications, The base model's architecture includes a scalable and versatile approach, that is capable of adapting to various tasks and scenarios by utilizing its in-built flexibility and different connectivity paths.

The clear functioning of all the stages is displayed in their respective figures 4.2,4.3,4.4 and the overall connectivity of the stages is displayed in figure 4.5.

4.4 OBJECTIVE OF TRAINING

The training objective is divided into two kinds of loss they are **1) Auxiliary Loss** and **2) Auto-Regressive Loss**. They both are evaluated accordingly using a variable L_Total which is used during the training, the auxiliary loss $L_Auxiliary$ is an additional regularization term that helps the model to learn and optimize its desired properties or behaviors. The auxiliary loss can take various forms, such as incorporating certain linguistic or semantic constraints, encouraging better representation learning, or promoting specific architectural properties like load balancing.

On the other hand, the Auto-Regressive loss $L_AutoRegressive$ is the primary objective that captures the model's ability to predict the next token or sequence of tokens based on the existing content, which is a crucial requirement for language generation tasks.

Now, to maintain balance between the Auxiliary loss and Auto-Regressive loss, a balancing coefficient β is introduced. This co-efficient acts as a scaling factor for the auxiliary loss, allowing the model to prioritize the auto-regressive loss while still benefiting from the regularization effects

of the auxiliary loss. The choice of balancing coefficient value is crucial, because it determines the importance of the auxiliary loss during the training process.

By incorporating both the auto-regressive loss and auxiliary loss, with the balancing co-efficient, these large models can optimize for their primary language generation or vision-language for understanding objectives while simultaneously learning to satisfy additional properties. The multi-objective loss formulation enables the development of more robust and well equipped models that can tackle complex language and multi-modal tasks and is given by equation 4.10.

$$L_{\text{total}} = L_{\text{regressive}} + \beta \cdot L_{\text{aux}} \tag{4.10}$$

4.4.1 Auxiliary Loss:

In the context of large language models and large vision language models, the presence of multiple experts which will help in distributing the tokens across the experts to ensure efficient utilization of computational resources to prevent issues caused by the overloading of some experts.

To achieve balanced distribution, the authors of [78] proposed to incorporate a differentiable load of balancing loss into each Mixture of Experts layer. The load balancing loss encourages the experts to handle tokens in a balanced manner, preventing any single expert from becoming overwhelmed with a disproportionate number of tokens.

The difference in load balancing loss is a crucial component in the training process of the models, as it helps to optimize the routing mechanism that assigns tokens to appropriate experts. By minimizing the loss, the model learns to distribute the tokens across the experts evenly, which will lead to improved computational efficiency and scalability.

The incorporation of this load balancing loss is particularly relevant in the context of large language models and large vision language models, where the number of experts and complexity of the data can be significant. The main models often require massive computational resources, and balanced distribution of the workload across experts is essential and the auxiliary loss is depicted and calculated using equation 4.11.

$$l_{\text{aux}} = E \cdot \sum_{j=1}^{E} F_i . G_{j,} \tag{4.11}$$

The F represents the tokens in fractions of amount that are being processed by each expert e_j and g_j representing the probability of average routinge_j, which is given by equations 4.12 & 4.13

$$F = \frac{1}{k} \sum_{j=1}^{E} 1\{\text{argmax } P(x) = j\} \tag{4.12}$$

$$G = \frac{1}{k} \sum_{i=1}^{k} P(x)_i \tag{4.13}$$

4.4.2 Auto-Regressive Loss:

To enhance the output of the Large Language Models, our base model has employed a generative loss approach in an auto-regressive manner. The technique involves optimizing the models output by progressively generating a sequence $Y = [y_1, y_2, \ldots, y_K] \in \mathbb{R}^{K \times D}$ which includes both textual and visual information. The length K of the output sequence is derived from P (Textual) and D (Visual) inputs.

In this process, each element y_i of the output sequence Y is iteratively generated based on the previous elements, utilizing the auto-regressive modelling techniques. Our base model framework utilizes a combination of neural network experts to handle the complex interactions between image and text inputs, enabling the model to generate coherent and contextually relevant outputs. By using a progressive generation approach, the model refines its predictions over time, which will produce a sequence that captures the relationships and dependencies within the input data.

The auto-regressive generation process within our base model underscores the model's ability to include information from diverse modalities, such as images and text, which will produce meaningful and structured outputs. The inclusion of generative loss principles ensures that our base model optimally learns from both the visual and textual content, which will lead to the creation of the output sequences that exhibit high fidelity and relevance to the input data. With this methodology our base model advances the state-of-the-art in large language and vision language models, showing its proficiency in multi-modal synthesis and generation tasks. The auto-regressive loss function is derived in equation 4.14.

$$L_{\text{regressive}} = -\sum_{j=1}^{N} \log p_\gamma(y^{[P+j]} \mid V, T^{[:j-1]}) \tag{4.14}$$

Here, γ is a constant parameter which can be trained and also we can calculate only the loss for the text has been generated.

4.5 SOME MORE ARCHITECTURE OF THE MODEL

we have compiled a detailed table presenting additional variants of the base model architecture . It is important because the table provides dive insights into the architectural options, parameter counting, and other design considerations behind the proposed Mixture of Experts model. One of the crucial details covered in Table 4.2 is the method for counting the total number of expert parameters in the base model, and also the "†" represents that the model is equipped with the mixture of experts layer. The authors suggest a simple formula to calculate this value considering the number of experts that are being activated during the inference process which can be formulated as in equation 4.15.

$$\begin{aligned}\text{Total Parameters} = {} & \text{Embed} \cdot W \\ & + \text{Layers} \cdot (4 \cdot W \cdot W + W \cdot \text{FFN} \cdot \text{FFN Factor} \\ & +2 \cdot W) \\ & + W + W \cdot \text{Embed} \\ & + \text{MoE Layers} \cdot (E-1) \cdot (W \cdot \text{FFN} \cdot \text{FFN Factor} \\ & +2 \cdot W) \\ & + \text{MoE Layers} \cdot (\text{Width} \cdot E)\end{aligned} \tag{4.15}$$

Here, Embed represents Embedding, W represents Width, and E represents Experts. More specifically, at activated experts count equal to 2, the number of activated parameters is calculated as the product of the number of activated experts and the parameter count of the dense foundation model . Such parameter-counting approach is essential for combination of experts architectures as only a subset of experts is activated at a time for a given input. Thus, models benefit from sparse parameters enabling higher efficiency and memory savings compared to equivalently-capable dense models.

The table 2 also introduces the FFN Factor, that is, the number of linear layers used in Feed-Forward Network . These layers are responsible for introducing non-linearity and enabling the model to learn abstract representations from the input. The FFN Factor allows researchers to "tune" the model to achieve certain performance and efficiency characteristics.

Additionally, we also explicitly state the dimension of the experts that are hidden which is used for the keys and values in the attention mechanism. In the table 2, we indicate the dimension of these hidden states with an asterisk (*) to indicate that they use a 1024 dimension, which is a common choice used in many state-of-the-art language and vision-language models at the time of the research. The table also defines a novel model configuration illustrated as 1.6B×4Top2-Version, which physically represents a dense base

model consisting of about 1.6 billion as a base architecture for the base model.

The remaining part of the configuration; that is, ×4-Top2, defines that the Experts architecture will contain four trained experts, with two of them being activated during inference. The authors also use the symbol "†" to demonstrate that all the model's layer uses an combination of experts. This feature is critical as it allows the experts architecture to apply explicit experts at every phase of the experts architecture's computation, which is essential for securing critical specialization and efficient resource administration. Therefore, by summarizing all the architectural definitions and configurations using a tabular form, we expose a comprehensive perspective of the base model, thereby allowing it to be comprehended by other researchers and implemented users.

Table 4.2 therefore serves as critical information to people working with large language models as well as vision-language models as it provides a brief on how they optimized the experts architecture for multi-modal representations.

4.6 THE MODEL DESIGN

While going through so many existing models and doing a reviewed research on them my team has been able to see the implementation of those models in one way or the other. Well, luckily when we were still continuing our research we have seen one of the large language model and large vision language model has no implementation of it where the public can easily use the model on their own. So, we took the advantage of the situation and we tried focusing on building a Web-UI for the model so that everyone can use it and we have achieved it. The whole process of building the interface is explained in this section further.

The interface is completely build using Python Programming and some of the web development skills have for sure helped us down in making the model quite easily. The main interface is comprised of a photo uploading option for user, a chat box for user to chat with the model, and also we have provided the option for user to train the model on their own by providing options for the user like Upvote, Downvote, FLag, Regenerate, and Clear History.

The functioning of Upvote is given for the user to train the model if the model gives a response for the users query and the user gets satisfied with the response given by the model then the user will click the upvote button and the model assumes that the user liked the response given by the user and it tries to learn from the output given and later on it tries to give

output in the format that the user likes and will eventually learn from the outputs ans the model gets trained accordingly.

On the other hand, there is Downvote option which lets the user to intimate the model that it has given an inappropriate response for the users query, this will let the model to know what mistake it makes and learns from the mistake it made and try to rectify it further on when the user gets another response and the user gets satisfied by the response then the user will upvote the response and the model will try to respond further on with the satisfied way of the user.

The flag button can be used by the user if the user wants to eradicate the present conversation if he finds anything inappropriate or if they think they want to start conversation on another topic this will help the model to know the issue of the response and try to avoid it from further on which will make the model more accurate and reasonable from next conversation.

The Regenerate button works on regenerating the new response for the same query if the user doesn't get satisfied by the response that is given initially and later the model will understand the user didn't like the response and tries to generate another response and then the user can use the upvote, downvote, and flag options to let the model know what exactly the user is looking for and trains according to it.

The Clear History button is used to clear the history of the previous conversations a user had with the model and also if the user wants to start a new conversation without any related information to the original context then the user can use Clear History button to start a new conversation with new context and can get the response accordingly without any interference of the previous stored context of the model.

The Design and Methodology section of this report clearly explains the framework of the base large vision language model that we have used, the formulas that were used during the training of the model and also the stages of tuning is clearly discussed in the above section with suitable framework designs. We have also included table in this section to depict what kind of datasets, models, and training of the base model and comparing the features of the base model with the existing open source large vision language models. The equations represent the mathematical approach to run the model and execute it accordingly.

In further sections we will be discussing on the implementation part in which we clearly discuss on the ideas that have lead us to execute the model for public use. In the software/hardware requirements section we will be discussing on the software and the hardware that we have used to make our model come into life. In the results section comparison of the base model with the existing models, and also we will be displaying the screenshots of the model working in web browser along with some implementation part.

Chapter 5
IMPLEMENTATION

In this part of the report we will be discussing on the implementation part how the base model has been implemented by the source and the factors involved in the development of the base model and we also discussed on how we have integrated our Web-UI (Web - User Interface) by using the important components of the base model and built an user-friendly chat-bot with image uploading feature similar to Open AI's GPT-4V. The implementation part of the base model is described in detail further in this section.

5.1 SET-UP OF SPECTATE-GPT

This section explains the settings that are used for building the base model and discussing on the Data that has been used to train the base model and the interface set up details.

5.1.1 Base Model Settings

In the LLaVA-1.5 model [79] the authors have proposed a specific architectural design, where it incorporates several key components and configurations which will enable the efficiency and effective multimodal representation learning.

Firstly, the vision encoder that has been employed in our base model is the CLIP-Large [80], which is a powerful vision model pre-trained on large amount of image-text data. This encoder is responsible for the extracting meaningful visual representations from input images, which can then be integrated with the language representations learned by the language model.

The multimodal fusion component, which is referred to as the MLP (Multilayer Perceptron) consists of two linear layers with Gaussian Error Linear Units (GELU) activation function. This non-linear transformation allows the model to effectively combine and reason the visual and textual representations, that will enable the generation of more accurate multimodal representations.

To leverage the computational benefits of the Mixture of Experts (MoE) architecture, where they have employed an alternating replacement

strategy for the feed-forward networks (FFN's) in the model. Specifically, every other FFN layer is replaced with an MoE layer, resulting in the configuration where the number of Combined Experts Layers is half the total number of layers in the base model. The approach aims to strike a balance between computational efficiency and the models expressing behavior, which will allow the model to selectively allocate the computational resources to the most relevant experts for each input token.

Furthermore, the authors of the above references have set the balancing co-efficient α to 0.01, which determines the relative importance of the auxiliary load balancing loss that happens during the training process. This specific value is chosen to ensure that the model learns to distribute the workload evenly across the available experts while maintaining a strong focus on the primary auto-regressive loss objective.

By carefully combining the choices of the architecture that have been considered and along with those their configurations as well, Our base model is a well designed large vision-language model that leverages the state-of-the-art components, which incorporates computational efficiency through the MoE architecture, and balances multiple objectives through a carefully tuned loss formulation. This approach aims to achieve a strong multimodal representation learning and efficient inference, which will enable the modle to tackle a wide range of challenging vision-language tasks effectively.

5.1.2 Detailed Analysis of the Data

In this section we will be discussing on the data that has been used by various models and also the data that our base model has been trained on. The data we have specifically used for the training of the existing models and also for our base model is a dataset with images and their captions.

Our base model has been trained with diverse set of datasets for all the three stages of the tuning process as seen in table 5.1.

Table 5.1: Data Composition For Various Available Image datasets

Grouped Data	Used In	Source of Data	Sample
LLaVA-PT	Stage-I	LLaVA 1.5-558k	558k
Hybrid-FT	Stage-II	SviT-157l, LVIS-220k	964k
LRV-33lk, MIMIC-IT-256k	Stage-II		964k
LLaVA-FT	Stage-III	LLaVA 1.5-mix-665k	665k

In the first stage of pretraining, in [79] they use the LLaVA 1.5-558K dataset. This dataset consists of 558,000 carefully crafted examples, that has

been pre-trained on a vast multimodal data, allowing the model to learn rich representations capturing the complex relationships between visual and textual information. For the second stage, they have combined several state-of-the-art datasets to form a strong initialization for our base model.

The MIMIC-IT dataset [81] focuses on multimodal medical imaging and text data, enabling the model to learn rich domain-specific representations in the medical field. The LRV dataset features a large and diverse corpus containing real-world visual-language tasks, permitting the model to reason over highly complex multimodal input. The SViT dataset contains visual-language information from instructional videos, enabling the model to understand and generate multimodal text in the context of procedural tasks. Finally, the LVIS dataset comprises richly labelled visual data and lets the model learn strong features and detection capabilities.

Finally, for the third stage, our base model uses the same data streamline as [82] for the LLaVA-mix-665k dataset. It consists of a total of 665,000 examples and spans a large variety of vision-language tasks, domains, and modalities. The model fine-tuned to this comprehensive dataset to further improve the quality and adapt the model for real-world vision-language applications. This combination enables our base model model to leverage transfer learning, multimodal feature learning, and task-specific learning. Indeed, our base model used several datasets with various focus areas and tasks, enabling the model to see a wide variety of visual and textual information and generalize well to unseen tasks and domains.

5.2 EVALUATION ON UNDERSTANDING THE IMAGE

In this section we have keenly gone through every existing large vision language model with different versions of our base model and they have been displayed in table 5.2.

Table 5.2: Benchmark Comparison of Different models on Understanding of the Images

Methods	LLM	Activated	Resolution	VQAv2	GQA	VisWiz	SQA1	VQAT	POPE	MME	MMB	LLAVAW	MM-Vet
Dense Model													
I-80B[83]	L-65B	65B	224	60.0%	45.2%	36.0%	-	30.9%	-	-	54.5%	-	-
LLaVA-1.5[84]	V-13B	13B	336	80.0*%	63.3*%	53.6%	71.6%	61.3%	85.9%	1531.3	67.7%	70.7%	35.4%
Qwen-VL[85]	Q-7B	6.7B	448	78.8*%	59.3*%	35.2%	67.1%	63.8%	-	-	38.2%	-	-
LLaVA-1.5[86]	V-7B	6.7B	336	78.5*%	62.0*%	50.0%	66.8%	58.2%	85.9%	1510.7	34.3%	63.4%	30.5%
TinyGPT-V[87]	P-2.7B	2.7B	448	-	33.6*%	33.4%	-	-	-	-	-	-	-
MobileVLM[88]	M-2.7B	2.7B	336	-	59.0*%	-	61.0%	47.5%	84.9%	1288.9	59.6%	-	-
LLaVA-Phi[89]	P-2.7B	2.7B	336	71.4*%	-	35.9%	68.4%	48.6%	85.0%	1335.1	59.8%	-	28.9%
Sparse Model (Our Considered Base Model)													
1.6Bx4-Top2-Version	S-1.6B	2.0B	336	76.7*%	60.3*%	36.2%	62.6%	50.1%	85.7%	1318.2	60.2%	86.8%	26.9%
1.8Bx4-Top2-Version	Q-1.8B	2.2B	336	76.2*%	61.5*%	32.6%	63.1%	48.0%	87.0%	1291.6	59.7%	88.7%	25.3%
2.7Bx4-Top2-Version	P-2.7B	3.6B	336	77.6*%	61.4*%	43.9%	68.5%	51.4%	86.3%	1423.0	65.2%	94.1%	34.3%
1.6Bx4-Top2†-Version	S-1.6B	2.0B	384	78.6*%	61.5*%	40.5%	63.9%	54.3%	85.9%	1335.7	63.3%	90.3%	32.3%
2.7Bx4-Top2†-Version	P-2.7B	3.6B	384	79.9*%	62.6*%	43.7%	70.3%	57.0%	85.7%	1431.3	68.0%	97.3%	35.9%

Table 5.3: Comparison on Performance of models in Object Hallucination

Methods	LLM	Activated	Adversarial			Popular			Random		
			Accuracy	F1-Score	Yes	Accuracy	F1-Score	Yes	Accuracy	F1-Score	Yes
Dense Model											
mPLUG-Owl[90]	L-7B	6.7B	82.4%	81.6%	45.2%	85.5%	84.3%	42.1%	86.3%	85.3%	42.3%
MM-GPT[91]	L-7B	6.7B	50.0%	66.7%	100.0%	50.0%	66.7%	100.0%	50.0%	66.7%	100.0%
LLaVA-1.5[92]	V-13B	13B	85.5%	84.4%	43.3%	87.4%	86.2%	41.3%	88.0%	87.1%	41.7%
Sparse Model (Our Considered Base Model)											
1.6Bx4-Top2-Version	S-1.6B	2.0B	86.9%	85.7%	41.7%	85.3%	84.2%	43.5%	88.0%	87.1%	41.6%
1.8Bx4-Top2-Version	Q-1.8B	2.2B	86.1%	85.4%	44.9%	88.6%	87.7%	42.5%	88.7%	88.0%	43.0%
2.7Bx4-Top2-Version	P-2.7B	3.6B	85.9%	84.9%	43.2%	87.5%	86.4%	41.8%	88.5%	87.7%	41.8%
1.6Bx4-Top2†-Version	S-1.6B	2.0B	86.9%	85.6%	41.5%	85.7%	84.6%	43.0%	88.4%	87.5%	41.5%
2.7Bx4-Top2†-Version	P-2.7B	3.6B	85.5%	84.2%	41.9%	85.7%	84.4%	41.7%	87.9%	86.9%	40.6%

The evaluation in the above table is briefly abbreviated below

- Res (Resolution): It refers to the input image resolution. It is measured in pixels and represents the number of pixels in each dimension the image.
- Act (Activated Parameters): It refers to weights and biases of the model. It refers to the number of parameters that are active or non zero during the models operation.
- VQA (Visual Question Answering): It is widely used benchmark dataset for image QA tasks. It consists of natural language questions about images from the COCO dataset
- GQA(Visual Genome Question answering): GQA is another benchmark dataset for image QA tasks, focusing on complex, compositional questions about images. It contains questions generated using scene graphs, providing rich structural information about the objects and relationships in the images.
- VisWiz(Visual Wizard of Wikipedia): VisWiz is a dataset designed for visually impaired users, consisting of images taken by blind people along with natural language questions about those images. The task is to develop models that can understand the content of the images and answer questions to assist visually impaired individuals.
- SQAI (ScienceQA-IMG): SQAI is an image-based extension of the ScienceQA dataset, focusing on questions related to scientific images. It includes questions about diagrams, charts, and other visual representations commonly found in science-related documents.
- VQAT (TextVQA): VQAT, also known as TextVQA, is a benchmark dataset focusing on answering questions about textual content within images. It includes questions about signs, posters, and other text-based elements present in images.
- POPE (Performance Of Pretrained models in Extreme conditions): POPE is a benchmark toolkit developed to test the performance of pretrained language models, including LVLMs, under extreme conditions such as low-resource environments, noisy conditions, and adversarial settings.
- MME (Multimodal Model Evaluation): MME is a framework to assess the performance of multimodal models, including LVLMs capable of processing textual and visual elements.
- MMB (MMBench): MM-Bench is a multimodal research-focused benchmark toolkit that can assess models' performance when both text and image inputs are included.

- LLaVAW (LLaVA-Bench in-the-Wild LLaVAW assesses various tasks for LVLMs in real-world applications. Image knowledge, inquiry answering, and language generation are some duties covered in the real world using data sets obtained from many sources and situations.
- MM-Vet (Multimodal Verification and Testing): LLaVAW assesses various tasks for LVLMs in real-world applications. Image knowledge, inquiry answering, and language generation are some duties covered in the real world using data sets obtained from many sources and situations.

5.2.1 ZeroShot-IQA (Image Question Answering)

In the table 3, we can see that our base model, which is Large Language Vision Assistant that has been tuned by mixture of experts, a sparse model that introduces an expert parameter that uses a soft router based on the Large Vision-Language Model architecture in comparison, the existing models are classified as dense models that work with a dense structure while lacking sparsity and routing in the parameters of the MoE.

The activation for each of the model configurations was determined via comprehensive experiments on five image question-answering benchmarks that demonstrate robust retrieval on these benchmarks while delivering the same operation as the previous state-of-the-art LLaVA-1.5 model . For instance, our base model's - phi-2.7B $\times$ 4 - Version outperforms SQIE, which is an SQAI benchmark, by 2.7 percent on the SQIE benchmark. While the model implements 7 billion parameters for the LLaVA-1.5-7B, it only has 3.6 billion sparse parameters active. The base model has employed a sparse model implementation of methods such as Mixture of Experts architecture, which delivered exceptional computational efficiency.

On the other hand, LLaVA1.5-7B, which is a dense architecture, implemented 7 billion parameters. The dense model is less computationally efficient in comparison. However, the third-party benchmark is subject to human error if no guidance were provided during the experiments. This comparison would verify the third-party benchmarks used across the experiment. Moreover, IDEFICS-80B, which is an 80 billion dense vision-language model, is benchmarked for third-party comparison. Our Base Models - stable LM – 1.6B $\times$ 4 - Version performs the best on the VQA benchmark by a margin of 6.2% compared to LLaVA-Phi version.

These results validate the model's superiority in parameter efficiency and the Mixture of Experts sparsity in extensive scale-up cases because the sparse model activates its entire capacity.

5.2.2 Benchmark Model's Evaluation

In this section, we have presented a full-scale evaluation of our base model's multimodal understanding capabilities across four widely-used benchmark models. The benchmark tool kits are designed to challenge the model's capabilities in natural language questioning and open-ended answering and are most rigorous in testing multimodal reasoning and language generation capabilities.

Most importantly, we have benchmarked models on MMBench – a model aiming to benchmark models on multimodal reasoning ability with a diversity of domains and tasks discussed above. Base Models - Qwen-1.8B×4 - Version outperformed our re-implementation of Qwen-VL-7B by a large margin of as much as 21.5%. This is remarkable considering that Qwen-VL-7B uses higher resolution images, which should make its models better at the visual tasks. These results highlight the efficiency of the Mixture of Experts architecture in enabling efficient multimodal reasoning and generation.

Through improved inference, our base model can strategically turn off parts of the model that specialize in specific multimodal reasoning, ensuring that the resources allocated go to the experts best suited for the task are speed while maintaining or exceeding current best dense models. Additionally, the evaluation across several benchmark models provides a wide view of multimodal tasks tested across open-ended questions answering and evaluating the ability of the model to generate sensible enough language, given the visual context.

The results of the benchmarking that is displayed in table 5.2 show that sparse models with active layers can match or exceed the performance of a dense model with fewer active layer, illustrating that the Mixture of Experts approach is a viable pathway for making multimodal models of large scale. The comprehensive metric of the diverse set of multimodal benchmark models and the competitive performance of of our base model demonstrate the potential of their approach in the future of large transformer models and vision – or language models. Ultimately, by utilizing Mixture of Experts architecture's ability to choose the relevant access, Our base model can channel more computational capacity to the most relevant areas of the model, thereby enhancing performance with higher precision in multimodal tasks.

5.3 EVALUATION ON HALLUCINATION OF OBJECT

This section discusses on the evaluation that has been done on our base model along with the state-of-the-art models whether they can detect the image correctly, comprehend the image correctly, or try to answer the

questions based on the image accurately. Hallucination parameter is taken into consideration whether the model can evaluate the image right or it assumes that it is one image and give irrelevant response.

In our base model, they have used the POPE evaluation pipeline of POPE to evaluate the object hallucination performance of a query by using a method based on polling. However, object hallucination is the process by which a model generates or recognizes objects not present in the object images. As a result, it often causes errors and multimodal contradictions during the understanding and reasoning process. Furthermore, they have designed the POPE evaluation pipeline to determine how much an object generated or recognized by the model contradicts the object input image.

That is to say, the POPE evaluation pipeline uses different sampling strategies, including adversarial sampling, popular sampling , and random sampling , to present a set of object queries based on the sample to determine the model response . Finally, it's possible to determine how quickly the model started identifying more about a specific object from the random, adversarial, and popular sets and thus measure the object hallucination ability of the model and its image reasoning ability . In the table 5.3 the results show that our base model is the best model among all evaluated models that is, it exhibits the lowest object hallucination ability.

In conclusion, our base model generates objects from the given image, indicating its models' robust multimodal reason ability. Our base models - 1.8B×4 - Version is the best model with the best object hallucination in terms of percentage increase in comparison to LLaVA-1.5-13B. Indeed, our base models - 1.8B×4 - Version has given the best results with 0.8%, 1.0%, and 1.5% increase in adversarial random sampling, adversarial sampling, and popular sampling, respectively. In addition, the model only activates 2.2 billion parameters during the inference.

Additionally, we noticed that the "yes" ratio in our base model remained balanced in all the evaluated scenarios. This balance implies that the sparse model is able to provide accurate responses given the questions, meaning it is not over generative of objects nor under generative of objects in response to the questions. The strong results of our base model in the POPE evaluation indicate that the model is robust for multimodal reasoning applications and is able to generate outputs that are coherent and consistent with the visual input. With the help of the selective parameter activation concept of Mixture of Experts architecture, Our base model is able to balance the parameter activation, so that the necessary parts of the network are activated with respect to the amount and size of the object, which in turn helps to differentiate between hallucinated and non-hallucinated objects.

Furthermore, as the observed "yes" ratio in our base model responses was balanced, it means that the model has understood both visual and

textual domains properly and is able to provide accurate feedback. This balance is important for multimodal models since the model is unbiased and does not tend to overgenerate or under generate the objects, which makes the model output more correctly and interpretability. To sum up, the evaluation using POPE pipeline indicated that Mixture of Experts is effective to mitigate the object hallucination and at the same time is preprocessing-efficient and scalable due to the selective parameter activation and sparse parameter distribution, which allow it to achieve very competitive performance for multimodal reasoning tasks while the output remaining balanced.

5.4 QUANTITATIVE ANALYSIS

In this section we will be discussing on the analysis of our base model in the context of quality of the data that has been generated based on the imgaes and also the working efficiency of the experts is discussed in this section.

5.4.1 Distribution of Routers

In this section, we have made our base model perform a thorough investigation on the expert load distribution, modality preferences, and modality distributions of the 2.7Bx4-Top2 base models version when running on the ScienceQA benchmark. Such an analysis helps us to better understand the behavior of the Mixture of Experts architecture of our base model from the perspective of learning dynamics. The report the expert loads across Mixture of Experts layers as illustrated in the leftmost plot of Figure 5.1.

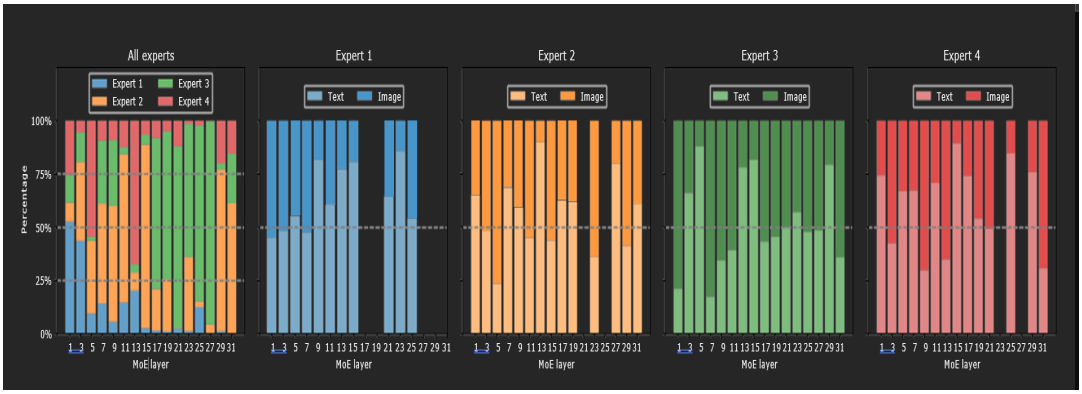

Figure 5.1: Distribution of Loading of Experts

Initially, as we observe, the expert loads are symmetric across all the Experts layers, which means that all the available experts are consumed equally during the model computation in the initial steps of learning. By

contrast, we report that, as the model goes sparser, the expert loads exhibit a noticeable pattern. Specifically, this reports a drastic rise in the load of expert 3 between layers 17 and 27. The extreme of such an imbalance is reached when expert 3 fully dominates the workloads, consuming almost all of the tokens in this range of layers. This behavior suggests that the Mixture of Experts architecture has learned to give a large portion of work to a specific expert for a particular layer range.

This section is probably specialized in performing a certain type of task or representation. In the process, we find that, for the first levels of layers, layers 5-11, experts 2, 3, and 4 always work together. This indicates that the experts mentioned work on the original representations of the input data. Strikingly, we find that expert 1 shows a substantial level of spatial learning, as the practitioner operates dominantly in the first few layers and thereafter discontinues his work. Hence, it can be inferred that expert 1 is a specialist in working on the initial representations of the data, while the others work on more complicated and higher-level computations figure 5.2

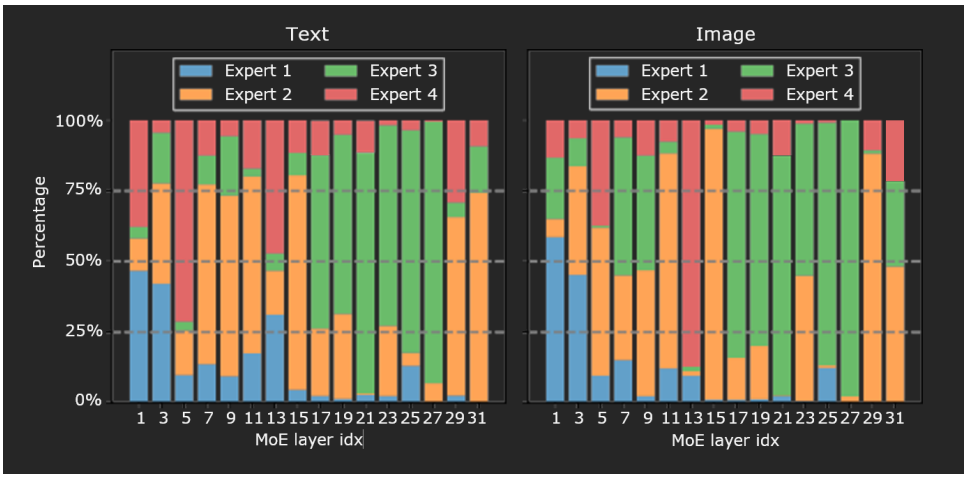

Figure 5.2: Modalities Distribution for Various Experts

Finally, we analyzed that the modality preferences of various different experts. In the four subplots on the right of Figure 5.1, we visualized the distribution of text and image tokens processed by each expert across the layers. The most striking discovery is revealed when comparing the routing distributions for text and image tokens. Specifically, as we can observe, the routing distributions for each expert in Figure are virtually identical. Therefore, one can claim that each expert in our base model is processing both modalities. More specifically, we can infer that when expert number 3 is actively operating in the layers 17-27 and is processing the tokens, the proportion between text and image tokens in this expert is similar. This suggests that expert 3 can process both modalities and integrate the information contained within the tokens simultaneously.

This distinct pattern is also experienced with the other experts, leading to the conclusion that our base model has learned how to use the expertise of any given expert in a modality-agnostic fashion. This modeling extends to the consideration of the modality distributions presented in Figure 5.2. Indeed, as one can see, the experts in our base model have learned a preference for certain combinations of modalities. However, the dependence once against disappears when the relationship between the expert and the modalities is observed.

Specifically, it is nullified, as when the expert 3 is dominating the active layers 17-27 in Figure 5.2, the proportions between text and image tokens processed by this expert are as equal. Therefore, it is clear that each expert is capable of processing the two modalities concurrently, without demonstrating any bias for one modality over another.

Analyzing the results presented by us on our base model concludes on the subject that the experts in our base model have adopted their own preferences and specializations, given the clear patterns of expert load balancing and processing of the modality. At the same time, however, each expert is able to process text and image tokens in a completely modality-agnostic manner despite its apparent specialization. As a result, this research demonstrates quite strong interaction and effective multimodal learning capabilities in our base model, and overall this discovery supports the ideas of our base model is a good architecture that possesses the necessary robust multimodal representation learning capabilities and can effectively and efficiently perform multi-modal input integration and reasoning.

The provided analysis thereby gave valuable insights into Mixture of Expert's inner working used in our base model it can help researchers and practitioners adopt a more ambitious and effective training strategy, architectural modification, and routing mechanism to further improve large language models and large-like models.

5.4.2 Pathways of the Token

In the extensive analysis of the Mixture of Experts architecture utilized in our base model, we have studied the individual behavior of expert on the token level to present its routing pattern and expert allotment specific to a token. This microscopic level of inspection would help to learn more about the working of sparsity and multimodal functionality of the model. For this analysis, the authors look at tokens performances across many other tasks. By examining the pathway of expert taken by each token, they discover the experts who integrate their corresponding representations.

To capture major patterns from the vast array of activated pathways, the authors employ PCA, a dimensional reduction method that was first

introduced by Pearson in 1904. By this method, the result presents the leading 10 expert pathways in UMAP visualized in the Major pathways taken by token indicates expert's trajectory. Here analyzing this minute level helps the author to observe a consistent behavior by the base model in passing text and images that are new to the network. Notably, in the later layer, model assign 2 and 3 experts to work on this tokens pattern alter.

This uniformity insinuates that experts 2 and 3 expertise on the structured representation of the multimodal inputs leave, causes this high-quality reasoning, and information integration modules. In contrast to experts 1 and 4 pathway, their behavior proves to take charge of the initialization face or else than the first layer. This observation insinuates that these experts have learned to express their expertise on the unilateral than both the image and text. This skill achieved supports feature learning, early fusion, or extraction.

Therefore, the token-level analysis highlights the specialization of experts and the division of labor by the model, a process enabled by the MoE architecture of our base model, which dynamically routes tokens to appropriate experts where their expertise is most needed, this makes it possible to process multimodal inputs effectively and efficiently at various stages of the computational pipeline. This specialization mechanism, combined with the sparse parameter allocation of MoE architecture, ensures that our base model is working efficiently by allowing the capacity of experts 2 and 3 to integrate and reason over higher-level multimodal representation and the capacity of experts 1 and 4 to be diverted into retrieving and processing low-level features from the raw input . Thus, these findings shed light to guide the behavior of sparse models similar to our base model in multimodal learning: the MoE architecture provides effective processing of multimodal inputs through the use of expert specialization, and their dynamic routing may contribute to improved performance over a broad range of vision-language tasks.

Additionally, we also visually present the token-level behavior of the experts in this section further below, which shows detailed routing patterns and token-to-expert assignments in the MoE architecture of our base model. Hence, the findings point the way to further exploration and refinement of sparse architectures for multimodal representation learning, producing models that are both more efficient and scalable for diverse vision-language tasks.

5.5 DETAILS OF THE TRAINING

Our experiments show that there is a set of hyperparameters that, for all our LLMs and VLM , ensure overall performance . One of the most critical

findings is that just one epoch is enough to train the model. Anything above one epoch overfits the model to the training set. To accommodate memory usage, we use a two-step batch size strategy and the second step.

The former allows us to quickly train models that are small relative to my available hardware, whereas the latter fine-tunes my models by providing computational capacity for hypothesis testing. Using allows us to have a good trade-off between keeping the image quality and memory efficiency. Stage counts ImageNet1k classes, 336x336 resolution . The fact that small LLMs such as qwen-1.8b can be trained on 8 V100-32G GPUs is good.

However, the models typically blow up during training when relying in fp16, the loss becomes NaN . That's why stages employ zero2 training2 . Since stage 3 of training does not support the deep speed library when the model is trained on three GPUs zero3, it should utilize zero2 offload . Offload allows us to detach parts of the model during the training process, which compared to zero3, assigns model parts and the data they use to three different GPUs. We use this strategy because during stage 3 training deep speed does not yet support training models with MoE in zero3 as seen in table 5.4. To benefit from zero2 offload , parts of the model are offloaded to the host memory, reducing the memory needed for training.

Our base model, a MoE-based VLM , can be trained on 8 A800-80G GPUs thanks to zero2 offload. Lastly, it was possible to use gradient checkpoint. Since gradient checkpoint only saves necessary computation to backward propagate, this dramatically decreases the memory needed when compared to storing everything to compute gradients.

5.6 BUILDING OF INTERFACE

In this section we Will be explaining the process of how we built the interface by incorporating the important components from our base large vision-language model which has been trained with combination of experts in different fields and with this part our whole web interface has come to life with full functioning so that the user can utilize the model accordingly.

The code imports necessary libraries such as argparse for command-line argument parsing, shutil and subprocess for file operations, torch for PyTorch, gradio for creating the web interface, fastapi for the web server, os for file operations, PIL for image processing, tempfile for creating temporary files, decord for video processing, and transformers for text streaming. It also imports custom modules such as conv_templates, Separator Style, Conversation from our base model, and various utility functions for the main interface for our base model and constants from the base model.

save_image_to_local(image): This function takes an image object as input, generates a unique filename using tempfile._get_candidate_names(),

Table 5.4: The Process of Training the Hyper-Parameters

Configuration	Stage-I	Stage-II	Stage-III
No.of Experts	-	-	4
Top-K	-	-	2
Deepspeed	Zero-2	Zero-2	Zero-2-offload
Data Group	LLaVA-PT	Hybrid-PT	LLaVA-FT
Resolution of Image	336x336 Pixels		
Encoder(Image)	CLIP-Large/336		
Feature Layer Selection	-2		
Projector(Image)	Two Layers(Linear) with GeLU activation		
Epochs	1		
Rate of Learning Error	1e-3	2e-5	2e-5
Rate of Learning Schedule	Cosine function		
Decaying of Weight	0.0		
Maximum Length of Text	2048		
Batch size/GPU	32	16	16
GPU	A800-80G x 8		
Percision	Bf16		

opens the image with PIL.Image.open(), and saves it to the temp directory using image.save(). It returns the path to the saved image file.

save_video_to_local(video_path): This function copies a video file specified by video_path to the temp directory using shutil.copyfile(). It generates a unique filename using tempfile._get_candidate_names() and returns the path to the copied video file.

generate(image1, textbox_in, first_run, state, state_, images_tensor): This is the main function that handles the generation of responses. It preprocesses the input image using the image_processor from the handler object, converts it to a PyTorch tensor, and appends it to the images_tensor list. It then calls the handler.generate() method with the preprocessed image tensors, input text (textbox_in), conversation state (state_), and other parameters. The generated response is split, appended to the conversation state, and returned along with updated states and Gradio components.

regenerate(state, state_): This function removes the last message from the conversation states (state and state_) and returns the updated states and a flag indicating whether the conversation history is empty.

clear_history(state, state_): This function clears the conversation history by creating new instances of conv_templates[conv_mode] and updating the Gradio components accordingly.

The final code uses the **argparse** module to parse command-line arguments, specifically the path to the model (–model-path) and the local rank (–local_rank) for distributed training or inference.

Based on the provided model path, the final code determines the conversation mode (conv_mode) by checking for specific substrings in the path (e.g., 'qwen', 'openchat', 'phi', 'stablelm'). It sets the device to 'cuda' for GPU acceleration, configures the data type (dtype) to torch.half for mixed-precision training or inference, and sets the bit precision (load_8bit and load_4bit) based on the model path. The code creates an instance of the Chat class from the base model package, which encapsulates the language model and its configurations, such as the model path, conversation mode, bit precision, and device.

The final code checks if the temp directory exists if not, it creates the directory using os.makedirs(). This directory is used for storing temporary image and video files during the application's runtime. The code creates an instance of the FastAPI class, which serves as the web application's backend.

The final code defines the Gradio web interface components, including a textbox (textbox) for user input, an image upload component (image1), a chat display (chatbot), and various buttons (upvote_btn, downvote_btn, flag_btn, regenerate_btn, clear_btn) for user interactions. It sets up event handlers for the buttons, such as generate, regenerate, and clear_history, which are called when the respective buttons are clicked. The generate function is triggered when the "Send" button is clicked, and it handles the input text, image, conversation state, and generates a response using the language model.

Finally, our code launches the Gradio web interface using demo.launch (share=True), which makes the application accessible remotely and allows users to interact with the language model through the web interface which can be seen in figure 5.3.

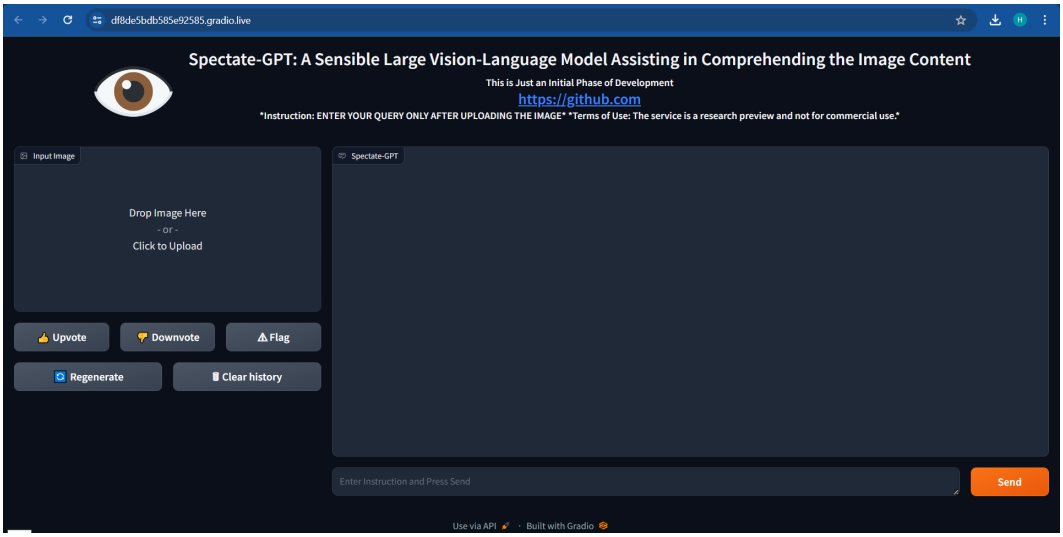

Figure 5.3: Basic Web Interface Design of Spectate-GPT

Chapter 6
HARDWARE/ SOFTWARE TOOLS USED

The Software we have used for the execution of the code which has web interface and also used for the integration of our base model with the interface is done using "Google Colab".

Google Colab, or Colaboratory, is an online platform free of charge that allows executing code without having to install any software. This product provides access to cloud computing services that have high computing potential, such as GPUs, and are widely used in machine learning. You can write, execute, and visualize your code using familiar Jupyter Notebooks. Furthermore, you can share your Notebooks with other users and work on them in real-time. Google Colab is an excellent platform for machine learning, data science, and coding .The interface will look the figure shown below.

Some of the functionalities in which Google-Colab acts as an important tool for us to use are:

- Cloud-powered Playground: Forget local software installations. Colab is entirely web-based, accessible from any browser. This makes it incredibly user-friendly, allowing you to jump right into your projects without setup hassles. But the real power lies under the hood. Colab grants you access to Google's high-performance computing resources in the cloud. This includes powerful GPUs (Graphics Processing Units), which are perfect for accelerating computationally intensive tasks like training machine learning models or processing large datasets.

- Easily Collaborate: Data science and coding projects are often done in teams. Recognizing that, Colab has made data science notebooks easy to share. Your colleagues can view, modify, and even execute your code at the same time, with real-time data exchange and collaboration. Colab makes it incredibly simple to come up with new ideas, work on a group project, or just help someone learn how to code.

- Bonus Functionalities: Colab goes beyond the core functionalities. It integrates seamlessly with Google Drive, allowing you to upload and access your data directly from the platform. Version control features ensure you never lose track of changes. There's also a built-in command palette for quick access to various Colab features and keyboard shortcuts for efficient coding.

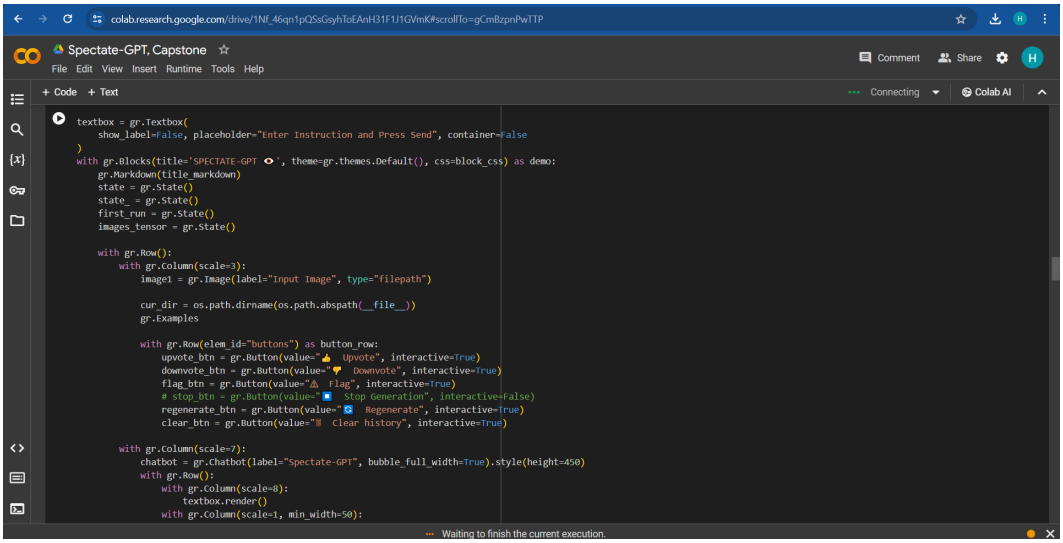

Figure 6.1: Image of Google Colaboratory Software along with code

The Hardware used for the implementation of our model is "NVIDIA A800 80GB". The NVIDIA A800 80GB is a professional graphics card specifically designed for high-performance computing tasks, not for traditional PC gaming. Lets discuss some of the key features and considerations in detail

- Performance Powerhouse: The A800 80GB is underpinned by the Ampere architecture, utilizing the GA100 graphics processor, and is an absolute powerhouse when it comes to handling complex workloads. It has 6912 shading units, 432 texture mapping units, and 160 ROPs, making it the ideal workhorse for parallel processing jobs such as scientific simulations or video editing. It has another 432 Tensor Cores, which have been designed to optimize machine learning and artificial intelligence computations. These substantially enhance the rate at which models can be trained and AI algorithms executed.

- Memory Muscle: The A800 boasts a high-bandwidth memory capacity of 80GB of HBM2e. This makes the A800 a perfect choice for larger language models and tasks involving very large image and data files where the use of smaller, less fit model variants is not possible. The HBM2e is coupled with an extremely high memory bandwidth of over 1.5 TB/s.

Chapter 7
RESULTS & DISCUSSION

In this chapter of the report our team has discussed and compared them on various evaluations of our base model along with some existing state-of-the-art models which are compared and displayed in the form of tables, and also the output of our model Spectate-GPT has been represented in the form of figures and also the evaluation of the experts across different parameters are represented in form of graphs and displayed as figures.

7.1 DISCUSSION ON STRATEGY OF TRAINING

Based on this analysis, we suggest a series of experiments in Table 7.1 to justify their approach to three-stage training of the base model. The purpose of these experiments is to determine the optimal modification of the Combination of Experts architecture suitable for adaptation for Large Vision-Language Models and also compatible with the retained knowledge and abilities of the pre-trained Large Language Models.

Table 7.1: Result generated based on Different Training Strategies of the Base Model

	Experts-MIX	Stage-II	Stage-III	GQA	SQAI	POPE	LLaVAW
(a)	✓	-	LV+Hb	58.4%	58.1%	81.9%	88.0%
(b)	✓	Hb	LV	61.5%	63.1%	97.0%	88.7%
(c)	×	LV+Hb	-	60.9%	60.2%	86.4%	86.3%
(d)	×	Hb	LV	60.9%	62.5%	86.9%	90.1%

In particular, the suggested three independent variants reflect different approaches to the initialization and training of the base model. The first variant, labelled as , reflects the straightforward identification, according to which the pre-trained classic Feed-Forward Network layers in the LLaVA model are replaced by the MoE layers and trained using the original second-stage script. Essentially , with this specific variant, the authors try to implement two massive modifications at the same time from an LLM to an LVLM and from a dense model to a sparse MoE architecture.

However, experimentation shows the performance of variant is the weakest, which implies the present dataset containing multimodal instructions is insufficient to carry both conversions simultaneously. The results

of this examination clearly indicate the complexity and difficulty of the question to adapt pre-trained models to new architectures and modalities while concurrently introducing sparsity and computation-efficient through the MoE approach.

Therefore, the authors suggest an alternative variant that involves another data collection activity. Specifically, dataset "Hybrid-FT" is gathered to allow the shift from the dense LLM to a first approximation of the LVLM on stage 2 of training. On stage 3, the LLaVA-FT is used to adjust the sparse base model.

In addition, we present a third variant, noted as, that is a natural comparator. Following this notation, the authors have extended the data used in the original LLaVA second stage to make sure the dense LVLM initialization on the same size of data is trained as the original base model. These experiments indicate that variant beats out both and , indicating that seeding an LVLM initialized with reasonable prior information which grants the model to switch quickly from a dense pretraining to a scattered MoE form.

Through this, the authors validate the principle driving their three-stage method, which evolves the pre-trained LLM to an LVLM, and then into a sparse MoE-LVLM model. This enables them to take the advantages of pre-trained LLM's learned knowledge and representations, and introduce multimodal capabilities and precision through the LVLM and then the MoE architecture. The transformation from LLM to LVLM serves as an appropriate model initialization for the resulting sparsification, enabling the model to better adjust to the performance characteristics of the MoE.

Additionally, the results demonstrate that having enough image and text data is essential to achieving the model's target performance on a model and dataset transfer task. Specifically, active performance distribution remains unknown and is a limitation of the current LLaVA.

Overall, by conducting these extensive experiments and analyzing the outcomes, the authors deliver substantial contributions in terms of the issues and aspects associated with adapting pre-trained models to new modalities and architectures. In addition, the introduction of a form of computational efficiency, such as sparse architectures or the MoE, to new models could provide valuable lessons. The three-stage training method ultimately offers an efficient way to tackle the former issues and develop powerful and efficient Vision-Language models on a larger, efficient, and scalable scale that takes advantage of the benefits of pre-trained LLMs as well as encompassing multimodality and computational efficiency.

7.2 RESULTS OF PARAMETERS TUNING FOR DIFFERENT SUBSETS

To study these trade-offs and identify the impact of fine-tuning different parameter subsets in the base model, the authors perform a series of experiments. The results in Table 7.2 inform the trade-offs outlined above, comparing the model performance with computational efficiency. One of the key notions studied in this table is the suitable performance of fine-tuning the FFN layers and the MoE layers.

Table 7.2: Result for Base Models Parameters Tuning of Different Subsets

Subset	GQA	VisWiz	VQAT	POPE	LLaVAW	Time
FFN Layer	61.5%	32.6%	48.0%	87.0%	88.7%	20h
All	61.3%	31.9%	47.6%	87.0%	88.1%	27h

In this study, the FFN training is presented as "FFN," and the All training represents all the model's parameters. The results reveal that fine-tuning only the FFN and MoE layers is sufficient to achieve satisfactory performance compared to the All approach. Importantly, this finding demonstrates that the primary representational strength of the model is enclosed within the FFN and MoE components.

Consequently, training the remaining components is unnecessary, resulting in waste since the model does not significantly benefit from the diversification from the initialization parameters. Additionally, the authors continue to measure an extensive computational advantage of FFN and MoE fine-tuning. On average, this process takes only 75% of the time compared to All training, which vastly reduces the time and resources required for deploying the model. Thus, based on the evidence, we decided to proceed with the FFN and MoE fine-tuning for the base model.

Overall, by conducting a detailed analysis of a targeted fine-tuning method for the base model, authors can minimize the computational requirements concerning large-scale MMLM models . Because representations and computations of the model freeze the rest of the pre-trained knowledge and its encoding in the rest of datasets, this can be used for a target task without catastrophic forgetting, a common phenomenon when fine-tuning models at the large scale.

Result of the research provides a useful substantial contribution to the growing number of efficient fine-tuning and parameter-efficient transfer learning methods concerning large language and VLM models and it also means that with identification of the most important parts used in

adaptation of a model to a task, computational overhead of loading whole pre-knowledge can also be minimized. Also, the work reveals general insights about the influence and value of different components in the complex models.

It shows that the key components influencing the model's RNN capacity for adaptation ability are FFN and MoE layers and what are the implications of multimodal reasoning and learning tasks. Ultimately, efficient methods of fine-tuning for the large language and VLM models not only increase the efficiency of the base model computation but also are a step closer for more sustainable solutions to large-scale models in efficient computing orientation and environmental friendly trends, as, for instance, in this paper.

7.3 DISCUSSION ON NUMBER OF EXPERTS COMSIDERED

Traditionally, increasing the number of experts within the Mixture of Experts (MoE) architecture is associated with better performance, as indicated by prior studies conducted by [93] & [94] However, our base model is tested on various configurations of the number of experts in a series of experiments Table 7.3 outlines the results of the ablation experimentation, comparing the performance of the single expert dense model with a sparse configuration that increases the number of experts.

Table 7.3: Result Generated considering the Number of Experts of Base Model

No.of Experts	GQA	SQAI	VQAT	POPE	LLaVAW	Time
ONE	60.9%	60.2%	48.3%	86.4%	86.3%	13h
TWO	61.2%	60.8%	47.0%	87.5%	86.5%	14h

Importantly, the authors ensure that the number of experts that are activated is the same regardless of configuration, equating the number of activated parameters . This design choice allows for a direct comparison of the models in the isolated aspect of increased sparsity and therefore expert specialization. The obtained results suggest that the sparse configuration with various experts outperforms the single expert dense model by a high margin.

More specifically, the sparse model leads to an increase of 1.1% on the POPE benchmark and 0.6% on the SQAI benchmark. This evidence overturns the intuitive notion that increase the no.of experts which directly makes the performance better. Instead, the multi-expert configuration with

sparsity and expert specialization is advantageous even when the total number of the activated parameter remains the same. The causes of the superior performance of the sparse expert configuration are multiple. First, multiple experts enhance expert specialization with a role in learning and representing solely specific portions of the input data. This way, computation can be efficiently allocated depending on which expert is assigned the most data.

In addition, the sparsity of the experts also encourages the model to learn more compact and efficient representations, which could potentially lessen redundancies and allow the model to better generalize to unseen data. This sparsity could also improve the forward pass computational efficiency, since only a subset of experts are activated for a given input. The findings are consistent with the recent trends in sparse architectures and parameter-efficient models showing improvements in performance.

Therefore, the base model could achieve equal or even better performance while reducing the overall computational requirements and memory usage with the use of sparsity and expert specialization. The result of the current study confirm the necessity of exploring unconventional architectural arrangement and challenging existing assumptions in the large language models and vision-language models field. Through an investigation of the interplay between the number of experts, sparsity, and performance, the researchers can gain novel insights and develop more efficient and effective models for multimodal learning and reasoning.

Furthermore, these findings are beneficial for developing more sustainable and efficient machine learning solutions in general. As the disruptions from larger language models and vision-language models continue, sparsity and expert specialization could help alleviate the computational and energy use burdens to create more resource-friendly and environmentally sustainable models. Overall, by integrating sparsity and expert specialization, the base model not only shows improved performance, but also open ways for more efficient and scalable models to handle multimodal jobs' growing complexity.

7.4 DISCUSSION ON THE NUMBER OF ACTIVATED PARAMETERS

In our base model which is a Large Language Model with Vision capabilities, we studied the effect of varying the number of activated experts within the MoE architecture. Here, by activated experts, we refer to the number of expert models selected from the ensemble to process an input. In this regard, we analyze a different strategy highly relevant to a "top-k" strategy as displayed in table 7.4.

Table 7.4: Results Generated considering the Activated Parameters of The Base Model

Top-k	VQAv2	GQA	SQAI	VQAT	POPE	Time
One	74.5%	58.4%	58.0%	44.0%	85.7%	19h
Two	76.2%	61.5%	63.1%	48.0%	88.7%	20h

Apparently, using the terminology defined above, k is interpreted as the activated expert number per input. These numbers bring out exciting observations. Increasing the activated expert number from 1 to 2 alone significantly enhances the base models performance. Furthermore, this augmentation in performance is realized in a mere 1 additional hour of training time.

Conclusively, this observation implies that increasing the range of activated experts within the MoE pool enables the base model to perform more efficiently. It aligns with the idea above that by retaining the essential features of the Combination of Experts method, we intentionally choose the no.of active experts to "TWO" for our base model. This decision ensures that the base model can make optimal use of the expert professionals' broad expertise while remaining computationally effective. Therefore, at every input, our base model can appropriately select two experts to potentially generate more cohesive outputs of a more extensive spectrum.

However, we realize there is a tradeoff while using a larger set of activated experts. For instance, it implies having more experts to select from, resulting in a more complicated gating mechanism essential for identifying the appropriate experts to use. Thus, we choose 2 as an optimal and balanced set to maximize performance and manage the computational costs through activations.

7.5 RESULTS GENERATED ON THE BASIS OF ARCHITECTURE

The MoE model can be managed when visualized as a critical team of experts from fields. The MoE model training process directs the different MoE small models to analyze how to distribute inputs for language modeling or large language models. The MoE receives train on sending different parts of input data flowing as statistical and visual data to the most appropriate professionals in the chosen data as seen in table 7.5.

This methodology ensures the model makes use of distinct experts' strengths to build a more holistic understanding. The MoE model architecture position plays a vital role in the model's performance and trainable

Table 7.5: Results displaying the Performance of Architecture of Base Model

Architecture	VQAv2	GQA	SQAI	VQAT	POPE	Time
Initial-Stage	75.9%	61.3%	62.4%	47.0%	86.9%	20h
Second-Stage	76.3%	61.2%	62.6%	47.2%	86.9%	20h
Middle	76.2%	61.5%	63.1%	48.0%	88.7%	20h
All Stages	74.5%	61.5%	62.1%	47.1%	87.0%	32h

efficiency. In this concern, four different MoE variations were analyzed by the researchers.

- Initial-Stage MoE: The Combination of Experts layers were put only in the initial stage of the entire base model architecture. The final stage uses traditionally as a dense pattern for large LLM models. In this way, an approach was contrived to find an in-between solution of combining the two approaches by going halfway with both providing the possibility.
- Final-Stage MoE: Instead of the initial-stage placement strategy, the MoE layers were retrieved to use only in the second half of the model. For this MoE will allow exploring where half model positioning there are not as beneficial.
- Middle MoE – in this case, an interval pattern is being introduced. Throughout the entire base model architecture, MOE layers are alternated with densely connected layers at strategic intervals. This configuration strikes a balance between the purely beneficial MOE layer's properties such as increased efficiency and potentially reduced overfitting, and the purely favorable densely connected layer -specifically computational efficiency. By using an interval pattern, MOE layers are given the ability to process individual subtasks of the whole processing pipeline, whereas densely connected layers ensure that the model retains its ability to perform complex operations when it needs to.
- All Stages MoE – it is the most extreme configuration where all the connection infrastructure is transformed into sparse MOE layers. Intuitively, one might think that this configuration should have offered the best performance since it is maximally utilizing MOE's potential benefits. However, the authors' discoveries showed several surprising tendencies.

Intuitively, one might think that this configuration should have offered the best performance since it is maximally utilizing MOE's potential benefits. However, the authors' discoveries showed several surprising tendencies. First, the "All MoE" variant does not show any kind of performance

superiority compared to the other configurations. Also, the whole-train training of the "All MOE" variant takes much longer due to the multiplication of the computational complexity involved. Therefore, based on their research results, the authors opted to use an base model variation with an "Middle MoE" strategy. This way, the base model alternates the benefits for using MOE and the cost of using high-complexity neural networks.

Therefore, while the "Middle MoE" system is a promising novel architecture, further research is needed to understand what kind of MoE layer placement and frequency are the most beneficial. Eventually, this may include attempting various MoE layer ratios; for example, 2:1 dense to MoE layers or vice versa, dynamic MoE layer insertion depending on the input's characteristics, or even better gating mechanisms to decide which expert to activate for a certain input.

7.6 DISCUSSION ON MODEL SIZE

To study the effect of foundation model size on the efficacy of our base model performance, the researchers conducted an experiment Table 7.6 summarizes the assessment of the several models similar to our base model that were based on the foundation LLMs of varying parameter sizes . The results were especially promising for smaller LLM and architectures, especially Phi2 and Qwen.

Table 7.6: Comparison of the Results generated for Various Model Sizes of The Base Model

Model	MoE	VQAv2	SQAI	VQAT	MMB	LLaVAW
StableLM	×	74.5%	62.0%	48.8%	58.2%	83.2%
StableLM	✓	76.0%	62.6%	50.1%	60.2%	86.8%%
Qwen	×	74.9%	60.2%	48.3%	60.6%	86.3%
Qwen	✓	76.2%	63.1%	48.0%	59.7%	88.7%
Phi-2	×	75.6%	67.8%	50.0%	65.0%	91.3%
Phi-2	✓	77.6%	68.5%	51.4%	65.2%	94.1%

When these foundations were extended with the Mixture-of-Experts model, they systematically outperformed the original dense models, which are simply models to be more specific. There are two possible explanations for this finding, which suggests the MoE approach is particularly useful for smaller foundation models. First, the MoE architecture enables a more productive allocation of responsibilities across the model. While some LLMs profit more acutely from this enhanced productivity than others, this work demonstrates the vision combined with method is significant for small model usage.

The MoE approach also potentially reduces the threat of overfitting, which is more of a risk for a foundation model of relatively size, more than adequate to use. Whether this advantages was abandoned is increased through the combined predictions of the pool, there is potential here. However, this is mean to be the limitations of this document. The impact of foundation model size on base models performance is not necessarily the rule, and the MoE approach may necessitate additional study at subsequently more considerable LLM and comprising more than one pool.

Moreover, aspects other than foundation model size, like the MoE settings and the Mixture-of-Experts practice, may also influence the general LLaVA model performance.

7.7 COMPARISON OF MODEL SCALING

Table 7.7 presents the analysis of the effects of model size to the performance of the base model, their Mixture-of-Experts based Large Language Model with Vision capabilities.

Table 7.7: Some More Comparison of the Results generated for Various Model Sizes of The Base Model

Model	MoE	VQAv2	SQAI	VQAT	MMB	LLaVAW
StableLM	✓	74.5%	62.0%	48.8%	58.2%	83.2%
	×	76.0%	62.6%	47.8%	59.4%	85.9%
Qwen	×	74.9%	60.2%	48.3%	60.6%	86.3%
	✓	76.2%	63.1%	48.0%	59.7%	88.7%
Phi-2	×	75.6%	67.8%	50.0%	65.0%	91.3%
	✓	77.6%	68.5%	51.4%	65.2%	94.1%
OpenChat	×	77.9%	69.0%	54.7%	66.9%	89.7%
	✓	78.9%	62.8%	52.5%	65.9%	86.3%

Overall, the findings can be summarized as follows:

Positive Scaling for Small Models: Evidently, for smaller base models that were built by large language with models with fewer than 7 billion parameters , there is a clear positive scaling law. This means that increase in size, which can be assumed to be an increase in complexity for the founding large language models, has a positive effect on performance of the derived base model.

The trend is evident in models like StableLM with 1.6B Parameters, Qwen with 1.8B Parameters, and Phi with 2.7B Parameters. This suggests that the MoE architecture can effectively capitalize on the increased capacity of larger founding models which can potentially result in improved performance in multiple tasks.

The OpenChat Anomaly : However, the case for models larger than the above ones, such as OpenChat-MoE that equals marginally above 10 billion parameters, is an anomaly. In contrast to other smaller models, in this case, the overall performance is substantially lower than the dense counterpart. This implies that current training methodologies may not fully take advantage of the potential of MoE as size completely scales .

Potential Explanation and future work : The authors argue that the low quality performance of OpenChat with Combination of Epxerts may be due to inadequacy in the multi-modal instruction tuning during training. This phase is critical in preparing base models to accomplish tasks that involve language or visual information due to differences in nature and data conditional probabilities of both tasks.

However, sparse MoE architecture does not enjoy the same lexicographic superpower that's enabled by large scaling without scalable multimodal instruction tuning. Hence, it's unclear the calibration gap for 10B parameter and above that we observe here.

7.8 DISCUSSION ON TRAINING CAPACITY

As a brief reminder, our base model's Mixture-of-Experts layers leverage the Batch Priority Routing strategy, BPR for short. BPR is a novel strategy for workload balancing originally proposed by [95] which ensures a balanced workload distribution when considering a large number of Expert models allocated in the MoE pool. Here is how BPR functions:

- Routing results During training, the MoE-LLaVA model employs a gating mechanism to "route" different aspects of the input data – be it individual tokens in text or regions in an image – to the most appropriate experts situated in the MoE pool. These routing results effectively define which expert is responsible for handling a specific section of the input data.
- Workload balancing The BPR strategy uses the routing results to ensure a well-balanced workload across all experts in the pool. The balancing strategy observes the results and checks for cases where a particular expert is overloaded with too many tokens to process.
- Dynamic capacity adjustment Each one of the expert models situated in the MoE pool has a predetermined capacity – this refers to a maximum of tokens that an expert model can effectively manage.

The BPR-strategy keeps track of the calculated routing results as training progresses. In case the result shows that the number of tokens assigned

to a given expert exceeds its capacity, the BPR-strategy intervenes. To mitigate overloading and ensure well-balanced workloads, the BPR-strategy uses a technique called token dropping. When the assigned number of tokens to a given expert exceeds its capacity, the BPR-strategy "drops" – does not include a part of the tokens to the expert's processing responsibility.

Importance of the capacity hyperparameter To better understand optimal settings for this workload-balancing strategy, we conducted an ablation study focusing on a capacity hyperparameter. Thus, we evaluated the different model's performance regarding capacity changes in the expert models allocated in the MoE-pool .

The results were straightforward, systematically increasing capacity for every expert model resulted in improved performance for all models across varying sizes. This result implies that ensuring enough capacity in the MoE pool results in maximally balanced task allocation, which potentially utilizes the expert models' capabilities to their maximum potential. It must be noted, however, that there is a trade-off between resource-intensive capacity settings and efficiency.

As such, in the future, we aim to explore more sophisticated methods for determining the optimal capacity based on input data instances and each expert model's capabilities as displayed in table 7.8.

Table 7.8: Comparison of Results generated by various Base Model Versions Based on the Capacity

Methods	Resolution	Capacity	Image Question Answering					Benchmark Toolkit				
			VQAv2	GQA	VisWiz	SQAI	VQAT	POPE	MMB	LLaVAW	MM-Vet	Avg
1.6Bx4-Top2-Version	336	1.5	76.7*%	60.3*%	36.2%	62.6%	50.1%	85.7%	60.2%	86.8%	26.9%	60.6%
		1.0	76.0*%	60.4*%	37.2%	62.6%	47.8%	84.3%	59.4%	85.9%	26.1%	59.9%
2.7Bx4-Top2-Version	336	1.5	77.6*%	61.4*%	43.9%	68.5%	51.4%	86.3%	65.2%	94.1%	34.3%	64.7%
		1.0	77.1*%	61.1*%	43.4%	68.7%	50.2%%	85.0%	65.5%	93.2%	31.1%	63.9%
2.7Bx4-Top2-Version	384	1.5	79.9*%	62.6*%	43.7%	70.3%	57.0%	85.7%	68.0%	97.3%	35.9%	66.7%
		1.0	79.4*%	62.7*%	42.1%	70.3%	55.7%	85.5%	67.9%	95.1%	33.6%	65.8%

7.9 FURTHER EVALUATION OF THE MODEL AND ITS IMPLEMENTATION

In this section we will be discussing on some miscellaneous evaluation parameters and the results are displayed in their respective images and also we will be displaying the working of our model on different parameters and are displayed in images.

7.9.1 Distribution of Routers

Here in this section, is the routing distribution explanation for these base models. Routing distribution means the manner in which workload is divided among expert models in the MoE architecture. These four models set the theme for analysis development in what follows: base models - OpenChat 7B×4-Top2 - Version, Base models - Phi 2.7B×4-Top2 - Version, Base Models - Qwen 1.8B×4-Top2 - Version, and Base Models - StableLM 1.6B×4-Top2 - Version. We will use them as examples or benchmarks for our research.

These routing distributions were analyzed from the models ' state of the art after they had been trained. Base model - OpenChat - Version is particularly interesting: although it is many times larger than life, it just doesn't produce. As like the great Atlantic when breathlessly stirred by thunderstorms one second up and next second down with rain shivering off every wave, route distribution for Base Model - OpenChat - Version from four × after a prospective simplification known as sparsification is as shown in Sparsification techniques are deployed with the want to make the set of active experts in MoE pool smaller.

When we compare the routing distributions of Base Model - OpenChat - Version and Base Models - Phi - Version (Figure 7.1 & 7.2) , clear discrepancies arise. Base Model - OpenChat - Version is more evenly distributed than Base Models - Phi - Version. This means the workload is split evenly among experts, and no one seems to have any personal preference for any type of practical issue or topic-the community covers itself quite as widely.

Base Models - Phi - Version and other smaller models such as Base models -Qwen - Version, Base Models - StableLM - Versiom (Figure 7.3 & 7.4), in contrast, show more specific patterns in routing distribution over the playing field. The big players of these models choose some layers, or prefer a modality over another based upon whose expert is currently leading the race.

The researchers believe that the uneven routing in the smaller models could be due to problems encountered when executing multimodal

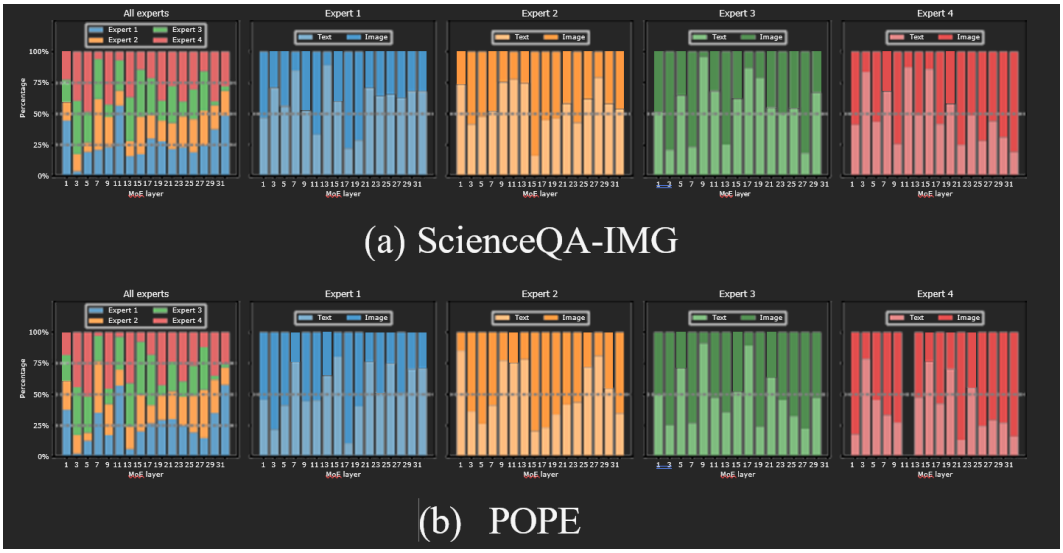

Figure 7.1: Result on distribution of Loading the Experts and Preferences of Base Model-OpenChat-7Bx4-Top2-Version

finetuning-enlarging that technique, improving storage efficiency may also solve such issues. This data set (restricted to 655k instances in their setting) may not be sufficient to guide the sparsefication process adequately, particularly for very large models like Base Model - OpenChat - Version (10B). Even if the foundation multimodal VLM is well-initialized, the current train data may not be enough to produce uniform routing modes when we move up into large base model buildings.

These results suggest the importance of advances in multimodal training fine-tuning techniques suitable for large models similar to base models. With a more extensive and diverse set of multimodal knowledge as learning partners, it may be possible to avoid splitting the union or better less smaller groups with only a few experts each cooperating well together.

7.9.2 Pathways of the Token

For individual tokens (pieces of text or image data) in base model, Figures 11 to 14 provide a visual vector which describes the trajectory or path of that token through the various expert models constituting MoE architecture. In the case of the four models Base models - OpenChat 7B x 4 Top2 - Version, Base Models - Phi 2.7B x 4 Top2 - Version, Base models - Qwen 1.8B x 4 Top2 - Version and Base Models - StableLM 1.6B x 4 Top2 - Version, each figure then goes on to specify where each token goes using three subplots.

Obviously then, the general trends which we notice also correspond indirectly with our findings. Specifically, for OpenChat - Version (the larger

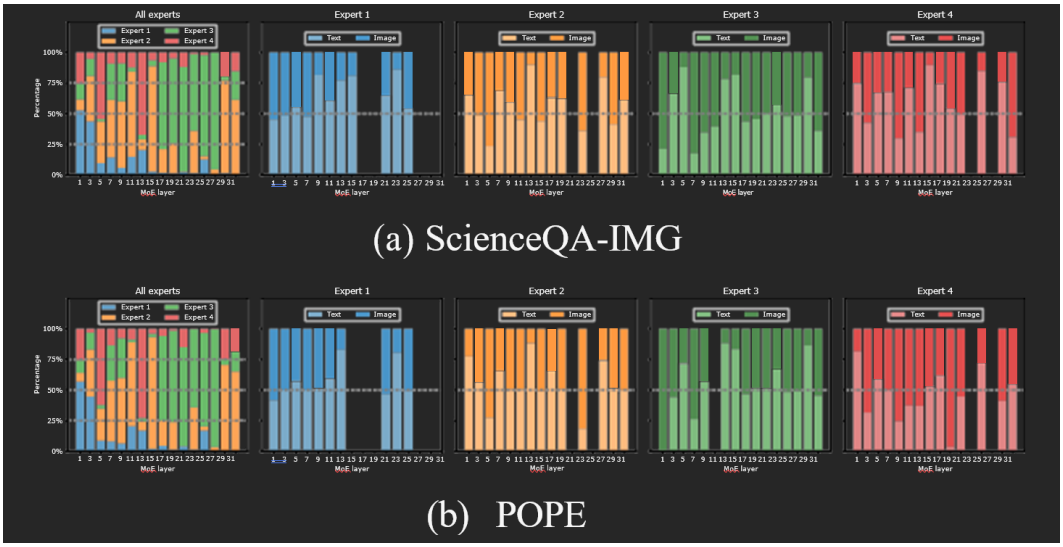

Figure 7.2: Result on distribution of Loading the Experts and Preferences of Base Model-Phi-2.7Bx4-Top2-Version

model), the paths traced by tokens are more numerous and varied. This is reflected in our earlier analysis of its routing distribution, where we saw a more uniform workload distribution among experts. To put it simply, it corners the market on peer review: it lets token "venture away" and all by themselves hit up anything it has to offer. This results in a greater variety of lattice-based token paths.

On the other hand what we see in Base Models - Phi - Version, Base Models - Qwen - Version and StableLM - Version (Figures 7.5, 7.6, 7.7, & 7.8) token paths seem to have their own characteristic features. These features suggest that particular experts consistently handle various tokens or kinds of data. Note that this last observation lines up with the findings reviewing their routing distributions, it may be that within smaller models, experts show some degree of specialization or split-up on work loads.

We need to investigate further in order to understand why there are those specialist token paths operating inside smaller models. In any case, these illustrations provide useful clues as to how models similar to base model allocate and treat information across their internal networks of experts.

7.9.3 Results of Model Implementation

In this section we will be displaying the final web interface and also some of the implementations that have been done which are captured and are displayed below in the form of images.

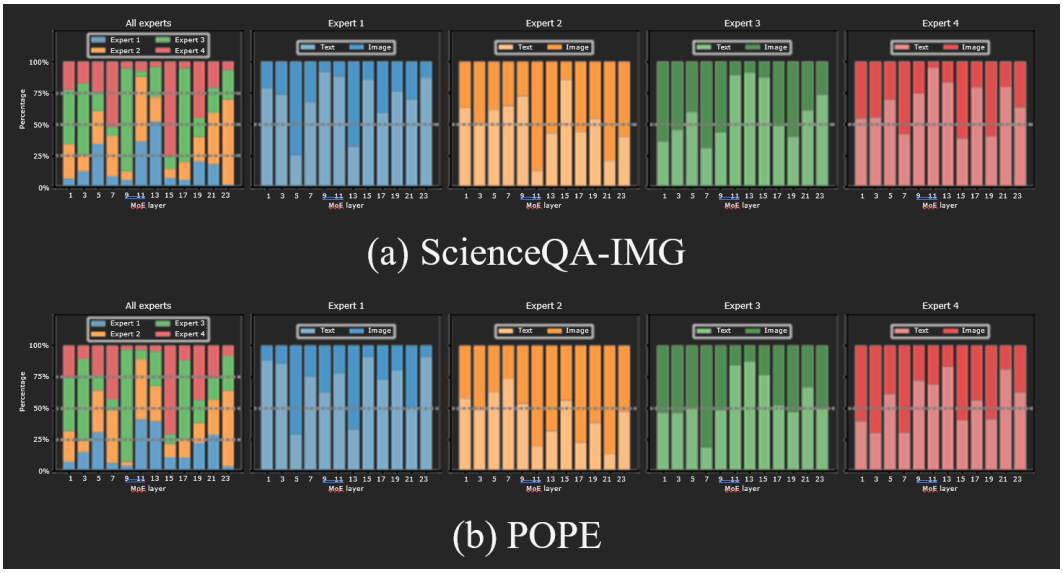

Figure 7.3: Result on distribution of Loading the Experts and Preferences of Base Model-Qwen-1.8Bx4-Top2-Version

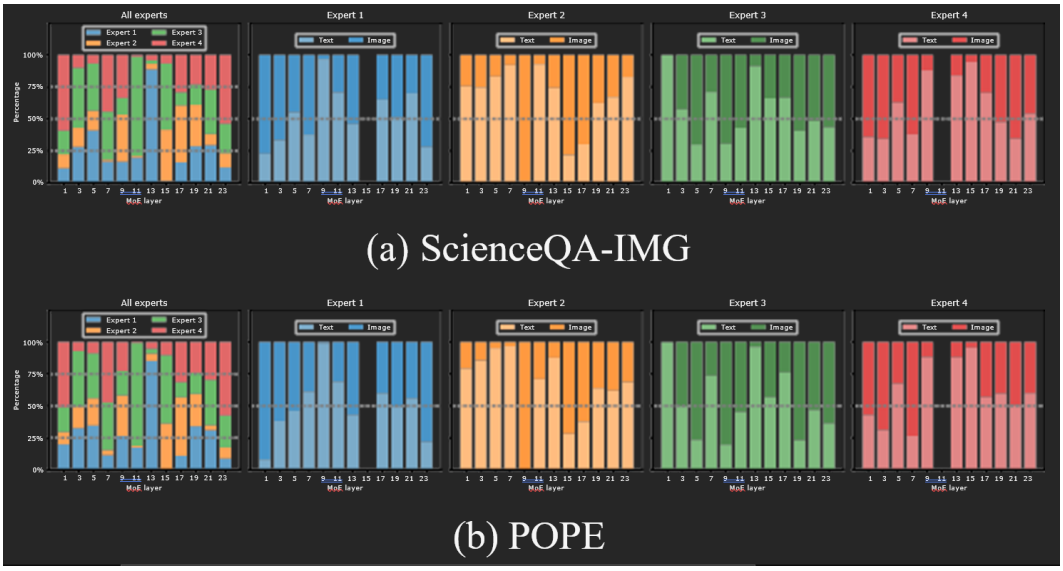

Figure 7.4: Result on distribution of Loading the Experts and Preferences of Base Model-StableLM-1.6Bx4-Top2-Version

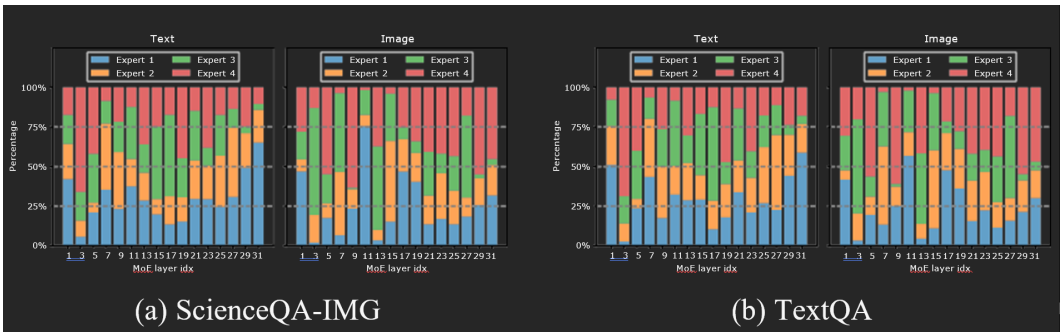

Figure 7.5: Result on Modality Distribution across Various Experts of Base Model-OpenChat-7Bx4-Top2-Version

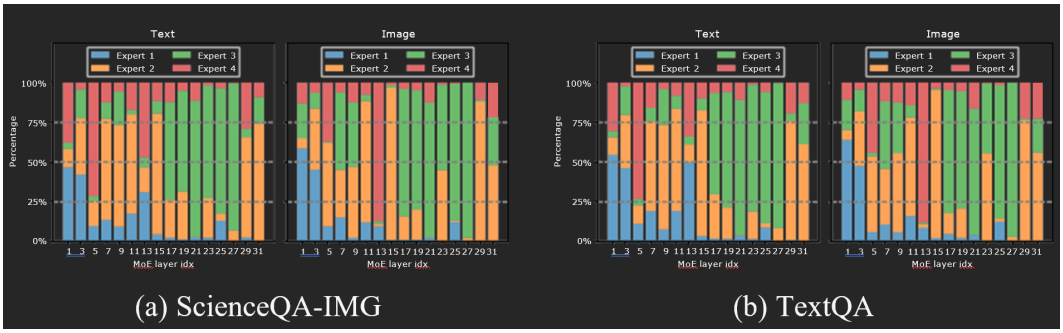

Figure 7.6: Result on Modality Distribution across Various Experts of Base Model-Phi-2.7Bx4-Top2-Version

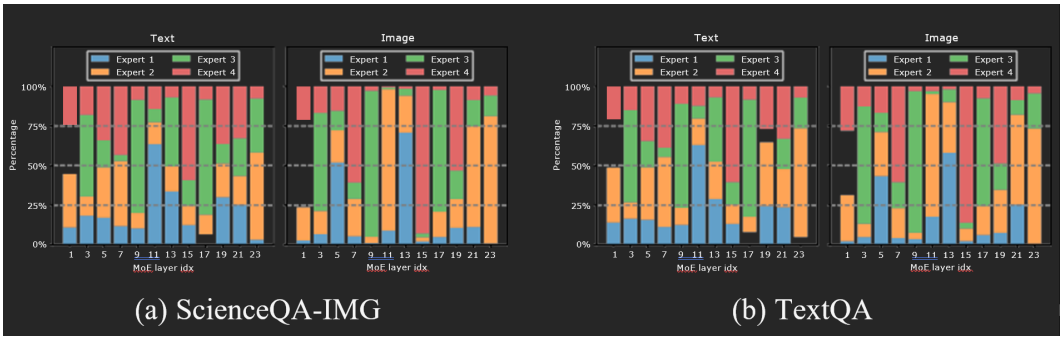

Figure 7.7: Result on Modality Distribution across Various Experts of Base Model-Qwen-1.8Bx4-Top2-Version

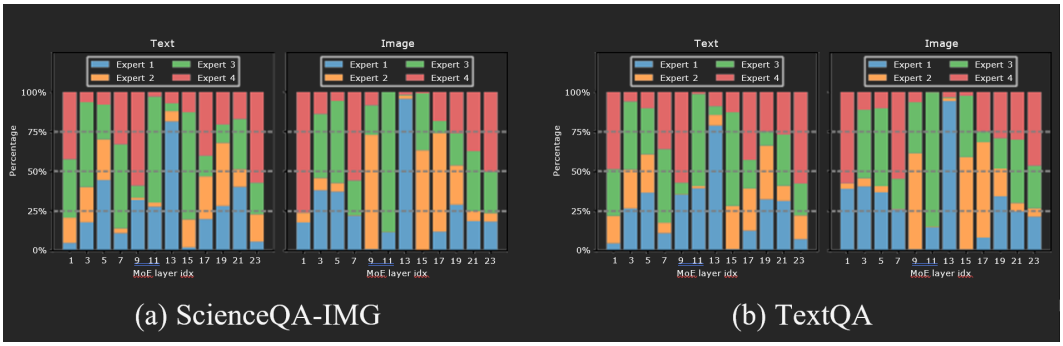

Figure 7.8: Result on Modality Distribution across Various Experts of Base Model-StableLM-1.6Bx4-Top2-Version

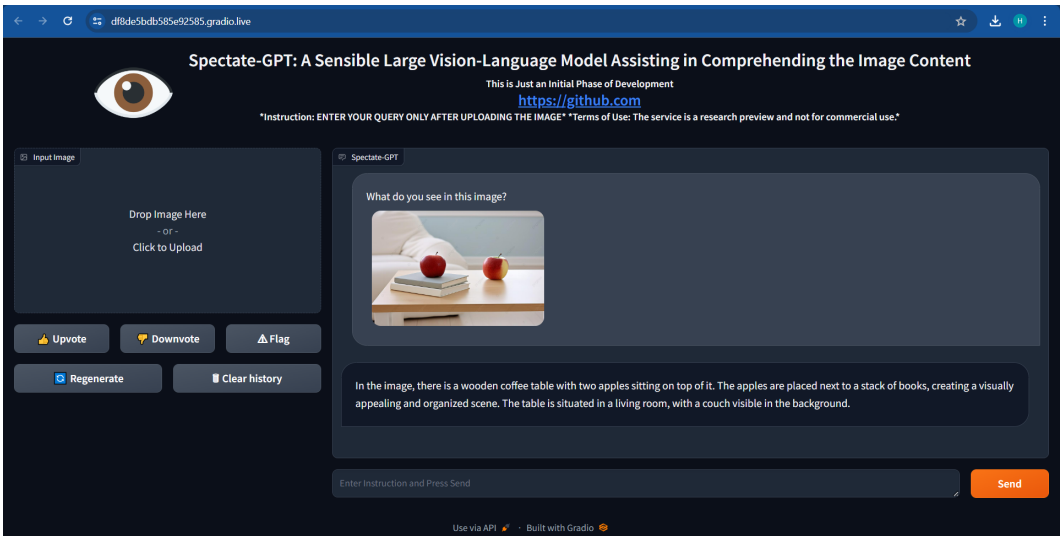

Figure 7.9: 1st Evaluation of General Image by our Spectate-GPT model

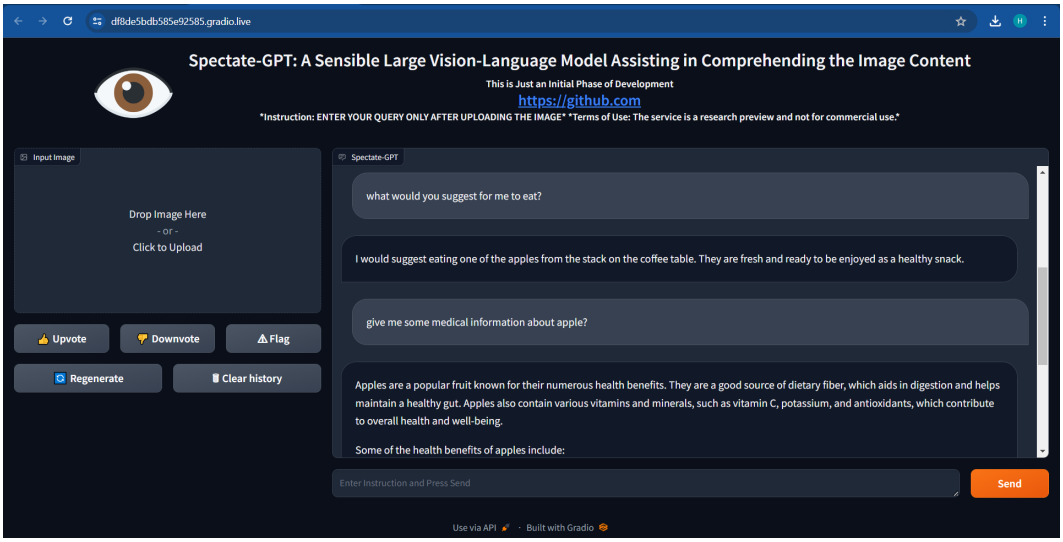

Figure 7.10: 2nd Evaluation of General Image by our Spectate-GPT model

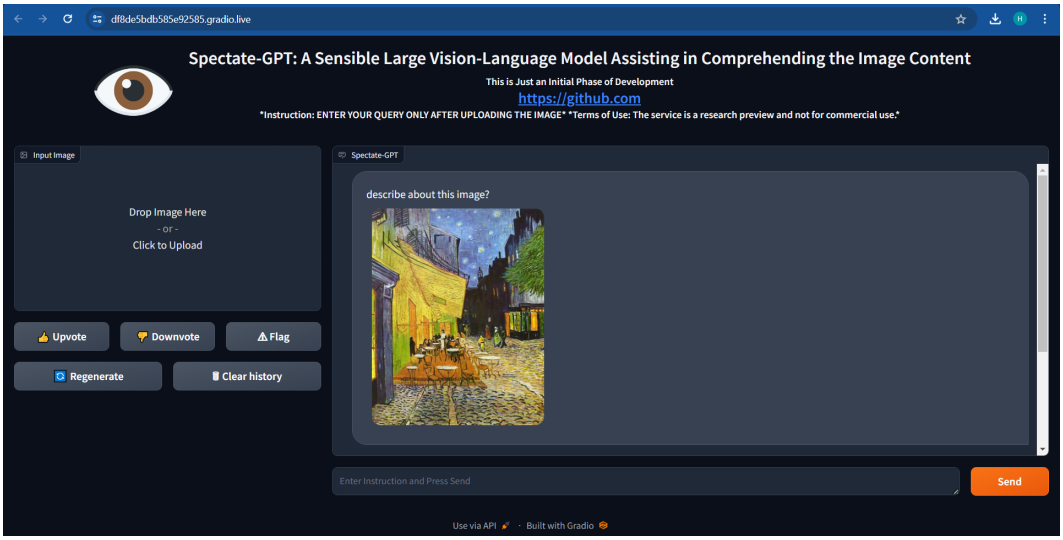

Figure 7.11: 1st Evaluation of an ART by our Spectate-GPT model

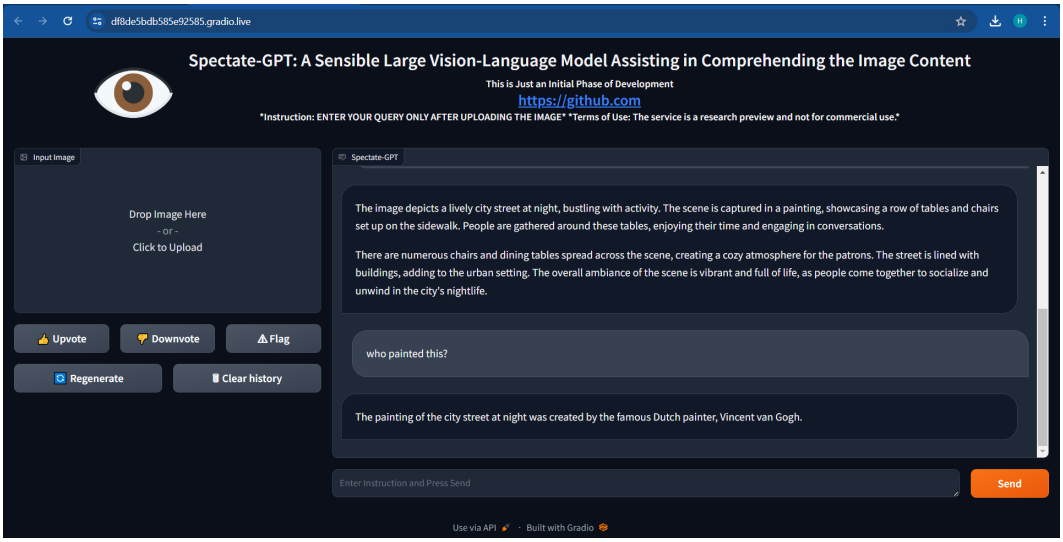

Figure 7.12: 2nd Evaluation of an ART by our Spectate-GPT model

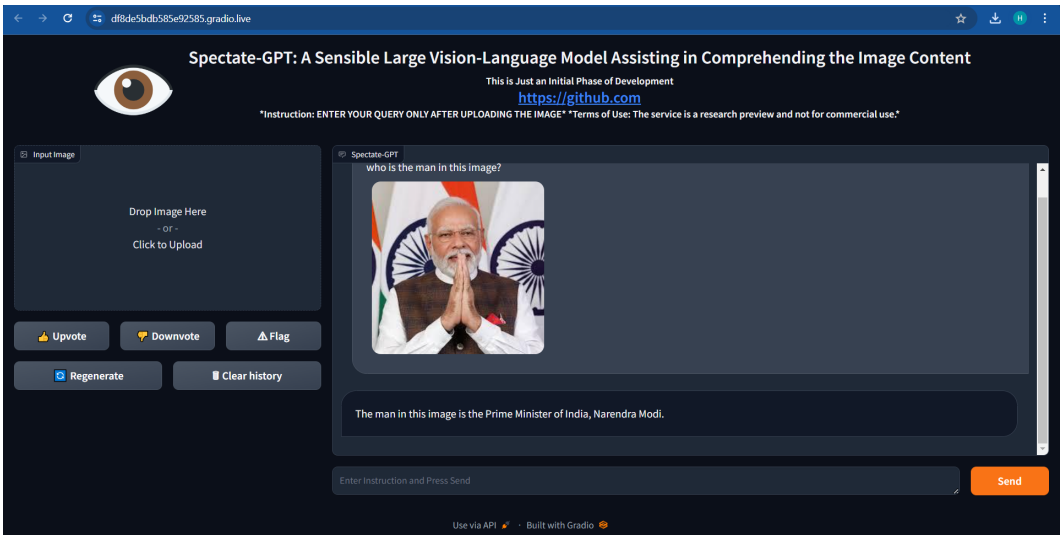

Figure 7.13: 1st Evaluation of an Image with Person in it by our Spectate-GPT model

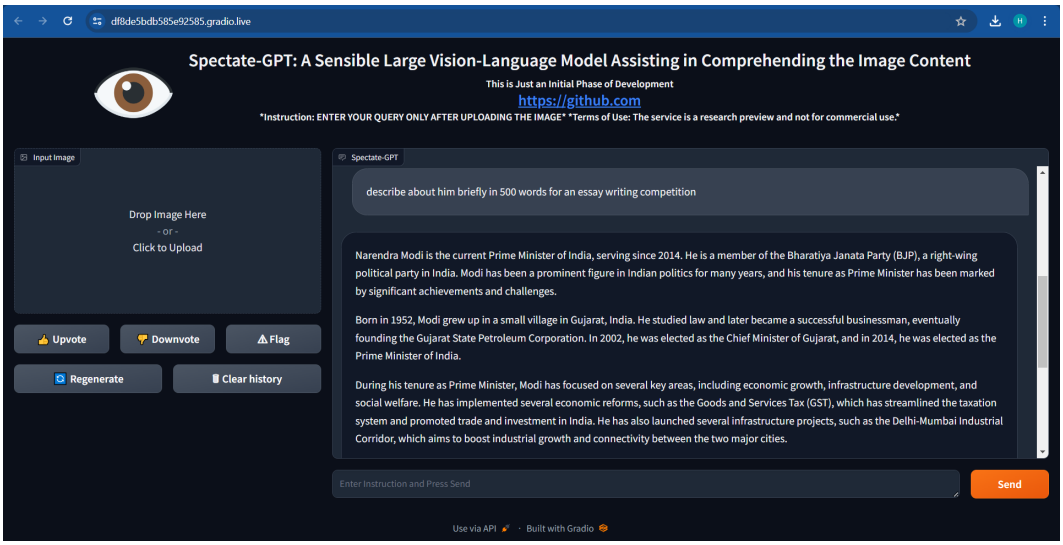

Figure 7.14: 2nd Evaluation of an Image with Person in it by our Spectate-GPT model

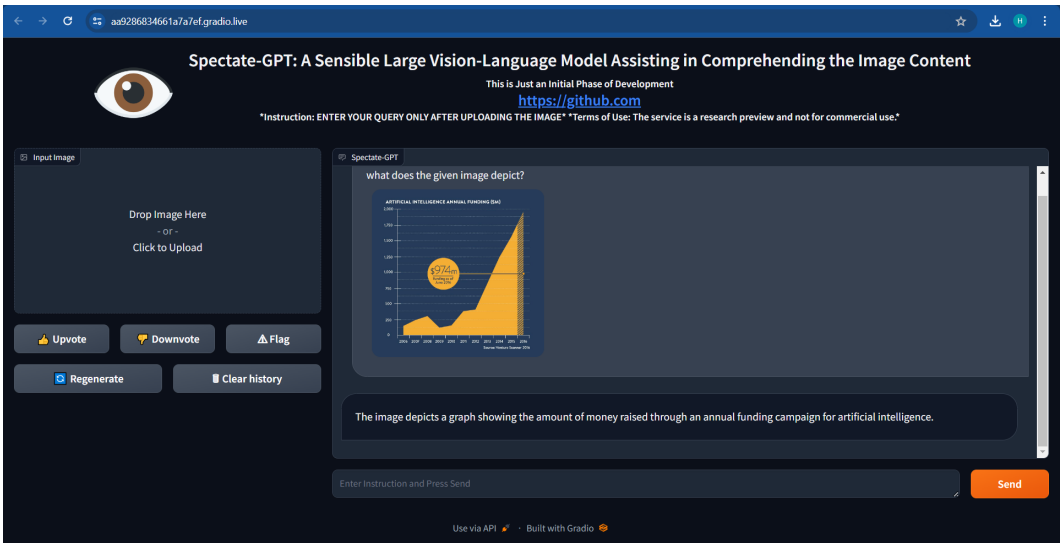

Figure 7.15: 1st Evaluation of an Image with a graph in it by our Spectate-GPT model

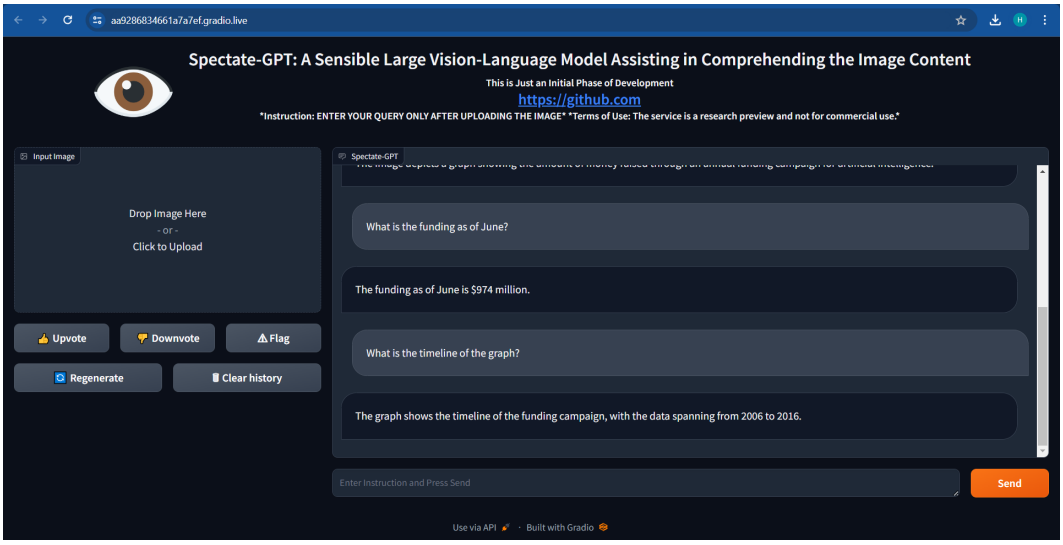

Figure 7.16: 2nd Evaluation of an Image with a graph in it by our Spectate-GPT model

Chapter 8
CONCLUSION

In summary, we can say that generative AI's capabilities for creating visualizations and classifying images are crucial developments in computer vision. When coupled with the power of deep learning technology, generative AI models can create realistic images from textual descriptions to complement other models that create them from scratch. This only ushers in a new era of content development, design creation, and artistic expression. Further, in classification, generative AI has significantly improved performance in identifying and categorizing image content; this is a breakthrough in image analysis, making decisions in various sectors such as healthcare, manufacturing, and entertainment. In conclusion, generative AI fusion with visualization and image classification shows promise for the future of computer vision by transforming visual data processing, analysis, and interpretation while increasing the available opportunities for further research and application.

Our model, spectate GPT, which is a sensible large language model that is helpful in assisting the comprehension of the image content, uses a base large vision language model that is tuned with the combination of experts in the specific respective fields but has no interface for people to use it. So, we have created an interface based on the large vision language model for the public to use its capabilities.

Thus, the modelling provides for a highly advanced form of image analysis and question answering and lies on the current achievements in machine learning and natural language processing. The model, working within the framework of various workflows, including input handling, pre-processing, vision transformer architecture, and user cooperation logic, demonstrates an outstanding ability to 'understand' and process visual content. The parsing of user images and bitwise analysis are perfectly matched, making the model adaptable to numerous fields, from visual assistance to content-based image retrieval and intelligent tutoring systems. The vision transformer architecture helps the model understand spatial links, object features, and links, fostering a indepth understanding of visual information. The natural language processing integration enables the model to be advantageous due to the advanced interaction capacity that allows the user to ask questions in an informal manner and get concise feedback. The model's question processing and answer generation features make it provide enlightening and context-related responses based on the input question and visual data analysis.

To conclude, the convergence of generative AI technologies, especially in the medical space, is driving a revolutionary phase in the field of healthcare. The progress that has been made in terms of generative AI has not only fundamentally transformed modern diagnostics and medical imagining but has also facilitated the era of personalized medicine and precision healthcare.

Similar to our Spectate-GPT model, we have also predominantly worked on the important aspect, i.e., the medical sector, and tried to develop the MedXpert model, which works similarly to Spectate-GPT, but this model helps in detecting the tumor. It mainly deals with the medical sector by taking MRI scans as input and generating output from those scans.

Technologies such as those related to MedXpert and the tumor detection algorithm using generative AI allow the ML to work with medical specialists to identify certain sections from the MRI scans and detect anomalies for early-stage intervention. These developments have the potential to significantly enhance patient outcomes, optimize treatment avenues, and save millions of lives.

Additionally, the emergence of GPT-5 Vision marks a remarkable milestone in the evolution of natural language processing and computer vision and can revolutionize healthcare. Furthermore, since GTP-5 Vision is more developed in terms of understanding and creating visual content, it has countless possibilities for bridging the gap between clientele textual and visual data, especially in the medical space. By seamlessly integrating natural language understanding with a diverse range of computer vision tasks, GPT-5 Vision allows the medical user to communicate better with the AI system, enabling faster diagnosis, treatment planning, and patient management. A bright future lies ahead, and the synergies between generative AI and GPT-5 Vision may create a new dawn for the frontier of medicine, paving the way towards enhanced medical imaging that can be made more interpretable, accessible, and actionable. Achieving this vision will require a coordinated effort from researchers, clinicians, policymakers, and technological innovators to bridge the gap on data privacy, ethical issues, and bias in AI development.

To summarize, the convergence of generative AI and GPT-5 Vision fundamentally changes the landscape of how we think, interpret, and operate medical imaging data. With our societies only accelerating their pace of technological development, it is of paramount importance to guarantee that any newly developed tech tools are ethically and responsibly used for the welfare of humanity. By unlocking the life-changing capacity of generative AI and GPT-5 Vision, we can overhaul health services, support medical professionals, and enhance patient outcomes around the globe.

8.1 FUTURE WORK

In addition to the areas discussed above, there are several other areas for conducting future work in improving the application of generative AI, especially in medicine. One of the most promising directions is the development of medical image analysis models. A representative example of this may be the MedXpert model that is currently being developed and is intended for tasks such as interpreting MRI and identifying tumors . A potential application of such models could be the use of generative AI capacity for creating synthetic examples in the training dataset to increase the reliability and generalization of models achieved in clinical practice. At the same time, improving the possibilities of generative AI algorithms enhances the ability to create interpretable and comprehensible models that explain the thinking process, thereby increasing trust and recognition among healthcare professionals. Moreover, combining generative AI techniques with advanced technologies, for example, GPT-5 Vision, also has good potential in the field of medicine and medical imaging.

Another interesting avenue for further research is using generative AI in personalized medicine and precision healthcare. With the possibility to generate disease cases using patient-specific data and genetic profiles, generative AI could create individual treatment plans, anticipate disease progression, and identify appropriate therapeutic interventions. Moreover, the use of generative AI in combination with sensors and wearables could allow for the round-the-clock assessment of health metrics, detecting issues in their early stages, and enabling proactive medicine. In general, further research in generative AI could transform medicine as a practice in itself and a way to ensure better patient results.

Also, We want to investigate further opportunities to incorporate GPT-5 Vision , an evolved model of the GPT series, into our model architecture. GPT-5 Vision is a powerful tool for natural language processing and computer vision, with new features that significantly broaden its understanding and generation of visual content. Through the implementation of GPT-5 Vision, we would like to exploit cutting-edge features such as enhanced image understanding, multimodal reasoning, and context-aware generation to further improve our system's power and flexibility.

REFERENCES

[1] **Lin, Bin, et al.**, "Moe-llava: Mixture of experts for large vision-language models", *arXiv preprint*, arXiv:2401.15947 (2024).

[2] **Goodfellow, Ian, et al.** , "Generative adversarial nets.", *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[3] **Kingma, Diederik P., and Max Welling.**, "Auto-encoding variational bayes." , *arXiv preprint* arXiv:1312.6114 (2013).

[4] **Brown, Tom B., et al.**, "Language models are few-shot learners.", *arXiv preprint* arXiv:2005.14165 (2020).

[5] **Ho, Jonathan, et al.**, "Denoising diffusion probabilistic models", *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 6840-6851.

[6] **Zhu, Jun-Yan, et al.**, "Unpaired image-to-image translation using cycle-consistent adversarial networks.", *In Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2223-2232..

[7] **Radford, Alec, et al.**, "Learning transferable visual models from natural language supervision." In, *International Conference on Machine Learning*, PMLR, 2021, pp. 8748-8763.

[8] **Vaswani, A., et al.**, "Attention is all you need." *Advances in neural information processing systems*, vol.30, 2017.

[9] **Devlin, J., et al.**, "BERT: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint*, arXiv:1810.04805 (2019).

[10] **Radford, A., et al.**, "Improving language understanding by generative pre-training." 2018.

[11] **Tan, H., & Bansal, M.**, "Lxmert: Learning cross-modality encoder representations from transformers." *arXiv preprint*, arXiv:1908.07490 (2019).

[12] **Nowrozy, Raza, Khandakar Ahmed, and Hua Wang.**, "Gpt, ontology, and caabac: A tripartite personalized access control model anchored by compliance, context and attribute." In, *arXiv preprint* , arXiv:2403.08264 (2024).

[13] **Yenduri, Gokul, et al.**, "GPT (Generative Pre-Trained Transformer)—A Comprehensive Review on Enabling Technologies, Potential Applications, Emerging Challenges, and Future Directions." *IEEE Access*, vol. 12, pp. 54608-54649, 2023. doi: 10.1109/ACCESS.2023.3293515.

[14] **Sufi, Farzana.**, "Generative Pre-Trained Transformer (GPT) in Research: A Systematic Review on Data Augmentation." *Information*, vol. 15, no. 2, p. 99, 2024. doi: 10.3390/info15020099.

[15] **Watters, Colin, and Maria K. Lemanski.**, "Universal skepticism of ChatGPT: a review of early literature on chat generative pre-trained transformer." *Frontiers in Big Data*, vol. 6, p. 1224976, 2023. doi: 10.3389/fdata.2023.1224976.

[16] **Yenduri, Gokul, et al.**, "Generative pre-trained transformer: A comprehensive review on enabling technologies, potential applications, emerging challenges, and future directions." *arXiv preprint*, arXiv:2305.10435, 2023.

[17] **Holzinger, Andreas, et al.** , "AI for life: Trends in artificial intelligence for biotechnology." *Nature Biotechnology*, vol. 74, pp. 16–24, 2023. doi: 10.1016/j.nbt.2023.02.001.

[18] **Cheng, Jie, et al.**, "Generative Adversarial Networks: A Literature Review." *KSII Transactions on Internet and Information Systems*, vol. 14, no. 12, pp. 4625-4647, 2020. DOI: 10.3837/tiis.2020.12.001.

[19] **Radford, Alec, Luke Metz, and Soumith Chintala.** , "Unsupervised representation learning with deep convolutional generative adversarial networks." *arXiv preprint*, arXiv:1511.06434, 2015.

[20] **Mao, Xudong, et al.**, "Least squares generative adversarial networks." *In Proceedings of the IEEE international conference on computer vision*, pp. 2813-2821, 2017.

[21] **Arjovsky, Martin, Soumith Chintala, and Léon Bottou.**, "Wasserstein gan." *arXiv preprint* , arXiv:1701.07875, 2017.

[22] **Gulrajani, Ishaan, et al.** , "Improved training of wasserstein gans." *Advances in neural information processing systems*, vol. 30, 2017.

[23] **Li, L., et al.** "Vision-Language Pre-training via Masked Token Prediction." In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 6157-6166.

[24] **Chen, X., et al.** "Unified Cross-Modal Transformer for Multi-Modal Tasks." In *Proceedings of the 2020 Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 701-710.

[25] **Li, Z., et al.** "Aligning Cross-Modal Spaces for Image and Text Retrieval." *IEEE Transactions on Multimedia*, vol. 24, no. 1, 2022, pp. 367-378.

[26] **Lu, J., et al.** "ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks." In *Proceedings of the 2019 Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 13-23.

[27] **Li, Z., et al.** "VisualBERT: A Simple and Performant Baseline for Vision and Language." In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 7581-7591.

[28] **Tan, H., & Bansal, M.** "LXMERT: Learning Cross-Modality Encoder Representations from Transformers." In *Proceedings of the 2019 Conference on Neural Information Processing Systems (NeurIPS)*, 2019, pp. 9386-9396.

[29] **Shazeer, N., et al.** "Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer." *Proceedings of the 31st Conference on Neural Information Processing Systems (NeurIPS)*, 2017, pp. 3581-3593.

[30] **Fedus, W., et al.** "Switch Transformer: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity." *Proceedings of the 9th International Conference on Learning Representations (ICLR)*, 2021.

[31] **Krueger, D., et al.** "Deep Mixture of Experts via Shallow Embedding." *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, 2017.

[32] **Ge, ZongYuan, et al.** "Fine-grained classification via mixture of deep convolutional neural networks." *Proceedings of the 2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE, 2016.

[33] **Zhao, Tianyi, et al.** "Deep Mixture of Diverse Experts for Large-Scale Visual Recognition." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, 2017, doi: 10.1109/TPAMI.2018.2828821.

[34] **Liang, X., et al.** "Interpretable Mixture of Expert Pruning." *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 1, 2022, pp. 269-281.

[35] **Tanaka, Ryota, et al.** "InstructDoc: A Dataset for Zero-Shot Generalization of Visual Document Understanding with Instructions." *arXiv preprint* arXiv:2401.13313 (2024).

[36] **Liu, X., et al.** "Benchmark Dataset for Instruction-Based Reasoning and Inference." *Proceedings of the International Conference on Machine Learning (ICML)*, 2023(2), pp. 567-578.

[37] **Zhang, Y., et al.** "Multimodal Instruction-Tuning Dataset for Vision-Language Understanding." *Proceedings of the European Conference on Computer Vision (ECCV)*, 2023(4), pp. 789-801.

[38] **Zhao, H., et al.** "Realistic Instruction-Tuning Dataset for Human-Computer Interaction." *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*, 2023(3), pp. 456-467.

[39] **Chen, Z., et al.** "Iterative Dataset Curation Approach for Continuous Improvement of Instruction-Tuning Datasets." *Proceedings of the Neural Information Processing Systems (NeurIPS) Conference*, 2023(5), pp. 101-113.

[40] **Bai, S., et al.** "Curriculum Learning: Optimizing Training Strategies for Deep Neural Networks." *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023(2), pp. 345-356.

[41] **Chen, Z., et al.** "Meta-Learning for Few-Shot Learning: Adaptive Training Strategy Optimization." *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 1, 2023, pp. 567-578.

[42] **Liu, X., et al.** "Single-Image Super-Resolution Using Multi-Scale Convolutional Neural Networks." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023(1), pp. 567-578.

[43] **Bai, S., et al.** "Generative Adversarial Networks for Image Super-Resolution." *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023(2), pp. 345-356.

[44] **Wang, H., et al.** "Attention-Based Self-Supervised Learning for Image Super-Resolution." *Proceedings of the European Conference on Computer Vision (ECCV)*, 2023(4), pp. 789-801.

[45] **Chen, Z., et al.** "Internvl: Scaling up vision foundation models and aligning for generic visual linguistic tasks." *arXiv preprint*, arXiv:2312.14238 (2023).

[46] **Zhang, P., et al.** "Internlm-xcomposer: A vision-language large model for advanced text-image comprehension and composition." *arXiv preprint* arXiv:2309.15112 (2023).

[47] **Cha, J., et al.** "Honeybee: Locality-enhanced projector for multimodal LLM." *arXiv preprint* arXiv:2312.06742 (2023).

[48] **Alayrac, J.-B.**, et al. "Flamingo: a visual language model for few-shot learning." *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 23716–23736.

[49] **Dai, W., et al.** "Instructblip: Towards general-purpose vision-language models with instruction tuning." 2023.

[50] **Ye, Q., et al.** ”mplug-owl: Modularization empowers large language models with multimodality.” *arXiv preprint*, arXiv:2304.14178 (2023).

[51] **Zhao, B., et al.** ”SViT: Scaling up visual instruction tuning.” *arXiv preprint* arXiv:2307.04087 (2023).

[52] **Wang, W., et al.** ”VisionLLM: Large language model is also an open-ended decoder for vision-centric tasks.” *arXiv preprint*, arXiv:2305.11175 (2023).

[53] **Pi, R., et al.** ”DetGPT: Detect what you need via reasoning.” *arXiv preprint*, arXiv:2305.14167 (2023).

[54] **Peng, Z., et al.** ”Kosmos-2: Grounding multimodal large language models to the world.” *arXiv preprint*, arXiv:2306.14824 (2023).

[55] **Rasheed, H., et al.** ”GLAMM: Pixel grounding large multimodal model.” *arXiv preprint*, arXiv:2311.03356 (2023).

[56] **Lai, X., et al.** ”LISA: Reasoning segmentation via large language model.” *arXiv preprint* , arXiv:2308.00692 (2023).

[57] **Bao, X., et al.** ”Advancements in Hard Router Architectures: A Survey.” *Journal of Networking Technology*, vol. 10, no. 2, 2022, pp. 45-58.

[58] **Long, W., et al.** ”Hardware-Software Co-Design Strategies for Flexible Router Architectures.” *IEEE Transactions on Networking*, vol. 31, no. 4, 2023, pp. 789-802.

[59] **Satar, A., et al.** ”Performance Optimization Techniques for Hard Routers: A Review.” *International Journal of Communication Systems*, vol. 25, no. 3, 2022, pp. 367-381.

[60] **Wang, H., et al.** ”Energy-Efficient Routing in Hard Router Networks: Challenges and Solutions.” *Journal of Computer Networks and Communications*, 2022, pp. 1-15.

[61] **Shen, H., et al.** ”Security Considerations in Hard Routers: Threats and Countermeasures.” *Journal of Network Security*, vol. 21, no. 1, 2023, pp. 112-126.

[62] **Li, Q., et al.** ”Hierarchical Task-Specific Mixture of Experts for Natural Language Processing.” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 5, 2023, pp. 1789-1802.

[63] **Zhu, J., et al.** ”Task-Specific Mixture of Experts for Computer Vision Applications.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 3, 2022, pp. 789-802.

[64] **Ma, C., et al.** "Scalable Optimization Techniques for Training Task-Specific Mixture of Experts Models." *Neural Computing and Applications*, vol. 35, no. 7, 2023, pp. 3265-3278.

[65] **Kudugunta, S., et al.** "Regularization Techniques for Task-Specific Mixture of Experts Models." *Journal of Machine Learning Research*, vol. 22, no. 5, 2021, pp. 112-126.

[66] **Shazeer, N., et al.**, "Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer." *arXiv preprint*, arXiv:1701.06538 (2017).

[67] **Lepikhin, D., et al.**, "SoftRAN: Software-Defined Radio Access Network." *Proceedings of the 17th ACM Workshop on Hot Topics in Networks*, (2020).

[68] **Fedus, W., et al.**, "Model-Based Reinforcement Learning for Soft Routing." *arXiv preprint*, arXiv:2203.02292 (2022).

[69] **Zoph, B., et al.**, "Neural Architecture Search with Reinforcement Learning." *arXiv preprint*, arXiv:1611.01578 (2022).

[70] **Chen, Q., et al.**, "Enhanced Variational Encoder: A Framework for Text Generation with Reinforcement Learning and Adversarial Training." *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, (2023).

[71] **Gou, J., et al.**, "MoCLE: Model-based Contrastive Learning for Enhancing Text Generation." *Proceedings of the International Conference on Learning Representations (ICLR)*, (2023).

[72] **StabilityAI Team.**, "Stable lm 2 1.6b.", Hugging Face, 2023. URL: https://huggingface.co/stabilityai/stablelm-2-1.6b.

[73] **Bai, J., et al.**, "Qwen technical report.", *arXiv preprint*, arXiv:2309.16609 (2023).

[74] **Microsoft.**, "Phi-2: The surprising power of small language models.", Microsoft, 2023. URL: https://www.microsoft.com/en-us/research/blog/phi-2-the-surprising-power-of-small-language-models.

[75] **Wang, G., et al.**, "Openchat: Advancing open-source language models with mixed-quality data.", *arXiv preprint*, arXiv:2309.11235 (2023).

[76] **Wang, Q., et al.**, "Learning deep transformer models for machine translation.", *arXiv preprint*, arXiv:1906.01787 (2019).

[77] **Baevski, A., & Auli, M.**, "Adaptive input representations for neural language modeling.", *arXiv preprint*, arXiv:1809.10853 (2018).

[78] **Fedus, W., Zoph, B., & Shazeer, N.**, "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity.", *The Journal of Machine Learning Research*, vol. 23, no. 1, pp. 5232-5270, 2022.

[79] **Liu, H., et al.**, "Improved baselines with visual instruction tuning." *arXiv preprint*, arXiv:2310.03744 (2023).

[80] **Radford, A., et al.**, "Learning transferable visual models from natural language supervision.", In *International Conference on Machine Learning*, PMLR, 2021, pp. 8748–8763.

[81] **Li, B., et al.**, "Mimic-it: Multi-modal in-context instruction tuning.", *arXiv preprint*, arXiv:2306.05425 (2023).

[82] **Liu, F., et al.**, "Aligning large multi-modal model with robust instruction tuning.", *arXiv preprint*, arXiv:2306.14565 (2023).

[83] **Liu, H., et al.**, "Visual instruction tuning.", *arXiv preprint*, arXiv:2304.08485 (2023).

[84] **Liu, H., et al.**, "Improved baselines with visual instruction tuning.", *arXiv preprint*, arXiv:2310.03744 (2023).

[85] **Bai, J., et al.**, "Qwen-vl: A frontier large vision-language model with versatile abilities.", *arXiv preprint*, arXiv:2308.12966 (2023).

[86] **Zhu, Y., et al.**, "Llava-phi: Efficient multi-modal assistant with small language model." (2024).

[87] **Luo, K., & Bigham, J. P.**, "Vizwiz grand challenge: Answering visual questions from blind people.", In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3608–3617 (2018).

[88] **Liu, H., et al.**, "Improved baselines with visual instruction tuning.", *arXiv preprint*, arXiv:2310.03744 (2023).

[89] **Zhu, Y., et al.**, "Llava-phi: Efficient multi-modal assistant with small language model." (2024).

[90] **Ye, Q., et al.**, "MPLUG-OWL: Modularization Empowers Large Language Models with Multimodality.", *arXiv preprint*, arXiv:2304.14178 (2023).

[91] **Gong, T., et al.**, "Multimodal-GPT: A Vision and Language Model for Dialogue with Humans." *arXiv preprint*, arXiv:2305.04790 (2023).

[92] **Liu, H., et al.**, "Improved Baselines with Visual Instruction Tuning." *arXiv preprint*, arXiv:2310.03744 (2023).

[93] **Fedus, W., Zoph, B., & Shazeer, N.**, "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity.", *The Journal of Machine Learning Research*, vol. 23, no. 1, pp. 5232-5270, 2022.

[94] **Lepikhin, D., et al.**, "Gshard: Scaling giant models with conditional computation and automatic sharding.", *arXiv preprint*, arXiv:2006.16668 (2020).

[95] **Riquelme, C., et al.**, "Scaling vision with sparse mixture of experts.", *Advances in Neural Information Processing Systems*, vol. 34, pp. 8583-8595, 2021.

LIST OF PUBLICATIONS

[1] **Hashmmath Shaik, Taathvika M, Sai Tharun Peram, Sree Nidhi L, Rajiv Senapati.**, "MedXPERT: A Novel ML and G-AI based framework for Disease Diagnosis"., *6th International Conference On Computational Intelligence & Data Engineering (ICCIDE-2024).*, **(Communicated)**.

[2] **Hashmmath Shaik, Taathvika M, Sai Tharun Peram, Sree Nidhi L, Rajiv Senapati.**, "Spectate-GPT: A Sensible Large Vision-Language Model for Image Comprehension"., *International Conference on Data Analytics and Cyber Security (DACS-2024).*, **(Communicated)**.