



Computational Science and Engineering
(International Master's Program)

Technische Universität München

Master's Thesis

Object Recognition with LLMs

Aristotelis Tsoutsanis





Computational Science and Engineering (International Master's Program)

Technische Universität München

Master's Thesis

Object Recognition with LLMs

Objekterkennung mit LLMs

Author: Aristotelis Tsoutsanis
Examiner: Univ.-Prof. Dr. Felix Dietrich
Advisor 1: M.Sc. Mathias Sundholm (PreciTaste)
Advisor 2: M.Sc. Alexander Dolokov (PreciTaste)
Submission Date: September 24th, 2024



I hereby declare that this thesis is entirely the result of my own work except where otherwise indicated. I have only used the resources given in the list of references.

September 24th, 2024

Aristotelis Tsoutsanis

Acknowledgments

I would like to express my deepest gratitude to all those who have supported me throughout my journey in completing this master's thesis.

This thesis was only possible with the support of M.Sc. Mathias Sundholm and M.Sc. Alexander Dolokov from PreciTaste. I appreciate them for providing a weekly feedback and support during the thesis period. I am also grateful to PreciTaste for providing all the necessary hardware required for the research and development of this work.

I am grateful to Prof. Dr. Felix Dietrich for agreeing to examine the thesis and for his methodical feedback and discussions throughout the thesis.

Lastly, I want to thank my family and friends for their constant support and encouragement. Your belief in me kept me going, and I am truly grateful for everything.

Abstract

This thesis delves into the development of Vision-Language Models (VLMs) that utilize pre-trained backbones, aiming to make these models more efficient and accessible by reducing the computational resources needed for training. With the rise of Large Language Models (LLMs) in recent years, we have seen remarkable progress in natural language processing, achieving near-human performance on a wide range of tasks. Meanwhile, visual recognition has remained a critical challenge in computer vision, playing a pivotal role in fields like robotics and autonomous driving. Vision-Language Models combine the strengths of visual and textual data, enabling them to tackle complex tasks like image captioning and visual question answering with high accuracy.

In this research, we utilize a two-stage training approach: pre-training and fine-tuning. During pre-training, we focus on transforming image embeddings into the text embedding space using adapters. This process involves minimizing the Earth Mover’s Distance between the image embedding distribution from the image encoder and the text embedding distribution of the LLM to ensure the embeddings align well. In this way, the LLM is not part of the training, significantly lowering computational costs. In the fine-tuning stage, the LLM is brought back into the pipeline, and we use the quantized version of the LLM and we apply Low-Rank Adaptation (LoRA) [1] to fine-tune the model. Meaning that instead of updating the whole weight matrix, we update low-rank matrices that approximate the necessary adjustments. We explore three types of adapters: a simple Multi-Layer Perceptron (MLP) adapter that provides a strong baseline, and two more sophisticated transformer-based adapters that utilize attention mechanisms to enhance performance and alignment between modalities. The first one contains blocks of self-attention and feed-forward directly to the image tokens, while the second one employs learnable queries that learn to selectively extract the most relevant image tokens using self-attention and cross-attention.

Our experiments, conducted on the MSCOCO [2] dataset show that these pre-trained adapters are effective for handling visual-language tasks. However, the fine-tuning phase is essential for refining the model’s accuracy and ability to generate well-structured responses. By omitting the LLM during pre-training, our approach makes it feasible for individuals and smaller organizations to work with multi-modal models, broadening access to this advanced technology. The pre-training alignment facilitates a smoother and more effective fine-tuning process, leading to faster convergence and better overall performance. Moreover, the Food101 [3] dataset was used for finetuning our pipeline for classification tasks in order to quantify the performance of our architecture.

In summary, this thesis addresses the challenges of scalability and accessibility in vision-language models. We demonstrate that TerraAlign can be trained efficiently for image captioning on the MSCOCO dataset and for classification on the Food101 dataset that shows optimistic results.

Contents

1	Introduction	1
2	State of the Art	4
2.1	Attention and Transformer	4
2.2	Visual Features	6
2.3	Vision Transformers	7
2.4	Mapping between modalities	8
3	Vision Large Language Model using pre-trained backbones	15
3.1	Problem Definition and Goals	15
3.2	Methodology - TerraAlign	15
3.3	MSCOCO Dataset	24
3.4	Training details	25
3.5	Analysis	26
3.6	Captioning Results	30
3.7	Fine-tuning in Food101	31
4	Summary, Discussion and Future Work	38
4.1	Summary	38
4.2	Discussion	38
4.3	Future Work	40
	Bibliography	43

Abbreviations

LLM	Large Language Model
VLM	Vision-Language Model
EMD	Earth Mover's Distance
MLP	Multi-Layer Perceptron
LoRA	Low-Rank Adaptation
NLP	Natural Language Processing
DNN	Deep Neural Network
CNN	Convolutional neural network
ResNet	Residual Network
GELU	Gaussian Error Linear Unit
CoCa	Contrastive Captioners model
LLaVA	Large Language and Vision Assistant
PCA	Principal Component Analysis
GPU	Graphics Processing Unit

1 Introduction

The rapid advancements in artificial intelligence are transforming how we interact with technology, making our world more connected and intelligent. At the core of these developments lies the powerful combination of language and vision—two fundamental aspects of human experience. Language has always been central to communication, allowing us to share ideas, emotions, and knowledge. As technology continues to evolve, there's a growing need for machines that can not only understand but also generate human language effectively. This demand has led to the development of increasingly sophisticated models that bridge the gap between human communication and machine understanding, driving progress in both natural language processing and computer vision.

Language plays an important role in facilitating communication and expression in everyday life. The need for more intense interaction with machines, let more generalized models to be developed in order to fill this growing demand. Recently, a lot of breakthroughs have been made in this area, such as computational resources improvements and more and more quality datasets are being published. These developments have brought about a revolutionary transformation by enabling the creation of LLMs that can approximate human-level performance on various tasks [4]. Over the last years there has been a boom from different companies, institutions and individual contributors in the LLM world. Notably, many new and competitive LLMs have been released to the open-source community, aiding in the democratization of the field and providing new researchers with the opportunity to engage in this rapidly evolving area. However, as we pre-train larger and larger models, fine tuning all the model parameters becomes a harder task that requires a significant amount of resources. Therefore, new techniques have been developed such as LoRA [1].

Similarly, visual recognition has been a long standing challenge in computer vision research and is the cornerstone of domains such as robotics [5] and autonomous driving [6]. Recently, the novel learning paradigm of Pre-training, Fine-tuning, and Prediction has shown significant effectiveness across various visual recognition tasks [7]. In this new paradigm, a Deep Neural Network (DNN) model is initially pre-trained using readily available large-scale training data, which can be either annotated or non-annotated and then this approach still requires an additional phase of task-specific fine-tuning using labeled training data for each task.

However, drawing inspiration from recent breakthroughs in NLP, for instance, Llama and Mistral are now able to solve such a large variety of tasks that their usage is becoming more and more popular. Vision-Language Model (VLM) has gained significant interest [8], these models leverage the strengths of both visual and textual data, aiming to enhance performance across a variety of tasks by combining visual and language understanding and be able to recognize the environment without requiring new labeled data every time there is a new class. VLMs can be particularly powerful when addressing complex tasks that require an understanding of both modalities, such as image captioning and visual question answering. This thesis aims to explore VLMs using pre-trained backbones and to

propose a novel training procedure that minimizes the need for extensive computational resources.

Our work introduces an alternative approach to training Vision-Language Models by leveraging pre-trained backbones while also highlighting the critical importance of an initial pre-training step that omits the use of LLMs. By performing this pre-training without LLM, we streamline the alignment of visual and language representations. This alignment step allows the fine-tuning phase of the entire pipeline to converge faster and often reach better performance with fewer resources as shown in Figure 1. Through this research, we aim to advance the development of scalable and accessible VLMs, offering a more efficient pathway for future innovations in this field.

This thesis is organized into five main sections. Section 2 explores current state-of-the-art architectures, including transformers and vision transformers, and reviews existing VLMs built from pre-trained backbones. Section 3 introduces a novel architecture and training method for VLMs that reduces the requirement for extensive computational resources, presenting two distinct architectures. Section 4 discusses and summarizes the insights gained from the thesis and suggests directions for future research.

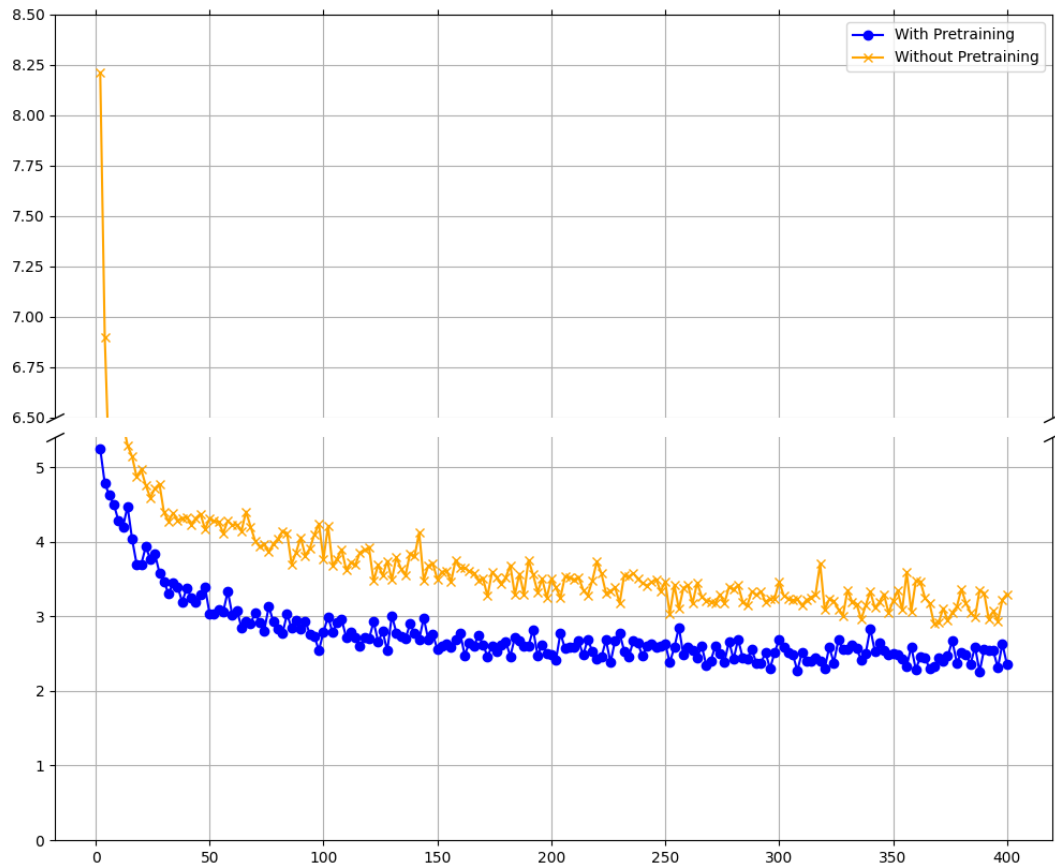


Figure 1: Fine-tune step - Validation loss comparison using the self attention adapter with and without pre-training.

2 State of the Art

2.1 Attention and Transformer

Traditional neural networks treat all parts of the input data equally when making predictions. However, in many real-world tasks, certain parts of the input may be more relevant than others. For example, in a sentence, the current word might depend more on some previous words rather than others.

The Attention mechanism [9] focuses on different parts of the input data and assigns varying weights to different elements, allowing the model to prioritize information in order to focus on the parts of the data that matter most. By calculating how relevant each part of the input is to the current step, attention helps the model to make better predictions and process information more effectively. This is especially useful in tasks like translating languages, summarizing text, and nowadays is also being used in computer vision tasks accurately.

An attention function can be described as a hash table. For a given query, we compute the dot product of the query with all the keys and apply softmax to obtain the attention weights as probabilities, which effectively highlights the relevant parts of the input. Mathematically, this process can be formulated as

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V$$

, where Q (query), K (keys), and V (values) are matrices representing the input data, and d_k is the dimension of the keys.

The above particular attention can be seen in the left part of Figure 2 and the authors called it Scaled Dot-Product Attention. However, it is beneficial to perform different attention layers simultaneously (Multi-Head attention) in order to capture different correlations, yielding in d_h output values. These are concatenated and once again projected to result to the final value, as depicted in the right part of Figure 2.

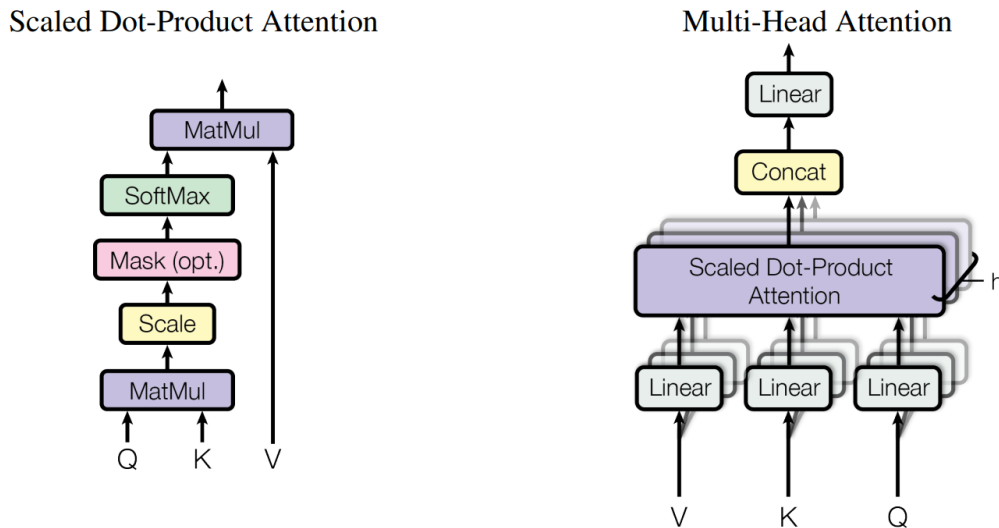


Figure 2: (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel. Image taken from [9].

Transformer-based architectures have been at the forefront on driving benchmarks in natural language processing. Especially models like GPT-4 [10], Llama 2 [11] and Mistral [12] have demonstrated remarkable capabilities in understanding and generating human language.

Transformer process sequences of tokens such as (sub-)words and the architecture is divided into two parts:

1. Encoder: This part processes the tokens from the input sequence.
2. Decoder: This part generates new tokens as output, taking into account both the input tokens and the tokens that have already been produced.

As shown in Figure 3, the transformer is an encoder-decoder architecture using stacked self-attention and point-wise, fully connected layers.

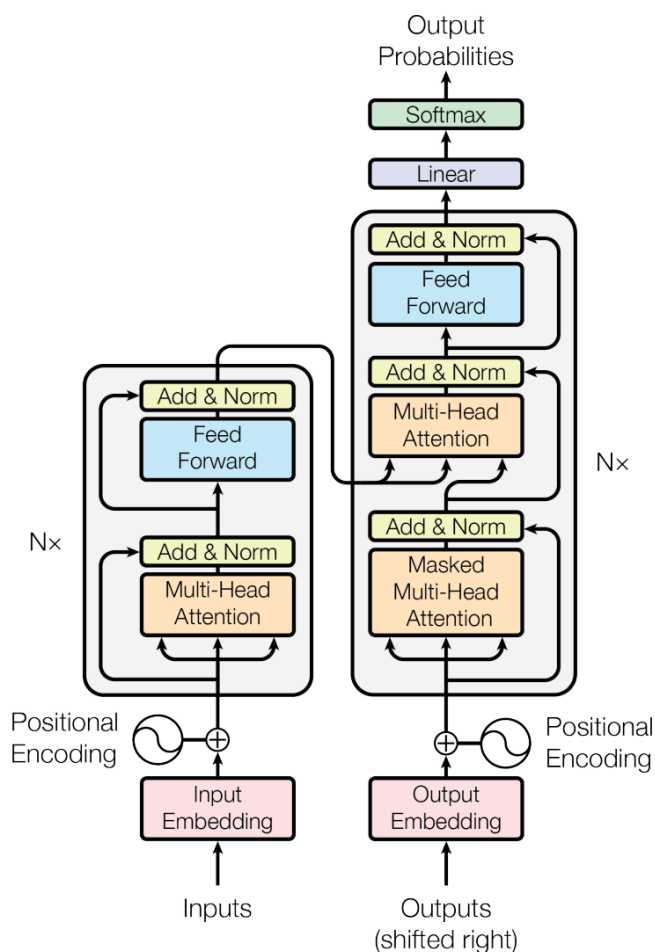


Figure 3: The Transformer - model architecture. Image taken from [9].

2.2 Visual Features

Images may have different scales, colors, and textures, making it challenging to capture their semantic information, depth, and other useful attributes. For instance, an apple may be red or green, large or small, smooth or textured. In the past, describing these features in numerical matrices required algorithms that focused on geometric transformations, such as different rotations and scales, and transformations in appearance, such as brightness and contrast adjustments.

However, in recent years, deep learning has revolutionized the extraction of visual features from images. Convolutional neural networks (CNNs), in particular, have become the standard for processing visual data. These models learn to extract the most suitable features for a given task through layers of convolutional filters that automatically detect

edges, textures, shapes, and more complex patterns as the depth of the network increases. One of the most commonly used feature extraction backbones is the idea of Residual Network (ResNet) proposed by He et al [13]. One key advantage of deep learning-based features is their exceptional generalization ability across different datasets, making them robust for various applications.

Despite these advances, traditional CNNs still face challenges in capturing long-range dependencies and contextual information within images. This is where the attention mechanism comes into play. By focusing on different parts of an image and assigning varying weights to different regions, attention mechanisms allow models to prioritize critical information and understand the image's context more effectively.

2.3 Vision Transformers

Vision transformers (ViT) [14] is an extension of the transformer architecture that was proposed for machine translation. To apply self-attention layers to an image would require to attend each pixel of the image with each other, meaning that it will be computationally expensive. Therefore, they split the image into patches and after flattening it, the patches are embed into a linear projection and positional embeddings are added on top of that. The sequence of image tokens are fed into the transformer plus the class token similar to BERT's approach [15]. After the final layer, since the model is designed for classification purposes, a Multi-Layer Perceptron (MLP) head is used for assigning the probabilities to the individual classes using the class token. The schematic of the vision transformer architecture is shown in Figure 4.

In addition, the recent years, research community has done many improvements related to optimizations in performance, adversarial training, as well as combining convolutional neural networks with attention mechanism that led to a more diverse variety of transformers mechanisms such as Swin Transformer [16] and Pyramid Vision Transformer [17].

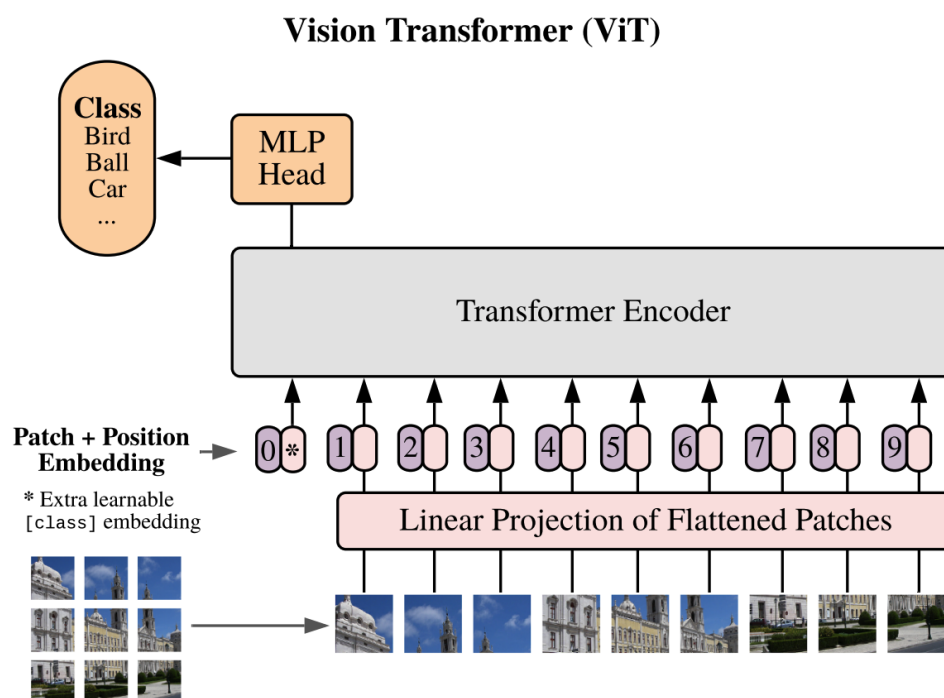


Figure 4: Vision Transformer (ViT) overview. Image taken from [14].

2.4 Mapping between modalities

Nowadays, real-world models are required to work on multiple modalities such as image, text, voice, etc., and each modality has a different encoding. Multi-modal models are designed to handle and integrate information from these diverse sources, enabling a more comprehensive understanding and interaction with data. For instance, in applications like image captioning, models must combine visual data from images with textual data to generate accurate descriptions. Similarly, in tasks such as video analysis, models need to process and integrate visual frames with audio signals. The integration of multiple modalities poses challenges, such as aligning the different types of data and ensuring that the model can effectively leverage the complementary information.

In 2021, Radford et al. proposed the Contrastive Language Pre-Training (CLIP) [18], widely considered a pioneer work in unifying visual and text information into a common latent space, where cosine distances could directly compute feature similarities.

CLIP's training process, as illustrated in Figure 5, is centered on a contrastive learning objective. The model simultaneously processes pairs of images and corresponding text descriptions, learning to associate images with their matching text while distinguishing them from non-matching pairs. By doing so, CLIP effectively learns a rich and generaliz-

able understanding of both visual and linguistic content, making it highly versatile across different tasks.

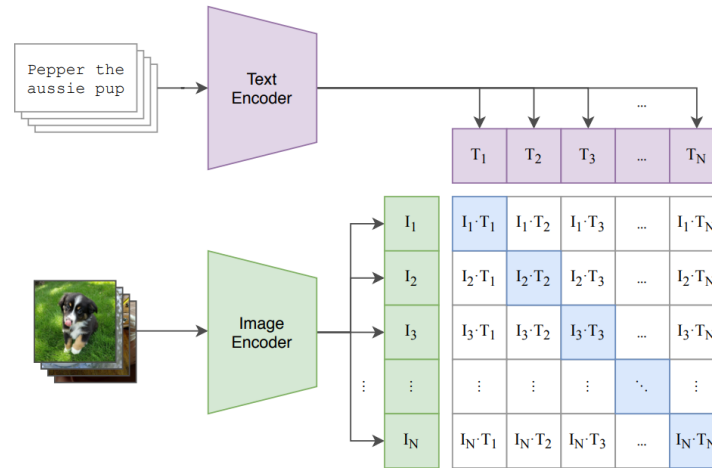


Figure 5: CLIP uses paired image-text inputs for training each of the text and vision encoders. The diagonal elements become the positive samples, and the non-diagonal elements become the negative samples for calculating the contrastive loss across all input combinations. Image taken from [18].

Contrastive Captioner (CoCa)

Recently, a lot more similar methods have been built on top of CLIP, among the approaches to multi-modal, the Contrastive Captioners (CoCa)[19] model, which is a minimalist design to pretrain an image-text encoder-decoder foundation model, stands out as an innovative approach in the field of vision-language models that combines the strengths of both contrastive learning and caption generation. It is designed to bridge the gap between image understanding and language generation, leveraging the principles of contrastive learning to enhance the quality and robustness of generated captions following the work from CLIP.

As shown in Figure 6, CoCa consist of a visual transformer image encoder and a text encoder that are jointly trained by contrasting the paired text against others. While the dual-encoder approach encodes the text as a whole, the captioner approach enable to the model to predict text tokens autoregressively.

Overall, contrastive captioner (CoCa) is similar to standards image-text encoder-decoder architectures, encodes images using vision transformer (ViT) to a latent representation, and decodes texts with a causal masking transformer decoder. In the decoder part, CoCa cross-attend the uni-modal text representation in the first half of the decoder layers and cross-attends the image tokens from the encoder for multi-modal image-text representations.

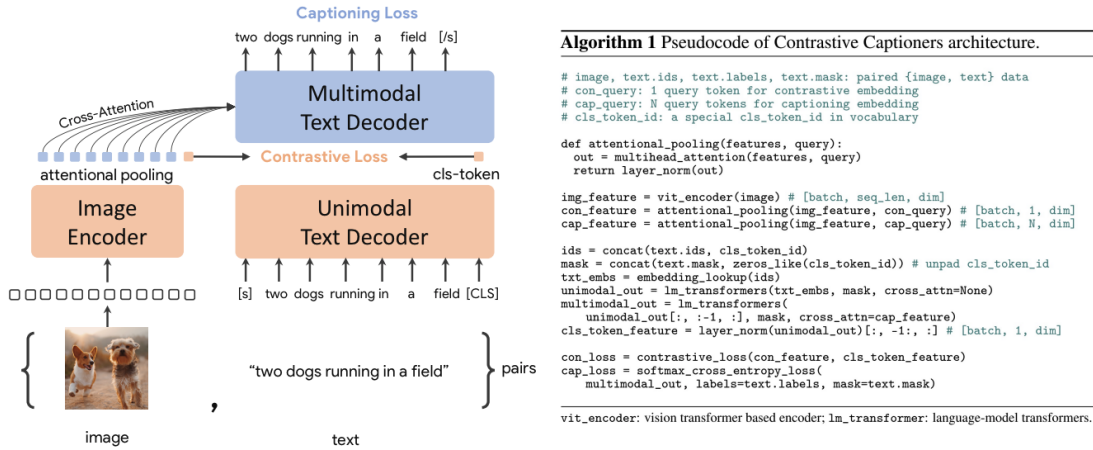


Figure 6: Detailed illustration of CoCa architecture and training objectives. Image taken from [19].

LLaVa

Auto regressive Large Language Models have shown impressive capabilities within text domain; however, such large scale language models are ‘blind’ to modalities other than text, making it difficult to communicate visual tasks, questions or concept to them. To address this limitation, many researchers have explored various approaches to bridge the gap between the visual and the textual spaces. These approaches often involve techniques such as cross-modal embeddings, where visual and textual data are mapped into a common latent space, and attention mechanisms that allow the model to focus on relevant parts of the input from different modalities. By minimizing the gap between visual and textual data, these methods aim to enhance the ability of large language models to understand and generate multi-modal content, thereby expanding their applicability to a broader range of real-world tasks.

Continuing with the exploration of multi-modal integration techniques, LLaVa [20] introduces the first attempt to visual instruction tuning. Especially, LLaVa tries to extend the instruction-tuning to the multi-modal space of language and image by utilizing both a pre-trained Large Language model and a pre-trained vision encoder. For instance, to allow Large Language models to answer real-world tasks and follow language instructions, many methods have been explored for instruction-tuning such as InstructGPT [21].

The network architecture is shown in Figure 7. Given an input image X_v into the pre-trained vision encoder that provides the visual feature $Z_v = g(X_v)$. A simple trainable linear projection layer is used to project the image features into the word embeddings space. Therefore, a sequence of visual tokens H_v which $H_v = W \cdot Z_v$ is concatenated with the sequence of text tokens into the Language Model. In this case, LLaVa researchers picked a pre-trained CLIP visual encoder ViT-L/14 [18] and a Vicuna [22] as the Large

Language Model.

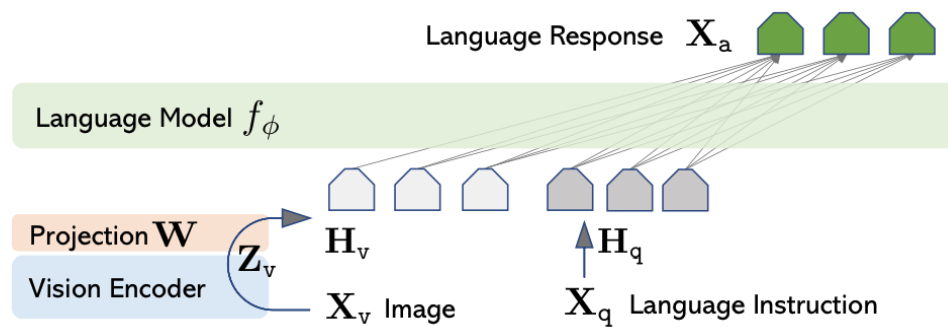


Figure 7: LLaVa network architecture. Image taken from [20].

The training process of LLaVa involves two main stages: pre-training and fine-tuning. During the pre-training stage, the vision encoder and the language model weights are frozen and only the adapter is being trained. In this way the image features H_v can be aligned to the word embedding space of the LLM. During the fine-tuning step, the visual encoder weights are still frozen, however this time the pre-trained weights of the projection layer and the LLM are being updated. Especially, the LLaVa model was trained for 1 epoch in the pre-training step and for 3 epochs in the fine-tuning step on two different datasets.

As shown in Table 8, the example requires a depth image understanding. The model is able to follow the user's instructions and answer the questions accurately.

Visual input example, Extreme Ironing:



Source: <https://www.barnorama.com/wp-content/uploads/2016/12/03-Confusing-Pictures.jpg>

User

What is unusual about this image?

LLaVA

The unusual aspect of this image is a man ironing clothes on the back of a minivan or van. This is not a typical place to perform this activity, as one would usually iron clothes in a more stationary and safe location, such as a home, using a regular ironing board. The scene depicted in the image is peculiar as it involves a makeshift ironing setup on a vehicle, which can be both unsafe and unconventional. Additionally, it is not clear how the man is able to maintain balance and stability while ironing clothes in such an unstable environment.

[Start a new conversation, and clear the history]

User

What's happening in the scene?

LLaVA

The image depicts a man standing on top of a yellow SUV in a busy city street. He is holding a portable ladder, seemingly ironing clothes while standing on the vehicle. Around the scene, there are other cars, a traffic light, and a fire hydrant. Additionally, there are two other people in the vicinity, one close to the man on the car and another further away. The man on the ladder appears to be attracting attention **due to his unconventional choice of ironing his clothes on top of a moving car**. The city street around him is bustling with activity, adding to the unique nature of the scene.

Figure 8: Extreme Ironing example. Image taken from [20].

BLIP-2

BLIP-2 [23] follows the same paradigms as LLaVa, meaning that the authors try to fill the gap between the visual embedding and the text embedding space from pre-trained backbones as shown in Figure 9. In this case, BLIP-2 proposes Q-Former as the trainable adapter to bridge the gap between a frozen image encoder and a frozen Large Language model. It uses learned queries to extract a fixed number of features from the image encoder.

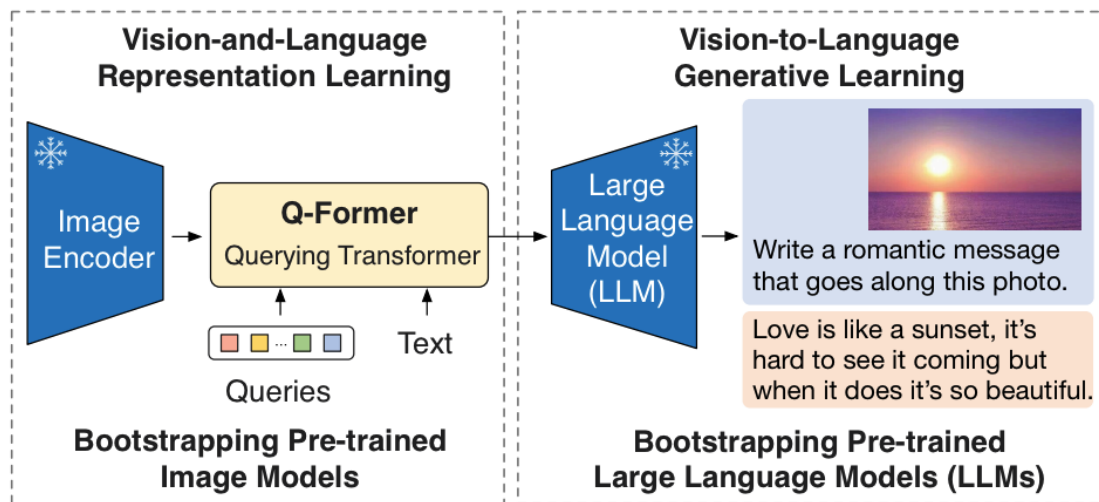


Figure 9: Overview of BLIP-2’s framework. Image taken from [23].

As shown in Figure 10, Q-Former includes two transformer submodules with shared self-attention layers: (1) an image transformer interacting with the frozen image encoder for visual feature extraction, and (2) a text transformer functioning as both an encoder and a decoder. Learnable query embeddings serve as input to the image transformer, interacting with each other via self-attention and with image features through cross-attention layers. Queries can also interact with text through self-attention layers, with different masks applied based on the pre-training task. Q-Former is initialized with BERTbase weights [15], except for the randomly initialized cross-attention layers, and contains 188 million parameters, including the queries.

The Q-Former jointly optimizes three objectives:

1. **Image-Text Contrastive Learning** learns to align the image-text pairs that are similar against those that are negative pairs.
2. **Image-grounded Text Generation** loss trains Q-Former to generate texts. Since the text tokens do not communicate with the visual encoder. It uses the learned queries to extract all the necessary information from the text and then passes the text tokens into the self-attention layers. Thus, the queries are forced to capture visual features that describe all the necessary information about the text.
3. **Image-Text Matching** aims to learn a fine-grained alignment between image and text. Especially, it is a classification problem of whether it is a positive or negative pair.

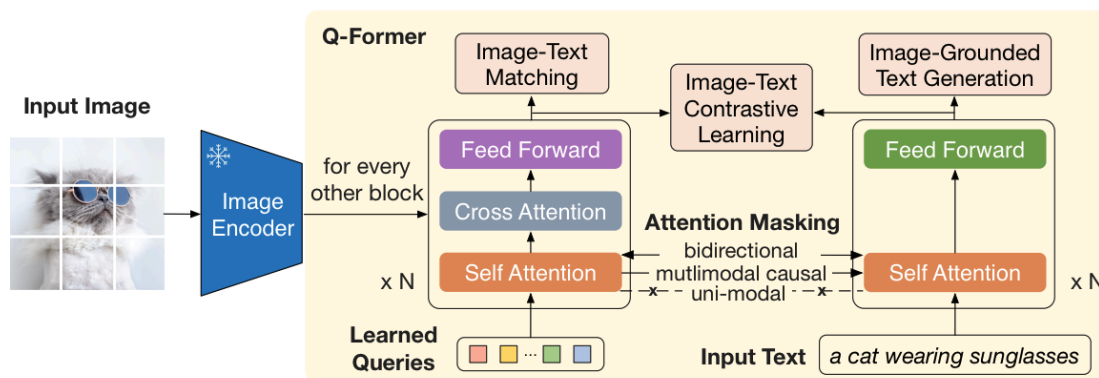


Figure 10: Model architecture of Q-Former and BLIP-2’s first-stage vision-language representation learning objectives. Image taken from [23].

Pre-trained image encoder and LLM, as for the frozen visual encoder, the authors explored two different vision transformer models: ViT-L/14 [18] and ViT-g/14 from EVA-CLIP [24] and for the Large Language model they explored the OPT [25] model family as a decoder-based LLMs.

3 Vision Large Language Model using pre-trained backbones

3.1 Problem Definition and Goals

Training a VLM is computationally expensive and demands substantial GPU resources. This complexity arises from the necessity to process extensive datasets through both pre-trained visual encoders and pre-trained LLMs. Despite leveraging these pre-trained components, the process still involves a significant computational overhead due to the requirement of forwarding passes through the LLM. Consequently, a critical challenge is to devise an efficient strategy to map visual embeddings into the text embeddings distribution without relying on the entire LLM, at least during the initial training phase (pre-train step).

Our primary goal is to streamline this process by utilizing techniques that minimize the reliance on full-scale LLM operations early in the training. By doing so, we aim to achieve a balance between computational efficiency and model performance, enabling more feasible training of VLMs on available hardware resources. This approach not only makes the training process more accessible but also opens the door to more scalable and adaptable VLM architectures, which can be fine-tuned with fewer resources while maintaining high performance.

3.2 Methodology - TerraAlign

Our approach is based on building a Visual Language Model (VLM) using pre-trained models, leveraging their powerful representations and transfer learning capabilities.

Similar to LLaVa and BLIP-2, our training setup consist of two different phases. While the second phase, known as fine-tuning step, is similar to the two aforementioned models, meaning that we minimize the Cross-Entropy loss [26] on token level autoregressively, the first training phase or pre-training step differs significantly. In this phase, we omit the LLM and we directly learn the adapter to map the visual embeddings into the text embeddings space that makes the pipeline more efficient and less computational expensive. We name our architecture TerraAlign since we aim to align the visual embeddings into the text embeddings space and because we employ the Earth Mover’s Distance to minimize the semantic gap between these two modalities. A high-level overview of the pre-training step is shown in Figure 11 and of the fine-tuning step in Figure 12.

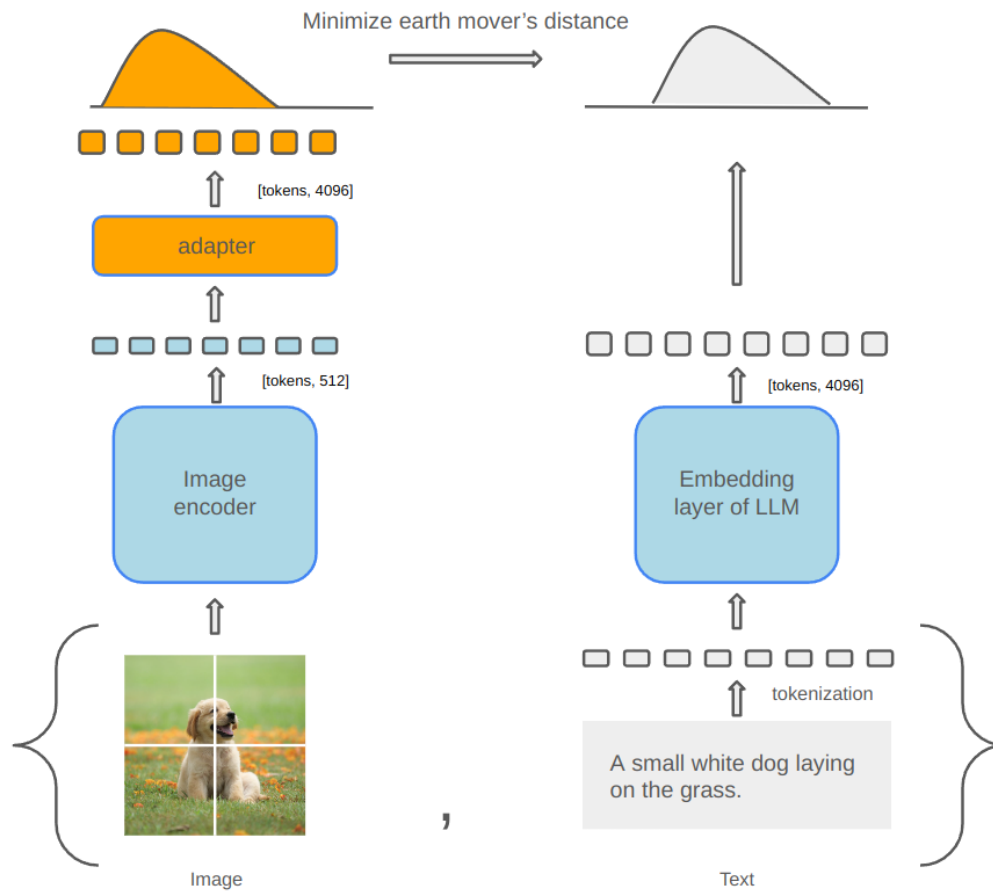


Figure 11: Our pre-training setup without utilizing the LLM. The blue color shows the frozen components and the orange the trainable.

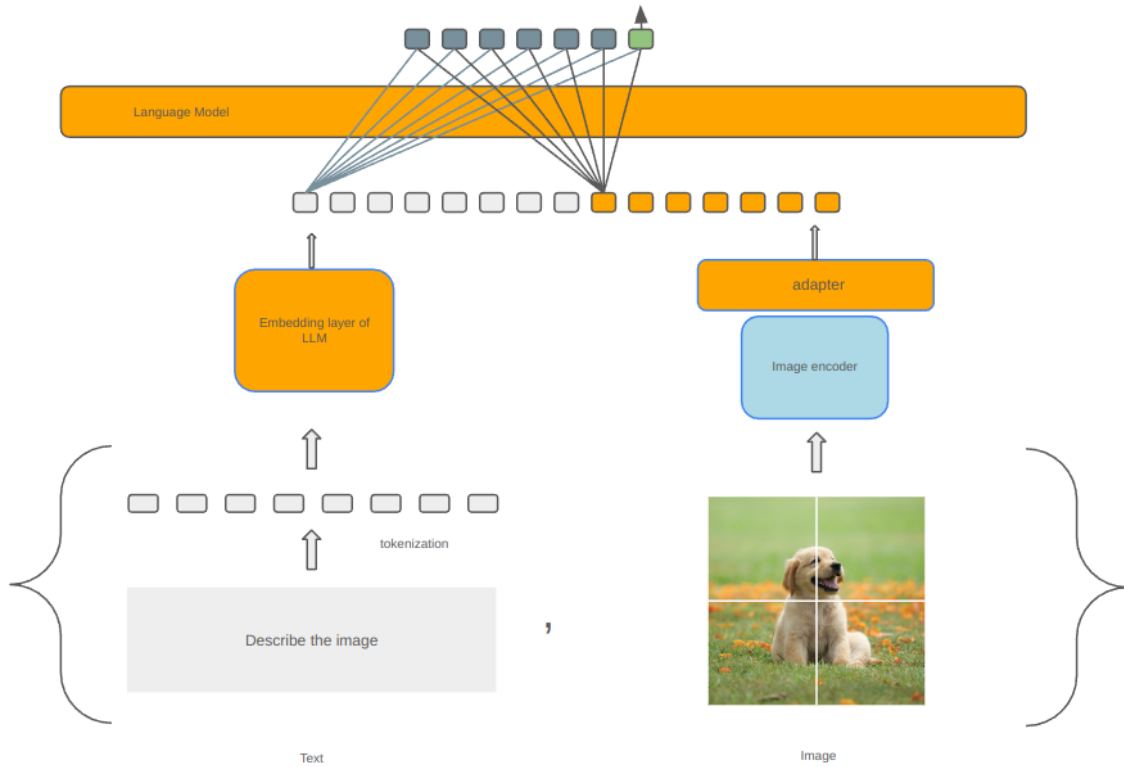


Figure 12: Our fine-tuning setup with the LLM. The blue color shows the frozen components and the orange the trainable.

Visual and Text Embeddings

For the **vision encoder**, we utilize the CoCa model [19], which has demonstrated robust performance in tasks involving image-text alignments. Specifically, we focus on the visual features obtained after the attentional pooling layer of the CoCa model. These features have a dimension of $[batch, N, dim]$, where N represents the number of query tokens for captioning embeddings (255 in our case), and dim is the embedding dimension (512).

For the **text encoder**, we utilize the quantized version of the Intstruct Mistral-7B [12], a state-of-the-art large language model. This model has been fine-tuned for instruction-based tasks, making it adapt at understanding and generating human-like text based on given prompts. The quantization process reduces the model size and computational requirements while maintaining high performance, making it suitable for our large-scale multimodal tasks.

Earth Mover's Distance

The Earth Mover's Distance (EMD) [27], also known as the Wasserstein distance, is a metric used to quantify the difference between two probability distributions. It does this by calculating the minimum amount of 'work' required to transform one distribution into the other. Intuitively, one can imagine one distribution as a mass of earth spread across a space and the other distribution as a set of holes in that space. The EMD represents the least amount of work needed to fill the holes with the earth, where a unit of work is defined as transporting a unit of earth over a unit of distance.

Originally proposed in the context of transportation theory, however, lately the EMD has been used for image retrieval and especially to compare different images by treating each image's pixel distribution as a signature. For example, in content-based image retrieval, the EMD can measure the dissimilarity between two images by computing the minimal effort needed to transform one image's pixel distribution to match the other's. This approach allows for more accurate image comparisons, especially when dealing with images that have undergone transformations such as translation, scaling, or rotation.

In conclusion, Earth Mover's Distance is a valuable metric for comparing probability distributions, with a wide range of applications across various domains. Recent research has focused on developing approximation algorithms and data-parallel techniques to overcome the computational challenges associated with EMD, enabling its use in practical scenarios and connecting it to broader theories in machine learning and data analysis.

LoRA - Low-Rank Adaptation

LoRA is a technique that is used for efficiently fine-tuning and updating a small portion of the trainable parameters. It offers a significant advantage over traditional fine-tuning methods by minimizing the number of trainable parameters, thereby reducing computational costs.

Especially, LoRA freezes the weights of the pre-trained model in order to not be changed during the fine-tuning step. It adds rank decomposition matrices into each layer that are smaller than the original weight matrices, reducing the trainable parameters. At the end only the low rank matrices are being trained during the fine-tuning, and this allows the model to learn new tasks and keeping the already known knowledge from the original pre-trained weight matrices while reducing the memory and computational usage. A detailed visual overview is shown in Figure 13.

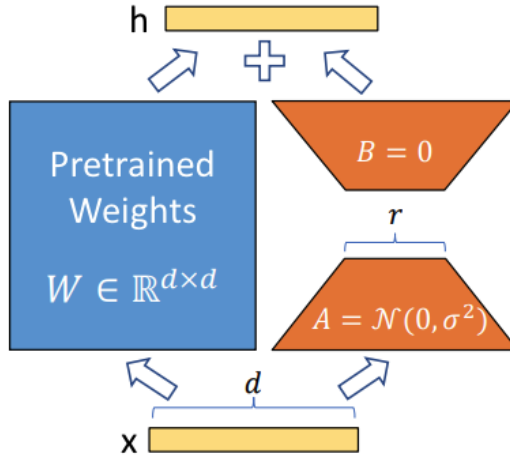


Figure 13: LoRA’s reparametrization. Image taken from [1].

Simple Adapter

In our first experiment, we utilize a simple MLP adapter to map the visual features into the text embeddings space similar to LLaVa. The simple projection model contains three main components: a fully connected layer, a GELU [28] activation function, and another fully connected layer. The architecture is defined as follows: the first fully connected layer ($fc1$) takes the input visual embeddings Z_v and projects them into a higher-dimensional space of size hidden dimension. Following this projection, we apply the GELU (Gaussian Error Linear Unit) activation function to introduce non-linearity into the model, and then it is followed by another linear layer than outputs the transform visual features H_v .

$$H_v = W_2 \cdot \text{GELU}(W_1 \cdot Z_v) \quad (3.1)$$

The choice of GELU is motivated by its smoothness and performance benefits over other activation functions such as ReLU [29] or Leaky ReLU [30], especially in terms of gradient flow and learning dynamics.

The hidden layer dimension is set to two times the output dimension, which we determined to be optimal through extensive hyperparameter tuning and empirical validation. This choice ensures that the model has sufficient capacity to capture complex mappings from the visual domain to the textual domain, while still maintaining computational efficiency.

After the non-linear transformation, the second fully connected layer ($fc2$) reduces the dimensionality from hidden dimension to output dimension, effectively mapping the processed visual features into the text embedding space. This final layer ensures that the

output dimensions match the required size of the text embeddings, facilitating seamless integration with subsequent natural language processing tasks.

This architecture was chosen for its simplicity and effectiveness, providing a robust baseline for further experimentation. More architecture details of the Simple Adapter are shown in Table 1 and Figure 14.

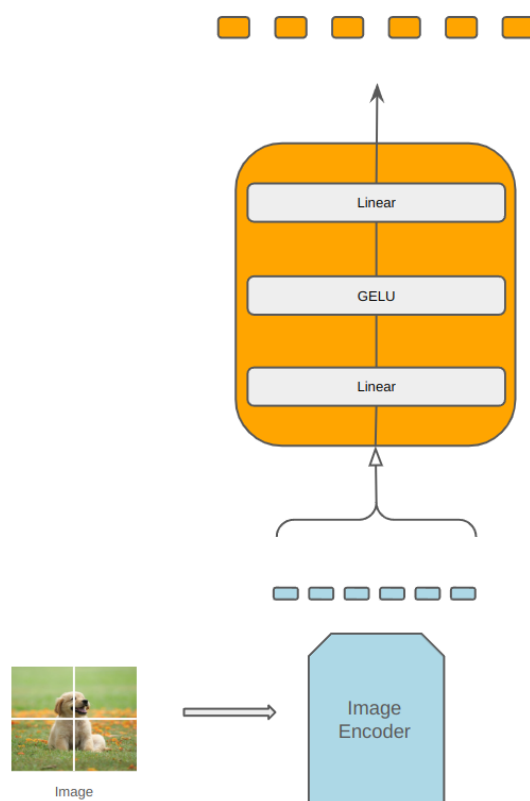


Figure 14: Our simple adapter architecture

Transformer-based Adapter with self attention

To enhance the capability of our visual-to-text mapping, we employ a more advanced adapter, replacing the simple linear projection with a transformer-based adapter. This new adapter incorporates self-attention mechanisms, which are crucial for capturing intricate relationships and dependencies among the image tokens produced by the vision encoder. As shown in Figure 15, each block within this transformer-based adapter consists of a self-attention mechanism followed by a feed-forward neural network, collectively working

Table 1: Architecture details of the Simple Adapter and the pre-trained backbones

Module	Details
Vision Encoder	CoCa ViT-B-32
Image embedding dimension	[batch, 255, 512]
LLM	Mistral-7B-Instruct
Text embedding dimension	[batch, sequence tokens, 4096]
Adapter architecture	[Linear, GELU, Linear]
layer: fc1	(512, 8192)
layer: fc2	(8192, 4096)
output dimension	[batch, sequence tokens, 4096]

to refine the representations of the image tokens. These components work together to translate the image embeddings into the text embedding space, ensuring that the visual information is accurately translated. More information related to the architecture of the adapter is listed in Table 2. At the end a linear layer is used to map the image tokens into the correct dimension, matching the text tokens.

Table 2: Architecture details of the transformer-based Adapter with self-attention and the pre-trained backbones

Module	Details
Vision Encoder	CoCa ViT-B-32
Image embedding dimension	[batch, 255, 512]
LLM	Mistral-7B-Instruct
Text embedding dimension	[batch, sequence tokens, 4096]
Adapter main query block	[Self-Attention, Feed Forward]
Number of blocks	4
Number of heads	4
Output dimension	[batch, 255, 4096]

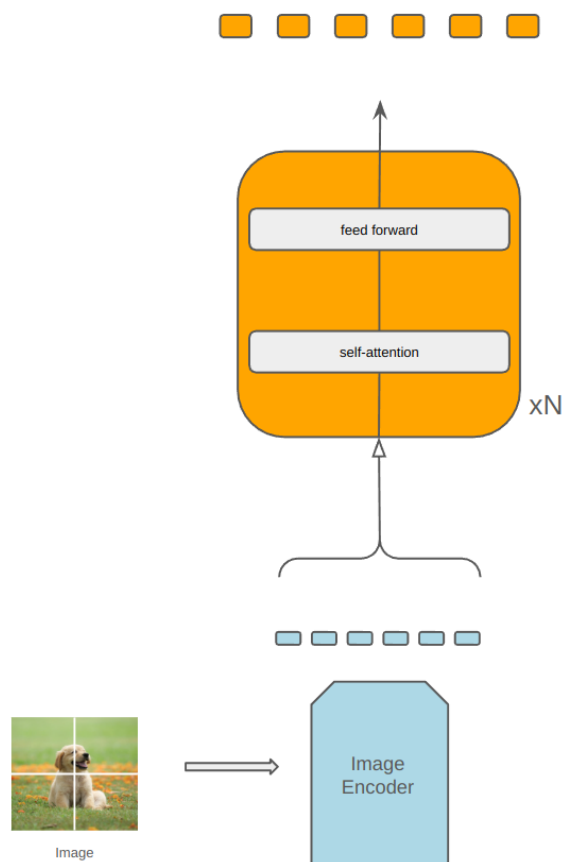


Figure 15: Our transformer-based adapter with self attention

Query transformer-based Adapter with learnable queries

To improve our adapter, we introduce a more efficient architecture utilizing and filtering the image tokens in order to select precisely only the useful image tokens. Inspired by the Q-Former architecture, this approach uses a query-based transformer architecture to encode images more effectively. The core idea is to utilize learnable queries to filter and emphasize the image tokens that capture the essential knowledge relevant to the task.

As illustrated in Figure 16, the Query Adapter starts by selecting a fixed number of learnable queries, which can range from 32 to 255, in our case we use 128 as trade-off between accuracy and efficiency. To initialize these tokens, we adopt a similar method to the BERT model’s initialization process. Specifically, we use a normal distribution with a mean of 0.0 and a standard deviation based on BERT’s configuration, ensuring that the initial values are set up in a way that is consistent with BERT’s training practices, promoting stable and effective learning.

In addition, these queries interact with each other through self-attention layers and with the image tokens from the visual encoder via cross-attention layers. This interaction allows the model to focus on the most pertinent visual features, thereby improving the integration of visual and textual information.

The use of learnable queries enables the model to dynamically adjust its focus based on the input image, providing a more flexible and powerful mechanism for visual feature extraction compared to the simple adapter. More information about the architecture is listed in Table 3.

Table 3: Architecture details of the Query transformer-based Adapter and the pre-trained backbones

Module	Details
Vision Encoder	CoCa ViT-B-32
Image embedding dimension	[batch, 255, 512]
LLM	Mistral-7B-Instruct
Text embedding dimension	[batch, sequence tokens, 4096]
Adapter main query block	[Self-Attention, Cross-Attention, Feed Forward]
Number of blocks	4
Number of heads	4
Number of learnable queries	128
Output dimension	[batch, number of learnable tokens, 4096]

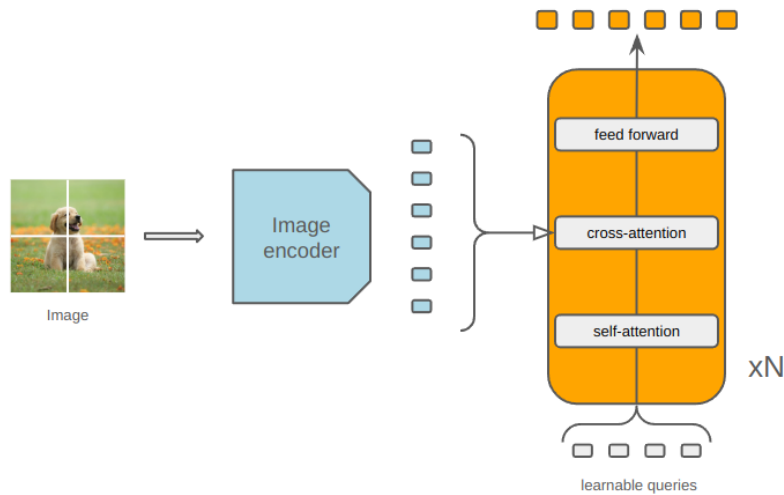


Figure 16: Our Query transformer-based Adapter with learnable queries

3.3 MSCOCO Dataset

In our case, we utilize the COCO dataset [2] and especially the COCO captions that contains captions for over 330k images for training and validation. We picked this dataset due to its rich and diverse collection of images paired with natural language descriptions, making it ideal for training visual-language models. A few examples can be found in Figure 17.

Data preparation

To prepare the dataset for training, we preprocess both the images and captions to ensure compatibility with our model architecture. The images are transformed by applying the necessary transformations required by the vision encoder. Captions are tokenized and encoded into the embedding space of the LLM. This involves breaking down the text into tokens, converting these tokens into numerical representations, and mapping them into the LLM's embedding space.



The man at bat readies to swing at the pitch while the umpire looks on.



A large bus sitting next to a very tall building.



A horse carrying a large load of hay and two people sitting on it.



Bunk bed with a narrow shelf sitting underneath it.

Figure 17: Example images and captions from the Microsoft COCO Caption dataset. Image taken from [2].

3.4 Training details

Our training process involves two main steps: pre-training and fine-tuning.

Pre-training Step: As previously discussed, in the pre-training phase, we simplify the model by removing the LLM component, allowing us to focus solely on the adapter. During this phase, our goal is to minimize the EMD between the projected visual embeddings and the target text embeddings. This approach helps ensure that the visual features are well-aligned with the textual domain before we integrate the language model.

Fine-tuning Step: In the fine-tuning phase, we reintroduce the LLM and follow the training paradigm used by current state-of-the-art models with the only difference that we utilize LoRA and we train only the 0.0470% of the parameters due to computational limitations. During this step, the vision encoder remains frozen, meaning its parameters are not updated. We fine-tune the rest of the model, including both the adapter and the LLM, to better align the visual and textual features for specific tasks.

In particular, we begin by passing a prompt message, such as "Describe this image" or "Provide a brief description of the given image" through the tokenizer and embedding layer of the LLM. This processed prompt is then concatenated with the image tokens generated by the adapter. By integrating these prompt-driven embeddings with the image tokens, we provide the LLM with contextual guidance that enhances its understanding and generation capabilities. The LLM then generates a response, during which it aims to minimize the Cross-Entropy loss between the predicted tokens and the ground truth tokens. This loss function effectively guides the model to produce outputs that are more accurate and contextually relevant.

Dataset and Split: For both training steps, we use the MSCOCO dataset. We split the dataset into 70% for training and 30% for validation and testing, ensuring a reliable validation of our model's performance.

By structuring our training process this way, we aim to first establish a strong alignment between visual and textual features during pre-training, and then refine and optimize this alignment during fine-tuning to improve the model's overall performance on visual-language tasks.

Training details - Simple Adapter

For the Simple Adapter, in the pre-training step, we train our model for four epochs to make sure that the visual and text embeddings aligned. During the fine-tune step, we train for another four epoch, during which we fine-tune both the adapter and the LLM by applying Low-Rank Adaptation (LoRA). Details about the training are listed in Table 4.

Training details - transformer-based Adapter with self attention

Similar to the simple adapter, we train in two stages. In the pre-training step, we train our model for four epochs to make sure that the adapter is able to translate the visual

Table 4: Training details for the simple adapter

Pre-training	Details
Epochs	4
Batch size	8
Fine-tuning	Details
Epochs	4
Batch size	8
LoRA r	8
LoRA alpha	32
LoRA dropout	0.1

embeddings to the text embeddings space. During fine-tuning, our adapter and the LLM are trained together for five epochs. Details about the training are listed in Table 5.

Table 5: Training details for the self attention adapter

Pre-training	Details
Epochs	4
Batch size	8
Fine-tuning	Details
Epochs	5
Batch size	8
LoRA r	8
LoRA alpha	32
LoRA dropout	0.1

Training details - Query transformer-based Adapter with learnable queries

We follow the same settings as the above transformer-based adapter with the self-attention layer. We train for four epochs in the pre-training step, and then for five epochs in the finetuning step. Details about the training are listed in Table 6.

3.5 Analysis

In this section, we take a small subset of (image, text) pairs from the validation set in order to visualize the alignment. Each image and text token is treated as a data point. All image tokens are passed through the adapter to be transformed into the text embeddings space, while all text tokens are processed through the embedding layer of the LLM, transforming them into the same embedding space. To facilitate visualization, we perform a

Table 6: Training details for the query adapter with learnable queries

Pre-training	Details
Epochs	4
Batch size	8
Fine-tuning	Details
Epochs	5
Batch size	8
LoRA r	8
LoRA alpha	32
LoRA dropout	0.1

Principal Component Analysis (PCA) [31], reducing the high-dimensional embeddings to a 2D space.

Analysis - Simple Adapter

Pre-training: As shown in Figure 18, after the first epoch of training, the image tokens (blue) begin to move closer to the text tokens (red), indicating initial progress in alignment. However, this early stage of training shows that further training is necessary for more accurate alignment. By the fourth epoch, depicted in Figure 19, the tokens show significantly better alignment. The image tokens are now closely aligned with the text tokens, demonstrating that the adapter effectively transforms the image tokens into the text embedding space. A zoomed view of this alignment is provided in Figure 20, highlighting the overlap of the image and text tokens.

To further evaluate the performance of our adapter, we visualize the image tokens in the embedding space concerning the entire vocabulary of our LLM. For this purpose, we select a few image examples and follow the previously described procedure. The results, shown in Figure 21, indicate that the adapter successfully maps the image tokens into the vocabulary space. The image tokens align predominantly with the text tokens in the center-right part of the Figure. This concentration occurs because of the small sample and because the adapter has been trained to transform image tokens into the text embedding space corresponding to plain English captions, while the LLM’s vocabulary includes a broader range of characters and tokens beyond plain English.

Fine-tuning: After fine-tuning, we examined the behavior of the image and text tokens. Upon attempting to visualize them in a 2D space, we found that the first two principal components captured only 15% of the variance, making meaningful analysis and visualization infeasible. To better understand how much information is needed to represent the data accurately, we calculated that the first 68 principal components are necessary to capture more than 80% of the information. This shows that the embeddings are quite complex and high-dimensional, and reducing them to just two dimensions loses a lot of important

details.

Therefore, we observe that, while fine-tuning improves the model's performance, it also increases the complexity of the embeddings, making simple 2D visualizations insufficient for a detailed analysis.

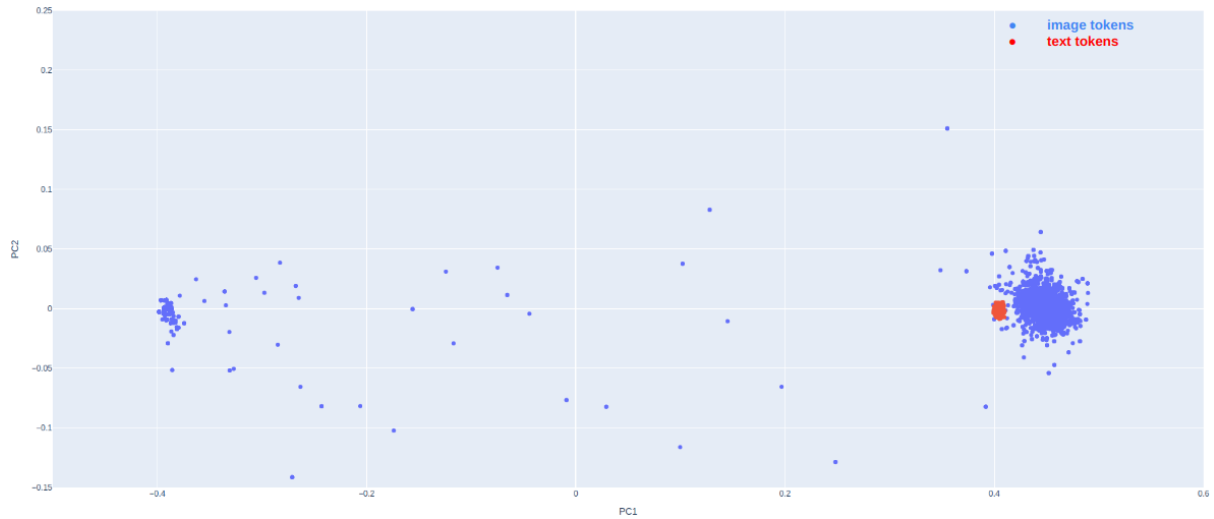


Figure 18: Simple Adapter: PCA on image and text tokens at epoch 1

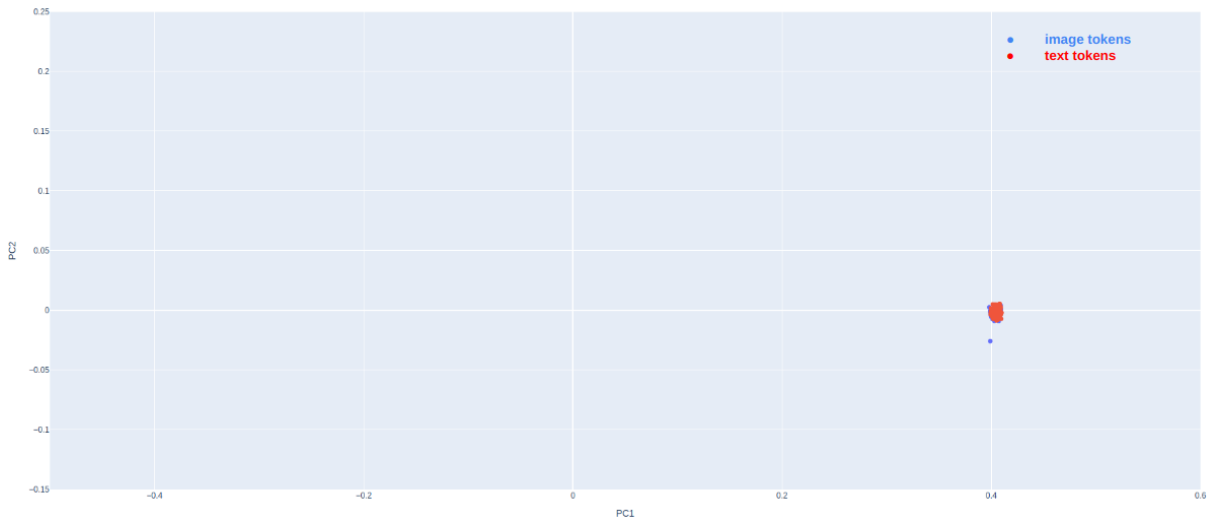


Figure 19: Simple Adapter: PCA on image and text tokens at epoch 4

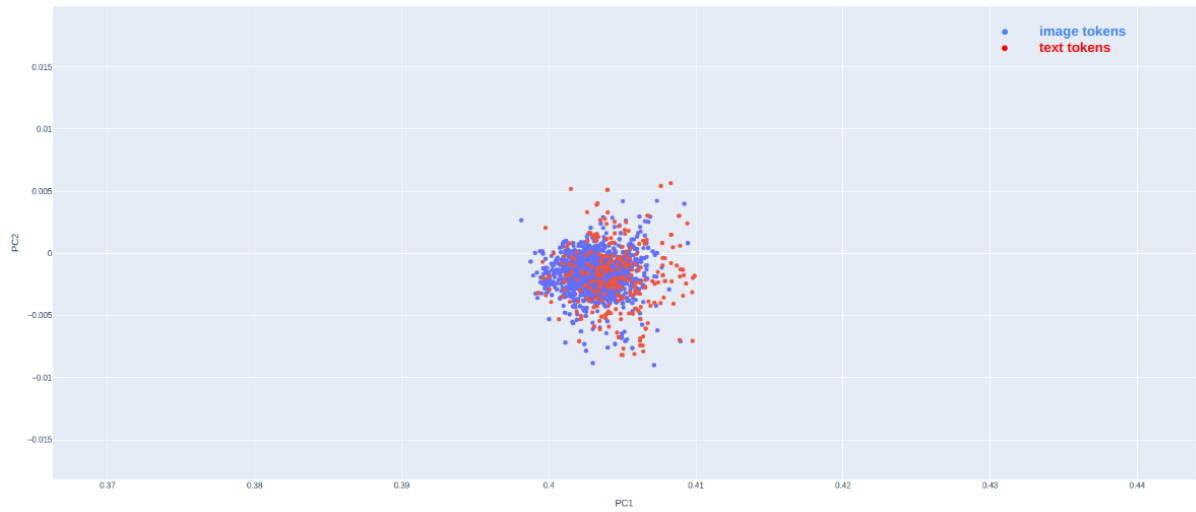


Figure 20: Simple Adapter: Zoomed-in PCA on image and text tokens at epoch 4

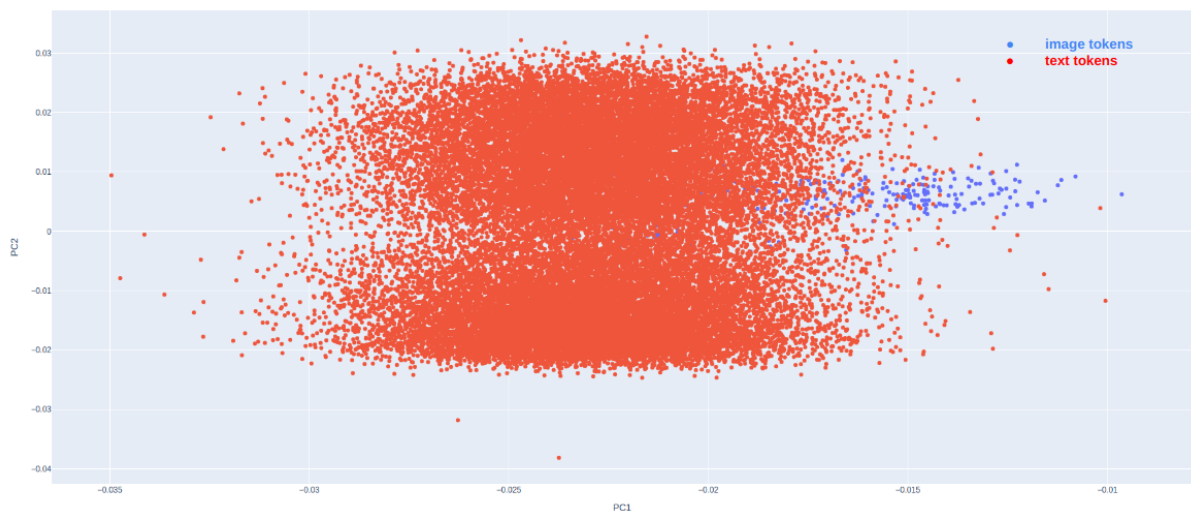


Figure 21: Simple Adapter: PCA Analysis with respect to the whole vocabulary in epoch 4

Analysis - Self Attention Adapter

In the self attention transformer-based adapter, during this process, we observed that visualizing the embeddings in just two dimensions is challenging. This difficulty arises due

to the high variance present in the embedding, which leads to significant information loss when attempting to reduce the dimensionality to such a low level.

To better understand the structure of the embeddings, we calculated the number of principal components required to retain a substantial portion of the information. Our analysis revealed that 14 principal components are necessary to capture more than 80% of the variance in the data. This finding highlights the complexity of the embeddings generated by the transformer-based adapter after the pre-training step. In the fine-tuning step the embeddings are even more complex and high-dimensional making it infeasible to reducing them to just two dimensions in both cases.

Analysis - Query Adapter with learnable queries

Similar to the self attention adapter, we experience a similar behavior of the embeddings. For instance, our analysis revealed that the embeddings are even more complex and that 33 principal components are necessary to capture more than 80% after the pre-training step.

3.6 Captioning Results

During inference, we generate captions using two approaches: first, with only the pre-trained adapter and the already trained LLM, and second, with the finetuned adapter alongside the finetuned LLM. In the initial approach, the pretrained adapter, having been trained to map image tokens to text tokens, is used to generate captions. This is achieved through carefully crafted prompts, such as "Below is a description of an image; please correct it and re-write a caption." This prompt is concatenated with the image tokens, which have been transformed to resemble text tokens related to the image content, allowing the LLM to generate a caption. Most of the time, using the pretrained approach, the model can classify and generate high-level descriptors of the image, such as identifying whether there is a human or animal present. However, it often fails to capture finer details and occasionally produces irrelevant or nonsensical captions, missing key visual information. For the finetuned pipeline, the captioning process is further refined. We use a simpler prompt, "Describe the image: ", which is also concatenated with the image tokens. In both cases, instruction tokens like [INST] and [\INST] are included to ensure that the LLM adheres to the intended instructions, as it was trained to follow such directives.

Simple Adapter

The following examples in Table 7 demonstrate that the pre-trained adapter can be used for visual-language tasks, particularly for simpler cases. This indicates that the LLM can understand the image tokens even though it has never seen them before, meaning that the image tokens are being transformed in a way that is close to other text tokens with the

same meaning. However, the pipeline is really sensitive in the prompt message and in difficult cases the pre-trained TerraAlign produces either low quality or non-sense captions or endless responses. After fine-tuning, the model performs significantly better, providing more accurate and precise answers. This improvement highlights the importance of the fine-tuning process in enhancing the model's capability to generate coherent and contextually appropriate descriptions based on visual input.

Transformer-based Adapter with self-attention

Given the more complex architecture and the ability of the adapter to learn correlations between the tokens using self-attention mechanisms and translate the image tokens into the text embeddings space more accurate. This result in a significant more improved and accurate captions, especially after finetuning our pipeline. For instance, as shown in Table 8 the results are more accurate with less mistakes and shows that the model can capture more information of the image.

Query transformer-based Adapter with learnable queries

This approach stands out for its efficiency, as it selectively filters the image tokens, reducing the overall token count by half while maintaining high accuracy. The key to this method's success lies in the learnable queries, which are designed to extract only the most relevant information from the image tokens. As a result, there is minimal information loss, allowing the model to generate highly accurate captions despite using fewer tokens. The results in Table 9 show that this method not only preserves crucial details but even create more accurate captions from the self-attention-based approach.

3.7 Fine-tuning in Food101

Classification Results

Currently, there is no standardized evaluation or benchmarking method for VLMs. Traditional metrics like BLEU are limited in that they primarily measure the similarity between the generated output and a specific ground-truth caption, without effectively capturing the underlying meaning and reasoning capabilities of the model. Consequently, these metrics may not fully reflect the performance of VLMs in more nuanced scenarios.

Given these limitations, a common approach to evaluating VLMs involves testing them on specific types of questions, such as yes/no, multiple choice, or classification tasks. For our evaluation, we chose to classify food images from the Food101 dataset [3]. The Food101 dataset is a comprehensive collection of 101,000 images spanning 101 different categories of food dishes. Each category contains 1,000 images, offering a balanced and diverse set of examples for each type of dish. 100 out of the 101 classes examples can be found in Figure 22.

Table 7: Examples from our test set using the simple adapter




Image	Pre-trained	Fine-tuned
	<p>The man in the middle of the line.</p>	<p>This is a man in a tennis outfit playing on a tennis court.</p>
	<p>A brown bear sitting on a green grassy meadow, with its reflection in the water.</p>	<p>This is a black bear walking through a forest.</p>
	<p>The man in the street.</p>	<p>This is a man with a green face and a hat.</p>
	<p>A group of people sitting around a table, enjoying a meal.</p>	<p>A large pizza on a table with a plate of food.</p>
	<p>A man wearing a hat in the street.</p>	<p>A man talking on a cell phone while standing in a crowd.</p>
	<p>The given text contains multiple errors</p>	<p>This is a remote control sits on a quilted couch.</p>

Table 8: Examples from our test set using the transformer-based adapter with self-attention

Image	Pre-trained	Fine-tuned
	<p>A man is sitting on a chair.</p>	<p>A man in a red shirt playing tennis on a red clay court.</p>
	<p>A brown bear standing on its hind legs.</p>	<p>A black bear standing in the woods.</p>
	<p>The man in the image is standing in a field.</p>	<p>A man in a hat and tie is smiling at the camera.</p>
	<p>A white rabbit sits on a green lawn, surrounded by flowers and trees.</p>	<p>A pizza box on a table with a pizza in it.</p>
	<p>The man in the street is walking.</p>	<p>A man is talking on his cell phone.</p>
	<p>The image shows a group of people sitting at a table</p>	<p>A remote control sitting on a couch.</p>

Table 9: Examples from our test set using query transformer-based adapter with learnable queries

Image	Pre-trained	Fine-tuned
	<p>A man in a nearby chair reads a book.</p>	<p>A man is playing tennis on a court with a racket.</p>
	<p>A black cat is sitting on a white mat, licking its paw.</p>	<p>A black bear standing in the middle of a forest.</p>
	<p>A man sits on a white bench in a quiet</p>	<p>A man in a green hat and a suit is standing in a crowd.</p>
	<p>A red boat on a blue lake.</p>	<p>A pizza with a variety of toppings is being cut into slices.</p>
	<p>This image depicts a group of people sitting on a white.</p>	<p>A man in a coat is talking on a cell phone.</p>
	<p>A collection of various chairs in an antique store is displayed in the image.</p>	<p>A remote control sitting on top of a table.</p>

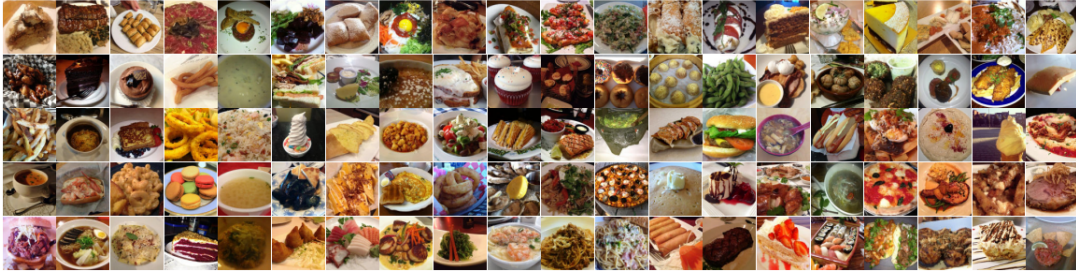


Figure 22: Food101 dataset examples for the 100 out of the 101 classes. Image taken from [3].

To this end, we fine-tuned our adapters and the LLM specifically for this task. During the fine-tuning process, we start with fixed questions in the prompt, such as, “What type of food is that?”. We follow the same fine-tuning approach as previously described, which involves appending the image tokens. We finetuned for two epochs using the same setting as before and then, the LLM is adjusted to respond by providing only the name of the food type as shown in the Figure 23. This fine-tuning process enabled the model to better understand and categorize the diverse range of food items present in the dataset, providing a practical measure of the model’s classification capabilities.

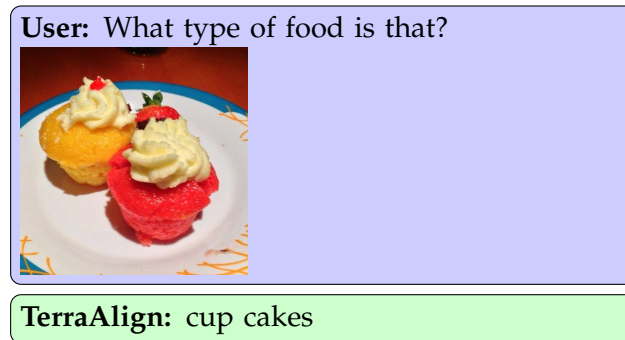


Figure 23: Chat-style question and answer for the Food101.

Table 10: Classification accuracy of different adapters on the Food101 Dataset.

Adapter Type	Classification Accuracy (%)
MLP Adapter	81.52%
Self attention Adapter	83.56%
Query Adapter with learnable queries	82.86%

The results of our TerraAlign framework on the classification task after fine-tuning demonstrate promising performance across different adapter architectures. Specifically, as shown

in Table 10 the self attention adapter achieved the highest classification accuracy at 83.56%, followed by the query transformer-based adapter with 82.56%, and the MLP adapter with 81.52%. This highlights the advantage of leveraging more sophisticated adapter architectures, such as transformer-based designs, which likely capture more complex patterns in the data.

It is important to note that the current evaluation metric was based on checking whether the exact class name appeared in the model’s output, meaning even minor typos or variations were counted as incorrect. In comparison to traditional state-of-the-art methods, such as EfficientNet [32], which achieves an impressive 93.0% accuracy on the Food 101 dataset, our approach shows there is room for improvement. However, the primary goal of this evaluation was to assess TerraAlign’s performance in a classification task and quantify its effectiveness. This provides a useful benchmark for measuring the impact of our pretraining phase and the LoRA fine-tuning method, offering insights into the strengths and areas for further development within our pipeline.

Implementation and Challenges

Through this work, we developed a flexible pipeline capable of aligning image tokens into the text tokens latent space, enabling seamless integration of different pre-trained image encoders and Large Language Models with minimal effort. Our implementation was carried out in two main stages: integration of pre-trained models and the design of an adapter architecture.

In the first stage, we integrated the CoCa model as the image encoder and the instruct-tuned Mistral7b as the LLM into our pipeline. This setup allowed us to leverage powerful pre-trained models for both visual and textual processing. The second stage involved implementing a model component known as an adapter, which maps image embeddings to the text embedding space. We explored three different adapter architectures: a simple multi-layer perceptron (MLP), a transformer-based decoder with self-attention, and a query-based transformer adapter designed to filter the image tokens selectively.

During the pre-training step, our adapters were trained without involving the LLM by minimizing the Earth Mover’s Distance between the image and text embeddings. This approach reduced computational costs and allowed the training to be carried out on standard hardware without requiring specialized systems. In the fine-tuning step, the LLM was introduced into the pipeline, where both the adapter and the LLM were trained simultaneously. We employed LoRA for the LLM, which allowed us to update only 0.047% of the LLM’s parameters. However, this phase still demanded substantial GPU resources; for example, fine-tuning on an NVIDIA A100 required 70 GB of GPU memory, even with a batch size of 16.

Moreover, fine-tuning required extensive hyperparameter tuning to achieve optimal results. For instance, in our transformer-based adapters, small adjustments in the number of heads or blocks had a significant impact on performance, indicating the sensitivity of these models to hyperparameter changes. Similarly, the number of learnable queries in the

third adapter required careful calibration; we settled on 128 queries as a balanced trade-off between accuracy and efficiency.

4 Summary, Discussion and Future Work

4.1 Summary

In section 2, we reviewed the current state-of-the-art, beginning with transformers that form the foundation of LLMs and concluding with multimodal models that integrate pre-trained image encoders and LLMs, such as LLaVa and BLIP-2. In section 3, we proposed our methodology that follows a new way of aligning the image embeddings into the text embedding space by minimizing the EMD without relying on LLMs. We evaluated our method by generating captions and in a classification task in order to quantify the evaluation.

4.2 Discussion

The thesis studied the VLMs using pre-trained backbones and focused on reducing the computational resources that are required to build such models in a multi-modal setting and benefit from the already trained Large Language models in computer vision tasks. The methods that were proposed are focused on using an adapter to transform the image embeddings into the text embeddings space so as to make the LLM to understand the content of the image. In order to tackle this challenge, two different stages of training are required, while in the first the LLM is completely omitted from the training setup.

In the pre-training step, the adapter is trained to minimize the Earth Mover's Distance, meaning that it learns to map the image embeddings distribution to the text embedding distribution. For the fine-tuning step, the approach mirrors state-of-the-art methods by minimizing the Cross-Entropy loss on a token level autoregressively. In this work, three different adapters are proposed

1. The first adapter is a simple MLP, is chosen for its simplicity and as a robust baseline for further experimentation. This adapter demonstrated the fundamental capability to map visual information into a format that LLMs can interpret.
2. The second adapter is a transformer-based model, that adds self-attention layers to capture correlations between the image tokens and makes it a more accurate and sophisticated solution compared to the simple MLP.
3. The third adapter is a query transformer-based model, leveraging the strengths of attention mechanisms to more effectively capture complex relationships within the data. This adapter employs learnable queries to focus on critical aspects of the visual data, enabling the model to adapt dynamically to different types of visual inputs. This adapter shows that can be more accurate because the queries learned to focus on the rich information image tokens but also more efficient.

The MSCOCO dataset was selected for our experiments due to its extensive collection of images paired with natural language descriptions, covering a wide range of real-world en-

vironments, making it ideal for training visual-language models. In addition, the Food101 dataset was selected for quantifying the performance of these VLMs on a classification task.

The most important results from the experiments were as follows:

1. The pre-trained adapter can be used for visual-language tasks, particularly for simpler cases, however the fine-tuning step is essential to provide more accurate and well-structured answers based on visual input.
2. By omitting the LLM in the first stage of training, makes it more feasible for individuals and low-budget institutions to experiment with the multi-modal world.
3. Ensuring alignment between image embeddings and text embeddings in the pre-training step is crucial for making the fine-tuning stage more efficient, resulting in better convergence and overall performance.
4. The MLP adapter provides a simple yet effective baseline, while the transformer adapters offers more advanced capabilities by focusing on essential features through attention mechanisms. This progression from basic to complex adapters highlights the trade-offs between simplicity and performance, allowing users to choose the best approach for their specific needs and computational resources.

The results of our research on TerraAlign provide several key insights into the development and optimization of Visual Language Models. One of the most significant contributions of our approach is the improvement in computational efficiency achieved by omitting the Large Language Model during the pre-training phase. By focusing on directly aligning visual and text embeddings using Earth Mover’s Distance, we were able to streamline the process without compromising the quality of the alignment. The EMD’s capability to effectively bridge the semantic gap between these two modalities proves that a LLM is not always necessary in the early stages of training, which significantly reduces the computational overhead.

Another critical finding is the importance of the pre-training step in the overall training pipeline. Our experiments demonstrate that this phase not only prepares the model for better alignment but also leads to faster convergence during the fine-tuning step. By pre-aligning the visual and text embeddings before introducing the LLM, we observed that the fine-tuning process not only converges more rapidly but often reaches a better minima compared to directly training the model with the LLM unfreezed from the start. This highlights the role of the pre-training phase in setting a strong foundation for the integration of the LLM.

However, our exploration of different adapter architectures revealed a trade-off between efficiency and model complexity. While simpler adapters like the MLP are less computationally demanding, they may not fully capture the intricate relationships between visual and textual features. Conversely, more complex architectures, such as the transformer-based adapters, provide better alignment at the cost of increased computational resources.

This trade-off is crucial for real-world applications, where the choice of adapter must balance performance with resource efficiency.

The fine-tuning step, despite the efficiency gains from the pre-training phase, remains essential for achieving accurate results. During this phase, the LLM is introduced and fine-tuned alongside the adapter, allowing the model to refine the alignment and enhance its performance on specific visual-language tasks. The fine-tuning process not only improves the model's accuracy but also ensures that it can generalize effectively across different tasks and datasets.

Lastly, while the MSCOCO and the Food101 datasets served as a valuable resource for training and validating TerraAlign, it is important to acknowledge that this and similar datasets may not fully capture the diversity of visual-language tasks that the model might encounter in real-world applications.

Overall, the methods and results presented in this thesis contribute to the ongoing development of efficient and accessible VLMs by leveraging existing language models and reducing the computational burden typically associated with such complex models. By systematically exploring different adapter architectures and training strategies, this research lays the groundwork for future advancements in the field, promoting the integration of vision and language processing in a cost-effective manner. The TerraAlign framework demonstrates the feasibility of using pre-trained models and advanced adapter architectures to achieve effective visual-language alignment. While our approach has shown promise, it also opens up several avenues for further research.

4.3 Future Work

There are several avenues for future research and development to further improve TerraAlign:

1. **Enhanced Vision Encoders:** Explore using more advanced vision encoders. Variations of CLIP with higher embedding dimensions could capture more detailed information, potentially improving the alignment and performance of the model.
2. **Advanced Adapter Architectures:** Investigate different and more sophisticated architectures for adapters. However, it is crucial to balance the trade-off between accuracy and computational efficiency.
3. **Improved Large Language Models:** Experiment with more advanced LLMs to determine their impact on the alignment and overall performance of TerraAlign. Newer models may offer better language understanding and generation capabilities. For instance, bigger and newer models may be able to perform much better even with utilizing only the pre-training step.
4. **Expanded Dataset:** Future work could involve training the pipeline on a more diverse dataset, similar to the one proposed by LLaVa, which goes beyond caption

generation to incorporate multiple vision-language tasks, such as visual question answering (VQA). A dataset that includes varied scenarios would better capture the complexity of real-world applications and help improve the robustness of the model.

5. Current evaluation metrics, such as BLEU [33], are often insufficient for capturing the nuances of visual-language tasks. Future work could involve evaluating the model's responses using another LLM as a judge, providing a more nuanced and contextually aware assessment of the generated descriptions.
6. TeraAlign for Classification Tasks:: Another promising direction is adapting the VLM for specific classification tasks. This could be achieved by replacing the final linear layer of the LLM with one that projects the desired number of output classes. The pipeline could then be fine-tuned on the classification task, either by training only the last transformer block and the new linear layer or by fine-tuning the entire model. Such an approach could tailor the VLM to a classification task, potentially yielding higher accuracy in domain-specific tasks.
7. Extended Applications: Extend the current pipeline to additional tasks beyond image captioning, such as object detection. Moreover, applying the approach to different modalities, like speech and language, to create more versatile multi-modal models.

Bibliography

- [1] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.
- [2] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.
- [3] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 446–461, Cham, 2014. Springer International Publishing.
- [4] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.
- [5] Harry A. Pierson and Michael S. Gashler. Deep learning in robotics: a review of recent research. *Advanced Robotics*, 31:821 – 835, 2017.
- [6] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012.
- [7] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.
- [8] Florian Bordes, Richard Yuanzhe Pang, Anurag Ajay, Alexander C Li, Adrien Bardes, Suzanne Petryk, Oscar Mañas, Zhiqiu Lin, Anas Mahmoud, Bargav Jayaraman, et al. An introduction to vision-language modeling. *arXiv preprint arXiv:2405.17247*, 2024.
- [9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett,

editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

- [10] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeef Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O’Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ry-

- der, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024.
- [11] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shrutu Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.
- [12] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L elio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth ee Lacroix, and William El Sayed. Mistral 7b, 2023.
- [13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words:

- Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.
- [15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*, 2019.
- [16] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9992–10002, 2021.
- [17] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 548–558, 2021.
- [18] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.
- [19] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *Transactions on Machine Learning Research*, 2022.
- [20] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 34892–34916. Curran Associates, Inc., 2023.
- [21] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.
- [22] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. See <https://vicuna.lmsys.org> (accessed 14 April 2023), 2(3):6, 2023.

-
- [23] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 19730–19742. PMLR, 23–29 Jul 2023.
- [24] Yuxin Fang, Wen Wang, Binhui Xie, Quan-Sen Sun, Ledell Yu Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19358–19369, 2022.
- [25] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.
- [26] Anqi Mao, Mehryar Mohri, and Yutao Zhong. Cross-entropy loss functions: Theoretical analysis and applications. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 23803–23828. PMLR, 23–29 Jul 2023.
- [27] The Earth Mover’s Distance as a Metric for Image Retrieval - International Journal of Computer Vision — link.springer.com. <https://link.springer.com/article/10.1023/A:1026543900054>.
- [28] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv: Learning*, 2016.
- [29] Abien Fred Agarap. Deep learning using rectified linear units (relu). *ArXiv*, abs/1803.08375, 2018.
- [30] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *ArXiv*, abs/1505.00853, 2015.
- [31] Andrzej Maćkiewicz and Waldemar Ratajczak. Principal components analysis (pca). *Computers Geosciences*, 19(3):303–342, 1993.
- [32] Mingxing Tan. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*, 2019.
- [33] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In Pierre Isabelle, Eugene Charniak, and Dekang Lin, editors, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.