MSc Artificial Intelligence
Master Thesis

Fine-grained image understanding with VLMs

by
Anesa Ibrahimi
15128350

August 13, 2025

36 EC
Jan 2025 - Jul 2025

Supervisors:
PhD Ivona Najdenkoska
PhD Mohammad Mahdi Derakhshani

Examiner:
Dr Marcel Worring

Second reader:
PhD Ivona Najdenkoska

Universiteit van Amsterdam

Contents

Abstract

Vision-Language Models (VLM) have gained impressive generalization abilities, learning to identify a vast range of concepts from web-scale data without direct supervision. A key limitation, however, is their difficulty with fine-grained image understanding, often failing to capture the intricate details that define complex scenes. To address this shortcoming, we propose a straightforward and efficient method for augmenting frozen foundation models with a persistent memory mechanism. By strategically replacing Multi-Layer Perceptron (MLP) sub-layers in a Vision Transformer with trainable, key-value memory modules, we enhance the model's architectural capacity for detailed feature storage. A teacher-student knowledge distillation framework is then employed to efficiently transfer knowledge from a pre-trained CLIP model into our memory-enhanced student, eliminating the need for costly retraining from scratch. Our results demonstrate that a memory-augmented vision encoder can be effectively trained to achieve a new level of performance on long-caption fine-grained retrieval benchmarks. Moreover, they highlight an important trade-off between specialization and generalization. Enhancing fine-grained retrieval capabilities through this architectural modification can impact performance on pixel-level tasks like zero-shot semantic segmentation. These insights improve our understanding of how architectural changes affect pre-trained VLM and provide a foundation for future advancements in developing more comprehensive and efficient models for fine-grained image understanding.

Chapter 1

Introduction

The soul never thinks without an image

Aristotle, De Anima III

The emergence of Vision-Language Models (VLM) enabled machines to view the visual world through the lens of human language. These models, embodied by architectures like CLIP [28], have fundamentally transformed multi-modal understanding by seamlessly aligning images and text within a shared embedding space. This paradigm has enabled a significant shift from inferring concepts from limited examples to directly utilizing human language for visual understanding. These foundational models, however, exhibit a critical limitation in understanding fine-grained details while showcasing proficiency in recognizing general concepts. The core training objective of CLIP encourages a global alignment that pushes towards matching an entire image to a concise caption. Consequently, this approach often fails to capture the intricate, subtle attributes and complex interactions that define a rich scenery. As highlighted by recent research, this results in "blind spots" where models struggle to differentiate between visually distinct but conceptually similar images, a deficiency that descends into more complex multi-modal systems [33]. Furthermore, architectural constraints, such as the 77-token limit of CLIP's text encoder, prevent these models from effectively leveraging the long, descriptive captions necessary for true fine-grained evaluation [34]. While existing work aimed at addressing these issues through improved data curation [39], specialized pooling mechanisms [38], or by extending the text encoder's context length [44], these approaches often rely on fragile external captioning pipelines or introduce significant computational overhead. Critically, lacking any inherent mechanism to persistently store and recall the localized features they extract across multiple inferences. This thesis proposes a novel framework centered on a **memory-augmented Vision Transformer**. Our core hypothesis is that by strategically integrating trainable, persistent key-value memory slots into the ViT architecture, we can provide the model with a dedicated mechanism to store and recall localized, fine-grained features across multiple inferences. Moreover, we pair our method with a targeted distillation scheme to efficiently inherit the global alignment capabilities of the CLIP model while focusing on enhancing the vision tower for long and descriptive text grounding. Hence, this approach aims at enabling the vision encoder to store and retrieve fine-grained details without sacrificing the general-purpose representations learned during large-scale pre-training. This method is evaluated across short and long caption retrieval tasks as well as semantic segmentation. Ultimately, enabling a holistic view of our model's strengths and trade-offs. To guide this investigation we formulate the following primary research question:

To what extent can augmenting a pre-trained Vision Transformer with trainable memory layers improve its fine-grained image understanding?

This central question is decomposed into three specific sub-questions that address the core challenges of architectural design, efficient training, and empirical performance:

1. **Architectural Design:** How can trainable memory modules be effectively integrated into the architecture of a standard ViT to enhance its capacity for storing fine-grained visual features without disrupting its foundational structure?

2. **Knowledge Transfer:** How can the rich, general-purpose knowledge from a large-scale pre-trained CLIP model be efficiently transferred to a memory-augmented student model, avoiding the computational cost of retraining from scratch while preserving its core alignment capabilities?

3. **Empirical Performance and Trade-offs:** How does the model perform on a set of downstream tasks compared to the standard CLIP baseline and other SOTA models ?

Our key findings, at a glance, reveal our model's strength in text-to-image retrieval on standard and fine-grained settings, indicating discriminative visual features when ranking candidate images. On long-caption benchmarks, it managed to decisively outperform the CLIP baseline model on various datasets like DCI[34] and Urban-1k[44] with large margins while also remaining relatively competitive with several SOTA approaches. These gains are accompanied by trade-offs, nonetheless, suggesting an affinity towards using rich textual descriptions to retrieve images rather than ranking captions given an image. Our semantic segmentation experiments highlight the challenges faced by our model in preserving coarse and general scene structure. On the other hand, the ablation studies conducted in this work offer design insights with respect to the integrated memory layers of our method. The obtained results in the cross-modal retrieval task pinpoint an optimal configuration, revealing that our method benefits most from late and sparse memory layer infusion in upper transformer blocks while prioritizing high-capacity memory slots that can store and retrieve semantically rich features.

The primary contributions of this work are:

1. **A Novel Memory-Augmented Vision Encoder:** We design and implement a new VLM architecture that replaces standard MLP sub-layers in a ViT with trainable Memory modules, enhancing the model's capacity for fine-grained feature storage.

2. **An Efficient Knowledge Transfer Framework:** We develop a two-stage training pipeline leveraging knowledge distillation and targeted fine-tuning. Efficiently transferring knowledge from a pre-trained CLIP model into our memory-augmented student and avoiding the need for costly retraining from scratch while preserving foundational alignment.

3. **Comprehensive Empirical Analysis:** We conduct an extensive evaluation across ten datasets for cross-modal retrieval and semantic segmentation. Our results validate the efficacy of our approach on long-caption, fine-grained tasks.

1.1 Outline

Chapter 2 reviews the foundational literature on Transformers and CLIP and situates our work within the landscape of recent advancements in fine-grained and long-context models. **Chapter 3** presents our methodology in detail. **Chapter 4** outlines the experimental setup, including implementation details, datasets, evaluation metrics, and the training pipeline. **Chapter 5** presents and analyzes the quantitative and qualitative results of our experiments. Lastly, **Chapter 6** summarizes our findings, discusses their implications, and suggests directions for future work.

Chapter 2

Literature Review

The following section will begin by reviewing the Transformer backbone and CLIP's pioneering image–text pretraining (Section 2.1), then examine the core limitations that still penalize original CLIP (Section 2.2). Next, a spectrum of CLIP variants that push scale, context length, and semantic depth (Section 2.3) is covered, before zooming in on fine-grained VLMs and their own deficiencies (Section 2.3.1).

2.1 Foundational Models and Architectures

Transformers have impacted the field of natural language processing (NLP) and vision language models (VLM) with their attention-based architecture, excelling in tasks requiring long-term memory and complex reasoning [35]. At its core, a Transformer takes a sequence of token embeddings, which are then augmented with positional information, and passed through a deep stack of identical layers. These layers, comprised of two main cooperative blocks, enable every token to attend every other or refine the attended representations with residual connections and layer normalization. The former is also known as the multi-head self-attention block, whereas the latter is the feed-forward network block. On the encoder side, every layer contains exactly those two sub-modules—(1) multi-head self-attention and (2) a position-wise fully connected feed-forward network. Therefore, the encoder can be viewed as N identical copies of the same two-step processing unit. The decoder re-uses the same pair and inserts a third sub-module between them: a multi-head encoder–decoder attention layer that allows each target-side token to query the entire set of encoder outputs. To keep the generation auto-regressive, the decoder's self-attention is masked so that each position can only look left (i.e., at earlier tokens), and the target embeddings are shifted one place to the right before entering the stack; together, the mask and the shift guarantee that token i is conditioned only on tokens 1. . . i-1. When diving deeper into Transformer's contents and structure, the attention mechanism can certainly be considered one of the most prominent features for which its ability to capture long-range dependencies by dynamically focusing on the most relevant tokens across an entire sequence can be attributed. Because every self-attention head computes weighted combinations over the full sequence (subject to the decoder mask), the model learns global context without recurrence, enabling it to model complex linguistic or cross-modal relationships at scale.

Natural language processing (NLP) first embraced this paradigm shift through the Transformer architecture, whose self-attention mechanism substituted recurrent biases and enabled efficient global context modeling across thousands of tokens. Since then, the architecture has been extended and refined upon in several domains and applications, notably one of them being the computer vision field. Researchers began to treat an image as a series of patches, allowing multi-modal models to swap "words" and "pixels" like equivalent tokens within a shared embedding space. The transformer's architecture can therefore be viewed as a key step

toward bridging computer vision and natural language processing through its ability to learn associations between words and visual cues through multi-head self-attention mechanisms.

Several training paradigms translate this unified embedding-space approach into practical vision-language models [3]. Among them: masking-based objectives, pre-trained backbone alignment, and fully generative models; one eminent example of a VLM leveraging Transformer architecture is **CLIP** (Contrastive Language-Image Pretraining). It aligns images and text through a *shared* embedding space using *contrastive* learning, allowing robust performance across zero-shot and fine-tuned tasks [28]. As the authors note [28], CLIP's core idea is to "learn perception from supervision contained in natural language", reason being its ability to scale effortlessly with web-sourced text–image pairs, to ground visual features directly in human language, to exploit the vast abundance of publicly available data of natural language form. Earlier efforts aimed at training image convolutional neural networks (CNN) with text transformers in order to predict a caption of an image. Consequently, difficulties in scaling this method emerged when trying to predict exact words of text accompanying each image. Therefore, contrastive objectives soon become more favorable and advantageous than predictive objectives and explored a training system that predicted text as a *whole* instead of words of that text during pairing [28]. To this end, CLIP learns a multi-modal embedding space by *jointly* training an image-text encoder to maximize the cosine similarity of their respective embeddings while minimizing incorrect pairs. To further inspect and uncover CLIP's strengths an extensive study by [30, 28] set out to systematically evaluate how much a frozen CLIP encoder can improve a wide range of vision-and-language benchmarks, comparing zero-shot, few-shot, and fine-tuned settings across classification, retrieval, visual question answering, and caption grounding. Their findings revealed the performance matching between best performing CLIP model with original ResNet-50 [16] model on the ImageNet [10] without the vast & large dataset that was used during ResNet-50 model training. Moreover, further evaluations by [28] revealed that the CLIP model matched the accuracy of a 16-shot classifier trained on strong visual features. This behavior of the CLIP model can largely be attributed to CLIP's natural language prompt classifier, where visual concepts are directly specified through textual description in comparison to supervised learning, where it must infer concepts from one or a handful of example images, where many different interpretations of the data are equally plausible.

To extend this investigation of CLIP's superiority, the research of [14] studied the robustness of CLIP to various natural distribution shifts. As mentioned by the authors of this research, contrary to prior models which are trained with images-class annotations, CLIP models are directly trained on images and their unstructured text retrieved from the web. This characteristic was deemed as a contributing factor to its robustness on challenging distribution shifts. Therefore, throughout their systematic experimentation phase they studied various possible causes of robustness (training set size, training distribution, language supervision at training/test time, contrastive loss function). From the above-mentioned set of candidates, [14] show that CLIP's robustness is dominated and determined mostly by the training distribution ("the more diverse the training distribution, not the language supervision, the more robust the representations"). Thereby, revealing that CLIP's advantages are not necessarily constrained to its architecture but emerge from coupling it with a broad, diverse training distribution. Complementary evidence by [39] which aimed at uncovering CLIP's data curation process , due to limited data information from [28], also confirmed that high-quality and diverse image-text pairs drive CLIP's robustness and transfer performance. In more detail, their findings showcase the pure NLP-based filtering approach of CLIP, which relies on a simple caption-metadata string-matching filter followed by a balancing step that flattens the long tail of concepts and keeps noise low. In return, it yields a uniform diverse corpus suitable for pre-training that serves as a stronger foundation for pre-training.

2.2 Identifying CLIP's Core Limitations

While CLIP excels at broad generalization, recent evidence suggests it is far from infallible. Tong et al. [33] demonstrate that seemingly obvious visual differences can collapse into near-identical embeddings. A shortcoming, they expose through their *MMVP* (Multi-modal Visual Patterns) benchmark. This benchmark aimed at exposing visual shortcomings in MLLMs (Multi-modal LLMs), which the authors [33] claimed to be related to visual representations. Recent advancements of various MLLMs showed strong capabilities in various downstream tasks such as image understanding and visual question answering. However, most of open-source MLLMs are built on pre-trained vision & language models for which Tong et al. [33] claim that a limitation in the pre-trained models is said to cascade into the MLLMs that utilize them. Consequently, becoming a bottleneck in multi-model systems. As highlighted in [33], the majority of MLLMS made use of *CLIP* vision encoders to perform image processing. Through extensive experiments aiming at identifying failure examples of CLIP model, the authors were able to identify so-called 'CLIP-blind pairs' which led to errors in MLLMs that adopted CLIP vision encoders. The culprit of errors in MLLMS and proper encoding of CLIP, blind-pairs, occur when two visually different images are encoded similarly. This erroneous agreement in the vision embedding space led the authors [33] in finding 9 visual patterns misinterpreted from CLIP-vision encoders:

- Orientation & Direction
- Presence of Specific Features
- State & Condition
- Quantity & Count
- Positional & Relational Context
- Color & Appearance
- Structural & Physical Characteristics
- Text
- Viewpoint & Perspective

When evaluating various CLIP models on their benchmarks (MMVP, MMVP-VLM), which consisted of abundant cases of blind-pairs and human-annotated questions related to the visual details found inside the pairs (images), the authors [33] highlight the consistent challenges that remain in CLIP-based models despite the diverse architecture, training data volumes, and model sizes. Across five CLIP variants , scaling up model size and data led to measurable gains only on 2 ("color & appearance", "state & condition") patterns; the other seven patterns remain challenging. The study uncovers a strong correlation between vision and language understanding: multimodal LLMs that build directly on CLIP encoders (e.g. LLaVA 1.5 [24] and InstructBLIP [9]) inherit the same blind spots. This alignment of failures suggests that, even as CLIP-style models remain one of the most scalable vision backbones, their visual pattern deficiencies become a bottleneck for multi-modal systems.

Switching gears to fine-grained alignment, Urbanek et.al [34] introduce the DCI (Densely Captioned Images) dataset. Consisting of 7805 natural images paired with human-annotated mask-aligned descriptions, it is used as an effective tool to evaluate the image-understanding capabilities of VLMs via a novel approach involving the careful alignment of each image caption with a corresponding image sub-crop. The authors have taken inspiration and initiatives to build this dataset from insights of previous works that observed the crucial role of caption quality and density with respect to the efficacy of VLMs. To this extent, the authors have highlighted the reliance of VLMs (such as CLIP) on training datasets with loosely labeled and short captions. Underscoring the inabilities of VLMs in learning deep alignments and capturing rich visual details for fine-grained image understanding tasks. Therefore, [34] have constructed the DCI dataset which specifically aims at offering image descriptions averaging *over* 1000 words per image ; amounting to 100x more dense than other benchmarks such as COCO [6]. A major practical drawback of current CLIP models is its **77** text-token limit which consequently prevent it from being able to utilize the rich and dense captions of DCI dataset. A workaround to this

drawback, sDCI (a summarized version of DCI) was constructed by the authors [34] to enable CLIP models to process each sub-crop caption. They caution, however, that this summarization necessarily enforces only a lower bound on DCI's true evaluative resolution. Furthermore, the authors then constructed a Subcrop-Caption Matching (SCM) evaluation task which aims at benchmarking VLM's fine-grained ability by challenging them to correctly distinguish different parts (sub-crops) of an image via caption-matching. On this crucial benchmark, [34] shows the poor performance of CLIP baselines and infers that no existing model can perform well at matching image (general) captions with sub-crops of that image. This suggests a limitation in how CLIP-style models generalize their understanding across different types of fine-grained tasks. The DCI dataset and sDCI benchmarks were designed to precisely expose these limitations and reveal CLIP's reliance on large-scale, loosely-labeled data and their architectural constraints regarding text length. For that reason, limiting its ability to truly understand and process intricate relationships between visual and linguistic elements at a dense, fine-grained level.

Pushing CLIP's flaws even further into the spotlight, [15] has exposed drawbacks of current VLMs through its unique dataset IIW(Image in Words) and annotations guidelines. As mentioned by the authors of this work alongside the work of [34] for the DCI dataset, VLMs that are trained on large and noisy web datasets which rely on alt-text that is scraped from the internet serve as incomplete and imperfect image captions. Datasets that rely on human-annotated (DCI[34], DOCCI[27]) or model-generated (PixLore [2], DAC[12]) captions have quickly substituted the noisy web datasets. However, [15] noted that both of the substitutions above have limitations, with human-generated captions varying heavily on human attention, effort, and bias whereas model-generated ones are incomplete and containing hallucinations. The IIW dataset on the other hand combines the quality of human annotators and seeded metadata from machine generation, which was then used to accurately and thoroughly offer insight into the present gaps of current VLMs. The 9018 hyper-detailed images of the IIW dataset alongside its guidelines, further emphasize the flaw of web-scraped data by not only specifying their short-length captions but also its poor and irrelevant content. Moreover, the IIW annotation guidelines have explicitly attended to concepts beyond objects (visual perspective, spatial arrangement, human-object interactions) which further aided in detecting the drawbacks in existing VLMs that struggled to compare these more complex aspects of images. Further suggesting the areas where current models fall short without richer training data [15, 33]. Lastly, both works [34, 15] have discussed the model exploitation of 'language priors'. In other words, a good text encoder is able to achieve high accuracy without looking at images. [15] further quantified and tested this behavior by introducing a 'no-image condition' baseline for existing benchmarks (ARO [42], Winoground [32] etc.) which was then given to an LLM to correctly pick the image caption. On certain subsets of these benchmarks, the LLM was able to perform significantly above random chance purely based on language priors (inherent understanding of word structure and meaning). Essentially, revealing that certain benchmarks might not be effectively testing the VLM's vision-language reasoning capabilities, as well as calling into question the true "understanding" gained by models that show progress on such benchmarks.

2.3 Extending CLIP family

Recent research has enabled a clear thrust in CLIP-style models to unfold along major axes such as sub-caption alignment, dismantling the 77-token ceiling to embrace long contexts, and LLM infusion for richer semantics. All with a shared goal of drifting away from CLIP's original limitations and instead moving toward fine-grained, long-context, and language-aware vision models.

DreamLIP [45] has delved deeper in understanding the role of captions in VLMs. More particularly, whether language-image pre-training can benefit from long captions. Through a pre-trained Multi-modal Language Model (MLLM), the authors re-caption 30M images with detailed descriptions. The captioner, a pre-trained MLLM, is used to generate a collection of long and short captions of images . Its purpose is to dynamically sample sub-captions from the text label to construct multiple positive pairs which are then used inside a multi-positive loss framework to sophisticatedly interweave sub-captions with their corresponding image. Moreover, a subcaption-specific grouping loss was defined to ultimately enable DreamLIP's fine-grained alignment between local image patches and sub-captions. The zero-shot image-text retrieval of this model in various datasets such as Flickr30k [41] and MSCOCO [6] showcases its superiority over the original CLIP model. However, the drawbacks of MLLMs pose a limitation to the capabilities of this model. As previously mentioned in the work of [15], model-generated captions may face hallucinations. Zheng et al. [45] has further stressed the severity of hallucinations in longer captions for which it remains a barrier to model's fine-grained image–text alignment efforts.

Laying the groundwork for more expressive, hallucination-resistant models, research like TULIP, HoPE, and LoTLIP each introduce novel positional-encoding or token-management strategies to enable lengthy captions. LoTLIP [37], in particular, incorporated corner tokens to gather diverse textual information. Extensive evaluation by [37] has confirmed that long text-image pairs aid the pre-trained model in understanding long text yet it negatively impacts performance on short text-image retrieval. First, [37] incorporated long captions during the pre-training stage through the careful re-captioning of 100M images with long texts via three MLLMs. To this extent, corner tokens were proposed to collect diverse text features. When a long text is tokenized, multiple learnable corner tokens are inserted after the initial class token. These tokens are designed to aggregate diverse text features, benefiting the class token by helping it extract more representative features for both long and short text. To promote the diversity of extracted features, an attention mask mechanism was designed to guide class and corner tokens to efficiently attend sub-caption tokens within the long text. Ensuring that they gather comprehensive information from the entire input.

Extending the idea of expanding caption capacity, TULIP [26] goes further than token aggregation. Replacing CLIP's *fixed* absolute positional encodings with *relative* ones (RoPE), effectively lifting the 77-token limit. More specifically, RoPE (Rotary Positional Encodings) rotate the embeddings based on the relative distance between tokens, which injects positional information into query and key vectors within the self-attention layers of the Transformer architecture (backbone of CLIP model). Furthermore, the authors propose a distillation approach, applicable to CLIP-like models, which eliminate the need to retrain the model from scratch. After distillation, the model is further enhanced to process captions beyond the original 77-token limit through the use of Neural Tangent Kernel (NTK)-aware scaled RoPE. This in return, enables the model's adaptation abilities with respect to the varying length of the input through the scaling of the rotational frequency. Thus, leading to more accurate and contextually rich image generation, even capturing nuanced details that appear beyond the token limit.

An alternative to rotary positional encodings, HoPE [7] proposes replacements in specific components of RoPE to enhance model context awareness. The authors claim that RoPE was designed with the assumption that tokens farther away from the current position carry less relevant information, leading to a principle of long-term decay. HoPE on the other hand replaces specific components in RoPE with position-independent ones. Reason being, the author's [7] observation on the shape of attention scores formed by RoPE. RoPE was designed so that words closer together in a sentence would naturally pay more attention to each other, and those farther apart would "fade". However, its attention scores formed a U-shape which show strong attention at the very beginning and at the end of the sequence. Implying that RoPE

overvalues the ends of a sentence and almost ignores all the middle part. HoPE's fix involves the stripping of troublesome frequency parts and replacing them with simpler, position-independent ones in order to prevent the model being biased toward the ends or the start. By keeping only the high-frequency bits that genuinely help signal relative positions, HoPE restores a more balanced and reliable sense of "where" each token sits, enabling better context awareness even in very long texts.

2.3.1 Fine-grained VLMs

FineCLIP lays bare CLIP's limitations and fills them through its comprehensive representation space that aligns visual & semantic cues both broadly and in detail . As highlighted in [18], CLIP models fail to capture fine-grained alignment between image regions and their corresponding textual attributes. Partly, due to CLIP's alignment process which matches an image as a *whole* to the text description as well as it loss function which ignores the supervision of visual and textual dense features. Consequently, lacking the ability to understand fine-grained details. The appealing features of FineCLIP (regional contrastive learning paradigm, real-time self-distillation scheme) equip the model with extra 'views' of each region allowing it to peak into local patches while keeping the whole scene in view. The former, regional contrastive learning paradigm, leverages advanced Large Vision-Language Models (LVLMs) to generate high-quality and detailed region descriptions. The latter, real-time self-distillation scheme, transfers robust global representational capabilities to region features. Unlike previous approaches that relied on frozen teacher models (which limited performance and scalability), FineCLIP effectively allows the model to teach itself independently. Together, these components cooperate to create a comprehensive representation space where visual and semantic features are aligned at both global and local levels. The global contrastive learning preserves semantic consistency and strong representational capabilities, while the self-distillation mechanism transfers this knowledge to local features, and the regional contrastive learning enriches the model with fine-grained understanding.

Similar to FineCLIP, [19] introduced COSMOS model to create global *and* local views of images and texts. For images, it adopts a standard multi-crop strategy to generate global and local views. For text, it introduces a unique text-cropping strategy that randomly samples sentences of varying lengths from long synthetic caption datasets generated by MLLMs. This strategy promotes local-to-global text representation learning. Integrated into the student model, the cross-attention module allows image patch tokens to be conditioned on texts, and word tokens to be conditioned on images. Effectively localizing relevant information in both images and captions, leading to fine-grained embeddings. Contrary to FineCLIP's self-distillation scheme, COSMOS focuses on cross-modal instead of uni-modal alignment. Meaning that it incorporates student's model information from both image and text via cross-attention with teacher's global image & text embeddings.

Lastly, the FLAIR model [38] introduces a text-conditioned attention pooling to achieve fine-grained image understanding. Instead of indirect alignment through a global loss function, FLAIR generates image embeddings by conditioning them on relevant text embeddings. Initially, through the assistance of MLLMs, it samples diverse sub-captions which focus on different local and global regions. The text-conditioned attention pooling feature then constructs unique image representations for every image-text pair. Essentially, contextualizing the image representations with individual captions. Resulting in more targeted image embeddings which are useful for various retrieval tasks. Moreover, it adopts 2 main Sigmoid loss functions (Text-conditioned, Multi-positive) where the former adopts a contrastive loss based on SigLIP [43] while the latter aligns the global image embedding with the text embedding of every sub-caption in order to handle multiple positive captions per image more naturally. Ultimately,

FLAIR is able to learn a fine-grained alignment between image-text at the *token* level.

While models like FineCLIP, FLAIR, and COSMOS promise intricate image understanding, a closer look reveals distinct vulnerabilities and efficiency barriers. As explained above, FineCLIP [18] relies on an automated process to generate detailed regional descriptions of an image. To grasp fine-grained details, the authors reasoned that data should be shifted to a region-level where region-text pairs should be readily available. However, the majority of large-scale data and the ones typically used in CLIP models do not contain and fulfill this strict region-level data requirement of FineCLIP [28, 18]. Hence, the automated pipeline was put in effect to produce the necessary text description of image regions. This reliance on generated data means the quality of FineCLIP's learning is inherently tied to the capabilities of the tools used in this pipeline. Although FineCLIP leverages advanced LVLMs like BLIP-2 [21] for generating region descriptions, which were shown to be effective and superior to rule-based methods, the authors were unable to use the most powerful LVLMs available due to computational constraints. Suggesting that the quality of the generated region-text annotations, and consequently the richness and precision of the fine-grained knowledge acquired by FineCLIP, may not be maximized if computational constraints arise. Moreover, the current methods which FineCLIP used for proposing regions (RPN [29], YOLOv9 [36] etc.) struggle to balance 2 key aspects: category richness (identifying a wide variety of object types) and accurate segmentation.

The automation and reliance on external data generation pipelines has also been utilized by COSMOS [19], in efforts to aid its text capture strategy. Depending on the capabilities and biases of these external MLLMs used for caption generation, [19] note that COSMOS's learning can be constrained if generated synthetic captions are of lower quality and less varied. While not too restrictive, the fine-grained self-distillation scheme of the model requires 10-20% more training and GPU memory than the CLIP model [28]. This means that deploying and training COSMOS, especially at larger scales or with more local crops, demands higher computational resources than simpler CLIP implementations. Conversely, FLAIR [38], trained on a relatively small sample size (30M), managed to outperform larger models on various downstream tasks when using same training sample of FLAIR. Its text-conditioned attention pooling, while effective, causes substantial extra computation (multiple caption encodings per image) and GPU memory use. Attention-pooling approaches like FLAIR bind fine-grained signals only in the context of a single forward pass. Thereby, lacking a retrieval mechanism to persist and recall those localized features across inferences. Additionally, an apparent performance gap becomes visible when compared to larger-scale models (OpenCLIP 2B [8], MetaCLIP 2B [40]) on zero-shot classification tasks. Demonstrating that, while FLAIR excels at understanding specific details within an image, the range of visual concepts and general knowledge required for global image classification still benefits immensely from large and diverse image datasets. In summary, the conducted experiments [38] show that gains in fine-grained retrieval do not necessarily translate to global classification performance, and vice versa.

Fine-grained VLMs like FineCLIP, COSMOS, and FLAIR showcase impressive gains on sub-region tasks, yet remain bound to fragile external caption pipelines, experience significant compute and memory overhead, and lack any mechanism to persistently store and recall the very local features they extract. These limitations, alongside their sensitivity to caption quality and constrained generalization, highlight a clear need for a framework that can efficiently capture fine-grained detail and retain it across inferences. In the next section, we offer a glimpse of our solution: a unified architecture with built-in, long-term feature storage and a tailored distillation strategy, designed to close this gap with scalable, efficient fine-grained image understanding.

Chapter 3

Methodology

3.1 Overview

We seek a way to provide visual backbones with the ability to remember and act upon fine-grained details across multiple inferences, without leaning on fragile caption pipelines or massive compute budgets. To this end, our methodology consists of two key components:

1. Memory Layers: set of *trainable* key–value slots that live alongside the ViT's patch tokens and can store fine-grained features persistently
2. Knowledge Distillation: a teacher–student regime that transfers the memory-powered model's capabilities into a leaner student for efficient inference.

In Section 3.2, a formalized definition and notation of our problem is provided. Section 3.3, will then lay out the Memory-Augmented Vision Transformer, beginning from the background and foundational work of memory layers, the ViT backbone, and then describing how keys and values are written to and read from memory. Section 3.4 then shows how we distill the full model into a lightweight student that retains fine-grained recall at a fraction of the compute.

3.2 Problem Statement

Let $\mathcal{I}$ be the space of images and $\mathcal{T}$ be the space of text captions. A standard VLM, such as CLIP, consists of an image encoder $f_\theta : \mathcal{I} \to \mathbb{R}^D$ and a text encoder $g_\phi : \mathcal{T} \to \mathbb{R}^D$, where θ and ϕ are the model parameters and D is the dimension of a shared embedding space. The encoders are trained to maximize the cosine similarity of corresponding image-text pairs (I_i, T_i) while minimizing it for non-corresponding pairs within a batch. The image encoder f_θ , typically a Vision Transformer, processes an image I by dividing it into a sequence of N flattened patches. These are then linearly projected into a sequence of patch embeddings $X_0 = [x_{\text{cls}}; x_1, x_2, \ldots, x_N] \in \mathbb{R}^{(N+1)\times D_v}$, where x_{cls} is a prepended class token and D_v is the internal dimension of the transformer. This sequence is then processed by a series of L transformer blocks, each comprising a Multi-Head Self-Attention (MSA) sub-layer and a Multi-Layer Perceptron (MLP) sub-layer, where LN denotes Layer Normalization:

$$X'_l = \text{MSA}(\text{LN}(X_{l-1})) + X_{l-1}, \qquad \text{for } l = 1, \ldots, L \tag{3.1}$$

$$X_l = \text{MLP}(\text{LN}(X'_l)) + X'_l, \qquad \text{for } l = 1, \ldots, L \tag{3.2}$$

Notably, the standard MLP sub-layer, while effective for general feature transformation, lacks an explicit mechanism for storing and recalling specific, fine-grained visual details persistently. Limiting the model's ability to match detailed textual descriptions to precise image regions.

Our objective is to design a new vision encoder, $f_{\theta'}$, where the parameter set θ' is a superset of θ. This is achieved by replacing the MLP sub-layer in a selected subset of transformer blocks ($\mathcal{L}_{\text{mem}} \subset \{1, \ldots, L\}$) with a memory module, M_ψ, parameterized by ψ. The updated forward pass for a layer $l \in \mathcal{L}_{\text{mem}}$ becomes:

$$X_l = M_\psi(\text{LN}(X'_l)) + X'_l \tag{3.3}$$

The central challenges of our objective are as follows:

1. **Integration of Memory Modules:** How to design and integrate the memory module M_ψ such that it effectively stores and retrieves fine-grained information without disrupting the global context modeling of the MSA layers ?

2. **Knowledge Transfer:** How to efficiently train the new memory parameters ψ to be semantically aligned with the pre-existing knowledge encoded in θ and ϕ, avoiding the need to retrain the entire model from scratch ?

To address these challenges, we propose a methodology centered on a memory-augmented architecture and a specialized knowledge distillation and fine-tuning framework, which are detailed in the subsequent sections.

3.3 Memory-Augmented Vision Encoder

3.3.1 Overview of Memory-Enhanced Networks

The authors of [1] aim at augmenting the parameter space of models without significant computational costs (extra FLOPs). By replacing the Feed-forward Network of one or more transformer layers with memory layers, a dedicated capacity to store & retrieve information *cheaply* is enabled. Which according to [1], can then be used to augment dense neural networks and benefit them greatly. As mentioned in section 2.1, the core of transformers, attention mechanisms capture long-range dependencies by dynamically focusing on most relevant tokens. Memory layers work similarly to attention mechanism. A query ($q \in \mathbb{R}^n$), set of keys ($K \in \mathbb{R}^{N \times n}$) and values ($V \in \mathbb{R}^{N \times n}$) jointly work together to output a *soft* combination of values which are weighted accordingly to the similarity between a query and its corresponding keys. However, two major differences set memory layers apart from attention layers (keys & values in memory layers are *trainable* parameters, in terms of number of keys and values, memory layers are larger scale). As specified in [1] a simple memory layer can be described by the following equations:

$$\underbrace{I = \text{SelectTopIndices}(K_q), \quad s = \text{Softmax}(K_I q), \quad y = s\, V_I}_{\substack{I \text{ is a set of indices,} \\ s \in \mathbb{R}^k, \quad K_I, V_I \in \mathbb{R}^{k \times n}, \quad y \in \mathbb{R}^n}}$$

These layers, while computationally light, are memory-intensive from all the storage and retrieval capacities they facilitate. Thereby, requiring specific scaling and handling. Due to the trainable parameters of memory layers (keys, values) which are continually being trained and re-indexed, the query-key retrieval mechanism struggles with incorporating vector similarity techniques. A workaround by the authors [1] involved the splitting of a single key set ($N \times n$ matrix) into 2 smaller sets of half-dimensional keys. At lookup time, the model splits each query into two sub-vectors, retrieves top-k candidates from each half-key set, and then efficiently composes their scores to identify the true top matches. To scale, [1] shard both the parameter storage and the lookup computation across multiple GPUs. Each worker holds only its slice of the embedding table, performs its local queries and partial aggregations, then shares just the minimal results needed to reconstruct the final lookups. All while reducing both computation and memory overhead by orders of magnitude.

3.3.2 Base Architecture

Before diving deep into the memory slots and how they are used to augment the Vision Encoder of CLIP models, a brief overview of the internal structure of the Vision Encoder is necessary to understand beforehand. The work of original CLIP model laid in [28] made use of ViT [11] as one of the main architectures for its image encoder. While various other architectures [21, 22, 16] have been used in efforts to reach new performance heights in tasks like image-understanding and retrieval, it is no surprise why many CLIP models [31, 43, 23, 25] still rely on the ViT Backbone. The dominance and benefits of Transformer model [35] did not go unnoticed by the authors of ViT model [11] , which saw the potential in bridging Transformers with raw image patches. Instead of word tokens, they split images into a grid of fixed-size patches. Each patch is flattened and projected via a learnable linear layer into a D-dimensional embedding. A special "class" token is then added to the sequence of patch embedding. After passing through the Transformer encoder, the final state of this token serves as the global image representation for tasks like classification. Learnable 1D positional embeddings are added to every token so the model can recover spatial order, and the resulting sequence is processed by alternating blocks of multi-headed self-attention (MSA) and feed-forward (MLP) layers, each wrapped with layer normalization and residual connections. This patch-based Transformer backbone provides the foundation on which we now build our memory-augmented extensions. In the next section, we describe how we interleave dedicated key–value memory slots alongside the standard ViT layers. Enabling fine-grained feature storage & retrieval without sacrificing the global modeling power of the original architecture.

3.3.3 Memory Interface

The empirical gains from memory-augmented transformers in the LLaMA family (see [1] for details) suggest that injecting trainable key–value storage unlocks dramatic improvements in long-range reasoning and factual recall. Drawing inspiration from this cross-paradigm innovation, our goal is to replicate that "long-term memory" effect inside CLIP's vision encoder. As explained in section 3.3.2, the 2 alternating blocks (MSA & MLP) work together to enable the Transformer block to gather global context and refine each token's features independently. The MSA sub-layer handles the global context by allowing every patch to attend all other patches. Enabling the model to learn which regions of an image should influence one another. The MLP sub-layer on the other hand is responsible for combining information *within* each token's embeddings. When constructing the Vision Transformer block of CLIP model, we pass a memory object which encodes the hyperparameters of memory layer. Among other roles, this memory object also designates which transformer blocks should have their standard feed-forward (MLP) sub-layers swapped out for the memory-augmented modules. As explained in section 3.3.1, these memory layers are able to learn a trainable table of N keys and corresponding values. A top-k search over this table which yield top-k scores are then used to weight the stored values, resulting in a focused memory readout of fine-grained visual features. By infusing memory at the MLP slot, we supply each chosen layer with a persistent cache of local descriptors, while preserving the global context modeling of the self-attention sub-layer. For an in-depth visual depiction of memory layers please view the following figure 3.1.

In table 3.1 below, the exact "memory knobs" that can be tuned for fine-grained image understanding is provided and clarified before moving on to our distillation scheme that transfers these persistent, fine-grained capabilities into a lightweight student model. As mentioned earlier, each of the present hyperparameters below play a role in encoding and defining the memory layer's capabilities. These hyperparameters roughly control 4 main attributes of memory layers. The *mem_n_keys* parameter, defines the total memory size.

In other words, it controls the size of knowledge base and can therefore be considered as the

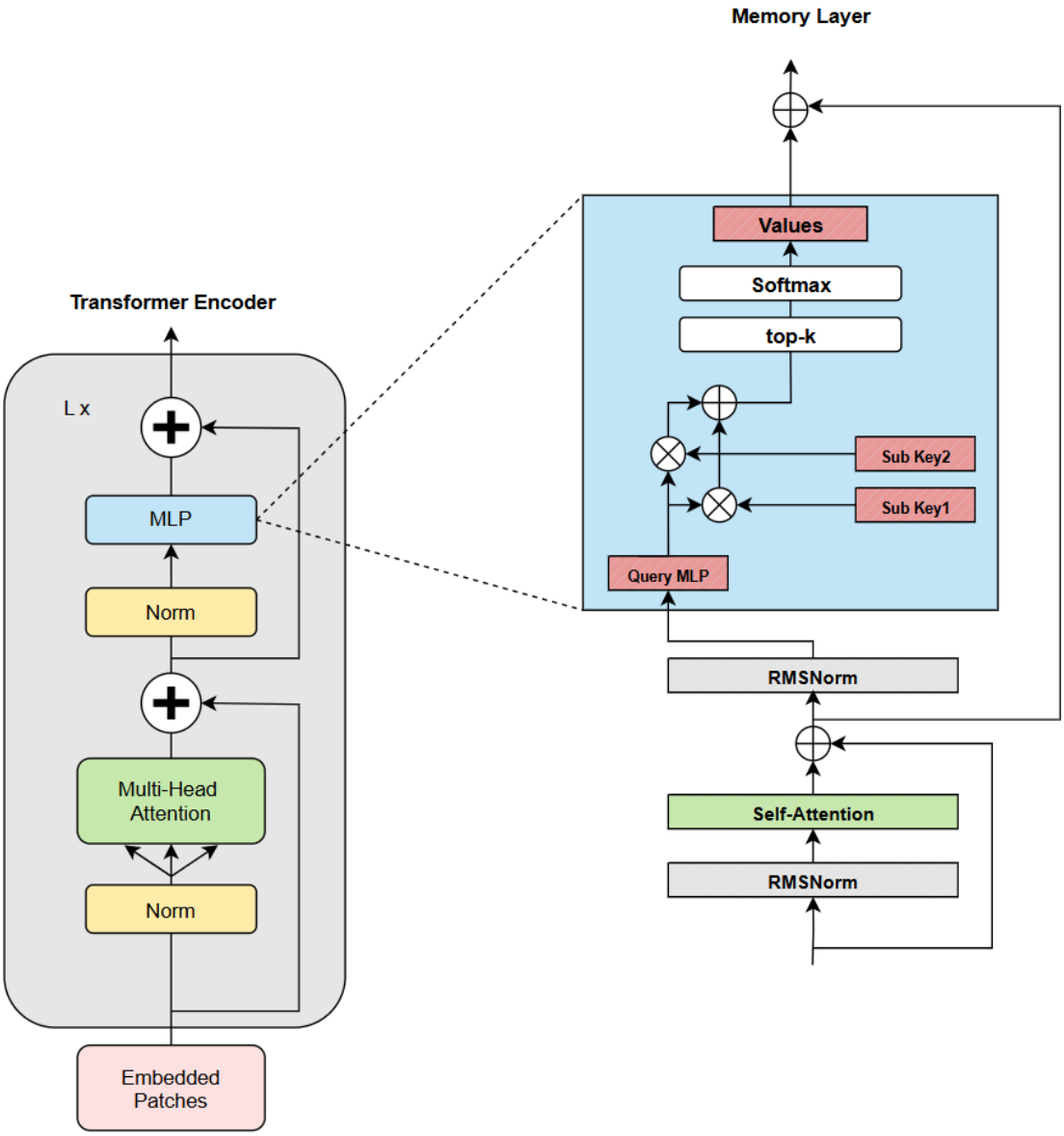

Figure 3.1: Memory Layer Architecture Structure

Parameter	Description
`layers`	Specifies which Vision Transformer blocks include memory slots (comma-separated indices).
`mem_n_keys`	Number of keys per memory sub-space, defining total capacity as $\texttt{mem\_n\_keys}^2$.
`mem_heads`	Number of parallel read-heads for simultaneous top-k lookups.
`mem_knn`	Number of nearest-neighbor slots retrieved per head.
`mem_k_dim`	Dimensionality of each key vector used in product-quantized lookup.
`mem_v_dim`	Dimensionality of each value vector (if positive, overrides block output dim).
`mem_share_values`	If true, all memory layers share a single global value table.
`value_fixed_lr`	Learning rate applied exclusively to the value embeddings.
`mem_gated`	Enables a learned gate to modulate the memory readout.

Table 3.1: Main parameters of the HashingMemory class for memory slots in the vision encoder

main impacting factor of memory capacity of these infused memory layers. The *mem_v_dim* parameter on the other hand controls the memory richness. The dimensionality of each vector

in the memory bank as well as the expressiveness of each memory slot is precisely impacted by this parameter. The number of memory slots that are then blended and retrieved for each query is controlled by the *mem_knn* parameter. Equally important, the *layers* parameter, determines the memory location and where these slots operate in the Vision Transformer architecture. Determining the type of features these slots will retain and store for further inferences.

3.4 Training Framework: Distillation & Fine-tuning

After introducing memory layers along with their capabilities and architectural changes, we shift to a complementary technique for downsizing and accelerating our enhanced model without ever needing to retrain from scratch. This idea originates from a revolutionary work and concept introduced by Hinton et.al [17], where the main aim was to be able to change the form of model while keeping the same knowledge. This was realized by making use of the class probabilities of a large model which acted as 'soft' targets for the training phase of a smaller model. Essentially, a large, over-parameterized "teacher" model trains a compact "student" by matching its softened output distributions. Building on this foundation, our distillation pipeline takes the standard pre-trained Vision Transformer (the *teacher*, f_T) and transfers its global alignment into a leaner *student* model (S_T) with memory-infused layers. The student model then mirrors the teacher's architecture, however, it replaces selected MLP layers with memory layers. While preserving the teacher's pretrained weights and parameters, the training phase exclusively updates the memory components while keeping all other parameters frozen. Enabling efficient, fast and scalable knowledge transfer and effectively equipping the student with fine-grained recall capabilities. Our pipeline, illustrated in Figure 3.2, consists of two distinct stages: an initial knowledge distillation phase to align the memory-augmented encoder, followed by a contrastive fine-tuning phase to adapt the model for specific downstream tasks.

3.4.1 Stage 1: Knowledge Distillation

The main objective of the distillation stage is to transfer the rich & general-purpose visual representations from a standard pre-trained Vision Transformer into our memory-augmented student model. Ensuring that the foundational alignment capabilities of the original CLIP model are preserved.

Teacher–student setup

Let I be an input image (or batch of images). We denote the frozen teacher's image encoder (f_T) and the student's (memory-augmented) image encoder (f_S) by eq.3.4 where B is batch size and D the embedding dimension.

$$z_T \;=\; f_T(I) \;\in\; \mathbb{R}^{B\times D},\; z_S \;=\; f_S(I) \;\in\; \mathbb{R}^{B\times D}, \tag{3.4}$$

We then optimize only the student's memory-slot parameters by minimizing the *cosine* distillation loss (eq. 3.5). By reusing the teacher's pretrained weights and updating only the memory layers, the student model preserves CLIP's original contrastive alignment while remaining lighter and faster at inference.

$$\mathcal{L}_{\text{distill}} = \frac{1}{B}\sum_{i=1}^{B} \ell_{\cos}(z_{T,i},\, z_{S,i}), \qquad \ell_{\cos}(u,v) = 1 - \frac{u\cdot v}{\|u\|_2\,\|v\|_2}\,. \tag{3.5}$$

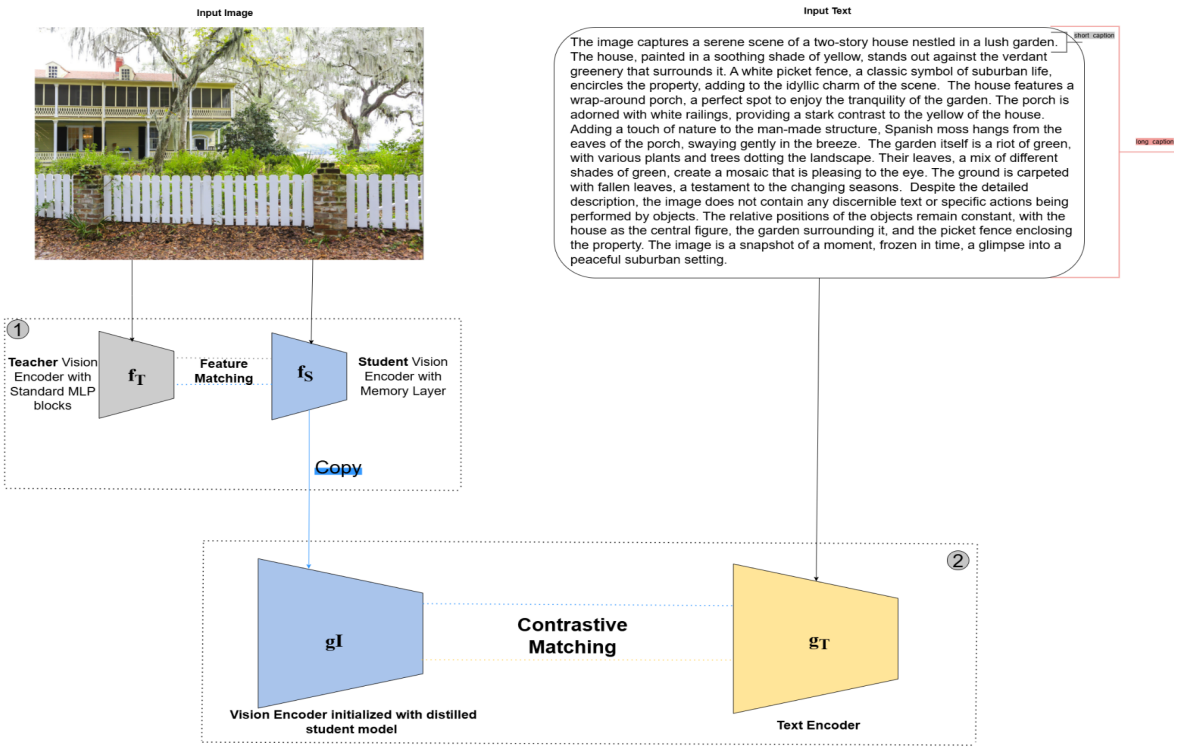

Figure 3.2: **Knowledge Distillation Procedure**. Initially, we instantiate a frozen CLIP ViT backbone without memory (teacher) alongside a matching ViT with Memory modules inserted into selected MLP sub-layers (student). During the Distillation stage (1), we perform feature matching between the teacher's and student's image-token embeddings, both the global class token and the per-layer memory readouts using a cosine-based loss. In the Contrastive Fine-Tuning stage (2), the distilled student vision encoder is paired with the frozen CLIP text encoder to perform zero-shot image–text retrieval, preserving the original alignment while benefiting from efficient, persistent memory lookups.

3.4.2 Stage 2: Contrastive Fine-tuning for Long-Context Retrieval

To ultimately extend the vision-language model's context length while preserving its capability to handle both long and short captions during retrieval, we jointly optimize two contrastive loss terms via a weighted summation, where α is a hyperparameter that balances the two objectives:

$$\mathcal{L}_{\text{total}}(I, x_{\text{long}}, x_{\text{short}}) = \alpha\, \mathcal{L}_{\text{long}}(I, x_{\text{long}}) + (1-\alpha)\, \mathcal{L}_{\text{short}}(I, x_{\text{short}}). \tag{3.6}$$

Both terms ($\mathcal{L}_{\text{long}}$,$\mathcal{L}_{\text{short}}$) use the same image features I which have been carefully retained and extracted via the infused memory layers. However, they process text independently in order to retain compatibility with standard CLIP benchmarks (via $\mathcal{L}_{\text{short}}$), learn robust long-context alignments (via dominant $\mathcal{L}_{\text{long}}$ term), and preserve computational efficiency by reusing image features. This two-stage framework provides an efficient pathway to developing a specialized, memory-augmented model that is both powerful and practical. In the upcoming section, a closer look at the training stage alongside the experimental setup will be covered before ultimately moving onto the Results (sec.5).

Chapter 4

Experimental Setup

4.1 Implementation Details

All of the below mentioned experiments were implemented with **Python 3.11** as our programming language. The deep learning framework was facilitated by **Pytorch 2.5.0** for CUDA 12.1. Our codebase [1] extends the original and official OpenAI CLIP [28] repository [2].Moreover, to enable data parallelism for various experimental trials and scale across multiple GPUs , we have utilized two high-performing clusters (Snellius[3] & DAS-5 [4]). In the Snellius cluster, depending on the available computation budget and system traffic, we alternated between 4 NVIDIA A100 GPUs and 4 NVIDIA H100 GPUs. Due to the constrained credits for this research and the computationally expensive nature of memory-augmented distillation and fine-tuning, we also employed two different GPU partitions from DAS-5 cluster(NVIDIA GTX 1080 Ti , NVIDIA Titan X Pascal).

4.2 Datasets

To assess the effectiveness of our memory-augmented Vision Transformer, we conduct experiments on three core downstream tasks (short-caption cross-modal retrieval, long-caption cross-modal retrieval and image segementation) using a total of ten datasets.
ShareGPT4V [5] is a dataset comprising of 100K descriptive captions generated by GPT4-Vision and 1.2M captions further generated by their caption model. More specifically, using a diverse set of image sources, each image was described by GPT-4-Vision which was prompted to extract fundamental attributes (appearance, spatial relations) alongside more detailed attributes (landmark names, locations). Those 100K high-quality examples were used to fine-tune an auxiliary captioning model, which then generated consistent, content-rich descriptions for a much larger pool of 1.2 million images drawn from public datasets. This dataset has therefore been used throughout our training and finetuning stage required for fine-grained memory learning (more details in sec.4.4).
MSCOCO [6] dataset has been used to evaluate our memory-augmented model on short-caption cross-modal tasks. It is characterized by its large scale, rich annotations. Consisting of 328K images and 2.5M labeled instances, this dataset distinguishes itself by focusing on segmenting individual object instances, providing pixel-level segmentation masks for each object instance. Proving its effectiveness and compatibility with of our model in settings that demand precise object-level understanding.

[1] `https://github.com/anesaibr/open_clip`
[2] `https://github.com/mlfoundations/open_clip`
[3] `https://servicedesk.surf.nl/wiki/spaces/WIKI/pages/30660184/Snellius`
[4] `https://www.cs.vu.nl/das5/home.shtml`

Flickr30K [41] is another dataset used in short-caption cross-modal tasks. It contains 30K images retrieved from Flickr website as well as 158K unique captions. Each image is described independently by five different annotators, thereby, offering a variety of descriptions for the same image.
DOCCI[27] was designed to address the limitations of existing datasets, particularly in providing fine-grained, comprehensive image descriptions for advanced cross-modal retrieval tasks. It consists of 14K images each with a human-annotated description. It is distinguished by the unique curation method which was solely enabled by a single researcher. Intentionally, collected to have substantially similar and contextually related groups of images. The image descriptions of this dataset are substantially longer and denser than datasets like MSCOCO. Offering a strict and challenging testbed for models aiming at understanding complex visual scenarios.
Urban-1k[44] dataset has been used for the long-caption retrieval downstream task due to the high average length of captions (101 words) accompanying the 200 carefully selected urban view images from an existing dataset [20].
IIW[15] is similarly used for long-caption retrieval tasks. With a collection of 9018 hyper-detailed image descriptions, the IIW dataset employs human-in-the-loop framework aided by VLM-generated drafts. As a result, IIW captions average over 217 tokens making it more comprehensive and specific compared to short-captioned datasets.
DCI[34] consists of 7805 images, each accompanied by a complete description aiming to capture the full visual detail present in the image. Much of the description in DCI is directly aligned to submasks of an image by building a hierarchy of submask-aligned annotations. DCI's density and alignment make it a rigorous benchmark for fine-grained VLMs where models must not only retrieve a correct whole-image caption but also match the right sub-caption to its region.
VOC20[13] uses images retrieved from Flickr to feature 20 object classes which are categorized into 4 main branches (vehicles,animals, household objects, and people). It is specifically designed to challenge models by presenting these objects with significant variability in pose, scale, and illumination within real-world scenes. Crucially, its exhaustive annotation for all instances of these 20 classes makes it a standard benchmark for reliable segmentation evaluation.
ADE20K[46] is a large-scale dataset that provides a benchmark for scene parsing. It is distinguished by its dense, pixel-level annotations created by a single expert to ensure high consistency across a vast 'open vocabulary' of classes. Its annotations capture not just discrete objects but also a detailed hierarchy of object parts.
CocoStuff[4] consists of pixel-level annotations where its primary goal is to enable a deeper analysis of 'stuff-thing' interactions. The dataset was created using a novel and highly efficient annotation protocol leveraging superpixels and the original 'thing' masks, which allowed for consistent and rapid labeling across all 164K images (or subset of 10k images).

4.3 Evaluation

When reporting the cross modal retrieval performances of the above mentioned downstream tasks, we use recall as the standard evaluation metric. During all experiments, the evaluation process remains uniform. For each retrieval split (e.g. COCO, Flickr30K, ShareGPT4V-1K/10K, DOCCI, Urban1K, IIW, DCI), we pair the model's image embeddings with the corresponding text embeddings and compute a full similarity matrix $S = \text{scale} \times (I\,T^{\top})$. We then measure the recall metrics on image-to-text, text-to-image, mean and median rank. All of this is handled by our evaluation scripts, which (a) collect and normalize features from the image and text loaders, (b) re-index IDs to align images and captions, and (c) extracts the rank-based metrics. By keeping the evaluation code identical across all models and datasets, we ensure

a fair comparison of zero-shot versus memory-augmented distilled performance on both short and long caption retrieval.

4.4 Training Pipeline

Table 4.1: Hyperparameters for Memory-Augmented Distillation Phase

Category	**Hyperparameter**	**Value**
Model Architecture	Teacher Model	ViT-B-16
	Memory Layers	4,6,8,10,11
	Memory Keys (K)	64
	Key Dimension	256
	Value Dimension	1024
	Shared Memory	True
Training Configuration	Batch Size	20×4
	Learning Rate	5×10^{-4}
	Epochs	38
	Loss Type	Cosine
	Weight Decay	0.1
	Precision	AMP-BF16
Data	Dataset	ShareGPT4V
	Training samples	1,245,902
	Validation samples	1000
	Input Size	224×224
	Context Length	77

The training pipeline of our model involves a multi-stage optimization framework. The distillation phase can be considered for being responsible for the memory-augmented knowledge transfer. During distillation, the objective is to compress a CLIP (ViT-B-16) model pretrained on OpenAI (a.k.a Teacher model) into a memory-enhanced student while preserving alignment capabilities. For a closer look at the exact parameters used to initialize this phase, table 4.1 contains the relevant details. While this phase is an important stage of our pipeline, the preceding phase, Fine-tuning is equally as important. The finetuning stage aims at adapting the distilled model to handle extended captions while retaining short-context performance. The existing table of distillation parameters can closely be followed to offer a correct finetuning initialization. However, during the training configuration, the standard cosine loss is shifted to a joint optimization between long and short captions (see eq.3.6) with fixed (80:20 α weighting). This forces the model to focus and aim at achieving 'near-perfect' alignment between long and detailed captions with images. Moreover, the learning rate, batch size, and epochs are reduced to $1 \times 10^{-}5$, 32 , and 10 respectively. These changes have been put into effect in order to prevent catastrophic forgetting and stabilize long-context learning all while only updating the memory layers and visual backbone in order to focus model's capacity on vision-language alignment without overfitting to text. Having detailed our training pipeline, we now validate its efficacy through zero-shot retrieval on standard benchmarks as well as ablation studies on memory layer configurations.

Chapter 5

Results

We now present the findings of our memory-augmented Vision Transformer across three downstream tasks: **short-caption cross-modal retrieval**, **long-caption cross-modal retrieval**, and **semantic segmentation**. We begin with a comparison of both zero-shot short- and long-caption retrieval performance across seven diverse datasets, reporting Recall@1 and Recall@5. This section further branches out in 2 subsections (5.1.1, 5.1.2) which compare our method against standard vanilla CLIP model as well as SOTA models. Furthermore, we analyze the robustness of our approach with a plug-in approach that integrates the TULIP[26] relative-position module into our model's text encoder, extending its textual context window and exploring its impact on short,long-caption retrieval tasks. Next, we report semantic segmentation mIoU values which analyze the impact of memory layers in pixel-level understanding. Finally, we then delve into ablation studies that isolate the contributions of memory-layer depth, key-value capacity, demonstrating how each design choice impacts alignment accuracy and model efficiency.

5.1 Cross-Modal Retrieval

5.1.1 Comparison to Standard CLIP

Short-caption cross-modal retrieval. To assess the effectiveness of our memory-augmented model in aligning images with concise textual descriptions, we first present results for the short-caption retrieval task. The comprehensive comparison, detailed in Table 5.1, evaluates our model against the baseline CLIP model on a diverse set of benchmarks. The results reveal that our architecture causes a distinct and advantageous shift in retrieval behavior. While CLIP establishes a strong, balanced baseline, our model consistently excels in the Text-to-Image (T2I) retrieval direction, particularly in improving Recall@5, suggesting an enhanced ability to identify the correct image within a candidate pool. On the standard MSCOCO and Flickr30k datasets, our model demonstrates a clear specialization. For MSCOCO, we observe a notable improvement in Text-to-Image recall. Similarly, on Flickr30k, our model boosts T2I R@5 performance by over a full point. This consistent gain in R@5 suggests that our memory mechanism helps create more discriminative visual representations, making the model more robust at identifying the correct image among the top 5 candidates. However, this T2I enhancement comes with a trade-off, as the model shows a slight degradation in Image-to-Text (I2T) performance on both standard benchmarks. This pattern suggests our architecture prioritizes the refinement of visual features for precise search over the broader semantic space required for caption ranking. The fine-grained datasets, DOCCI and IIW, offer deeper insights into our model's behavior. On IIW, our model reinforces the trend seen in standard benchmarks by achieving a surpassing T2I R@5 score. This is a crucial result, as it indicates our model's

Method	Standard Retrieval								Fine-grained Retrieval							
	MSCOCO				Flickr30k				DOCCI				IIW			
	T2I		I2T		T2I		I2T		T2I		I2T		T2I		I2T	
	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5
CLIP [28]	30.24	54.23	**49.42**	**73.66**	**61.22**	84.71	**79.68**	**95.27**	**13.57**	**27.17**	**42.18**	**68.94**	**25.76**	43.83	**79.08**	**94.12**
Ours	**30.43**	**56.2**	48.0	72.9	60.0	**85.8**	77.0	94.2	12.49	25.42	36.2	63.56	25.03	**44.79**	74.8	92.16

Table 5.1: Short caption cross-modal retrieval comparison on validation splits for both standard benchmarks (MSCOCO, Flickr30k) and fine-grained settings (DOCCI, IIW). All models are pretrained on OpenAI dataset under same training configurations and use ViT-B-16 as their vision encoder. The beating results of *our* model are in **bold**.

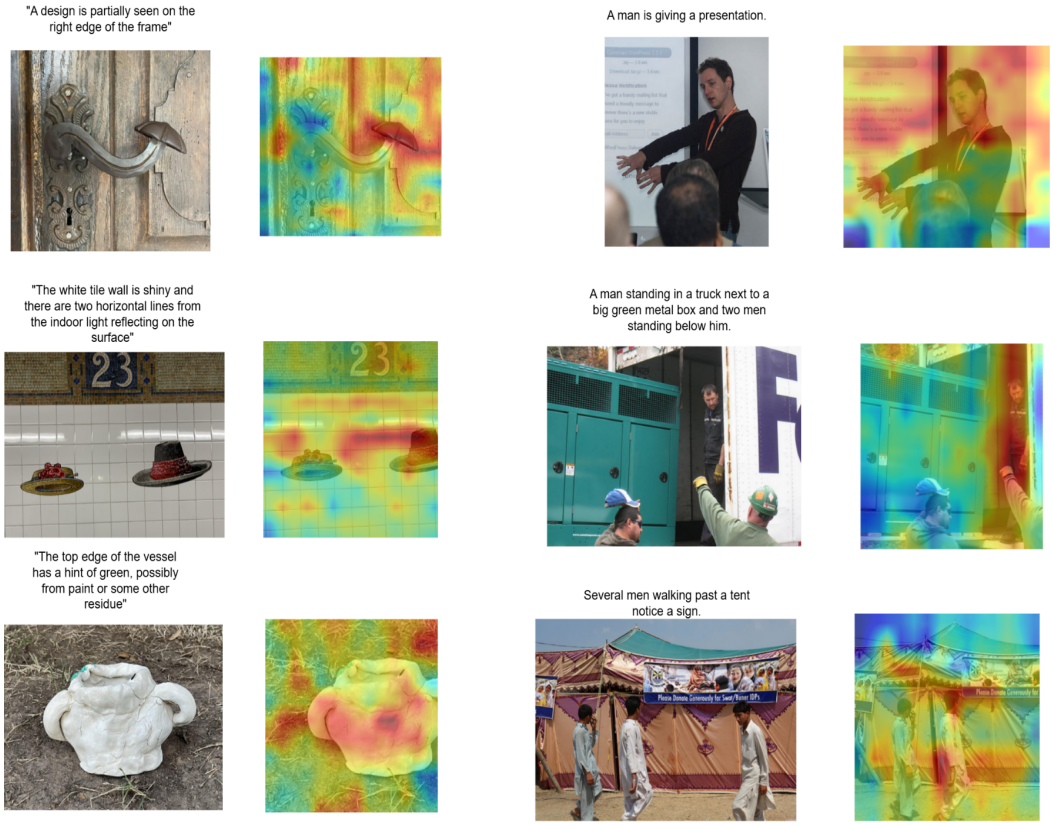

Figure 5.1: **Short Cross-modal Retrieval.** A visualization of the attention maps of *our* model on IIW [15] dataset (left set of pictures) and Flickr30k [41] dataset (right set of pictures) when performing text-to-image retrieval. Showcasing our model's focus on a retrieved image given a text query.

architectural advantages are particularly effective for detail-oriented retrieval where finding the right image is challenging. In contrast, on the DOCCI dataset, our model does not outperform the baseline, and across both fine-grained datasets, the I2T performance sees a more significant drop compared to CLIP. When inspecting the obtained attention maps in Fig.5.1, it is clear that our model can effectively localize and focus on the specific fine-grained parts of an image when aligning a text query with an image. The examples of T2I retrieval on both IIW and Flickr30k showcase the precise focus of our model on the specific objects and attributes mentioned in a text query, effectively filtering out irrelevant background information. This strong grounding capability is the primary source of its improved Text-to-Image retrieval performance, especially on fine-grained datasets where identifying subtle details is crucial to a successful match.

Long-caption cross-modal retrieval. While concise textual descriptions offer a measure of core alignment, it is important to understand and delve deeper into the capabilities of a model

when challenged with longer, more detailed captions. Models that can parse complex sentences and ground multiple entities within a single description demonstrate a more sophisticated level of vision-language understanding. Therefore, as detailed in Table 5.2, we have provided a comprehensive evaluation on three distinct long-caption benchmarks to examine our model's performance on this demanding task.

Method	DCI				Urban-1k				ShareGPT4V			
	T2I		I2T		T2I		I2T		T2I		I2T	
	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5
CLIP [28]	36.46	57.42	38.76	57.98	46.9	69.90	62.10	84.40	**74.1**	**94.6**	**80.8**	**95.6**
Ours	**44.2**	**67.63**	**44.68**	**66.73**	**59.6**	**82.90**	**64.3**	**87.20**	69.7	90.7	52.2	87.6

Table 5.2: Long cross-modal retrieval comparison on validation splits of long-caption benchmarks. The table shows Recall@1 and Recall@5 scores for image-to-text (I2T) and text-to-image (T2I) retrieval. All models use ViT-B-16 as vision encoder and are pre-trained on OpenAI dataset. Best results are in **bold**.

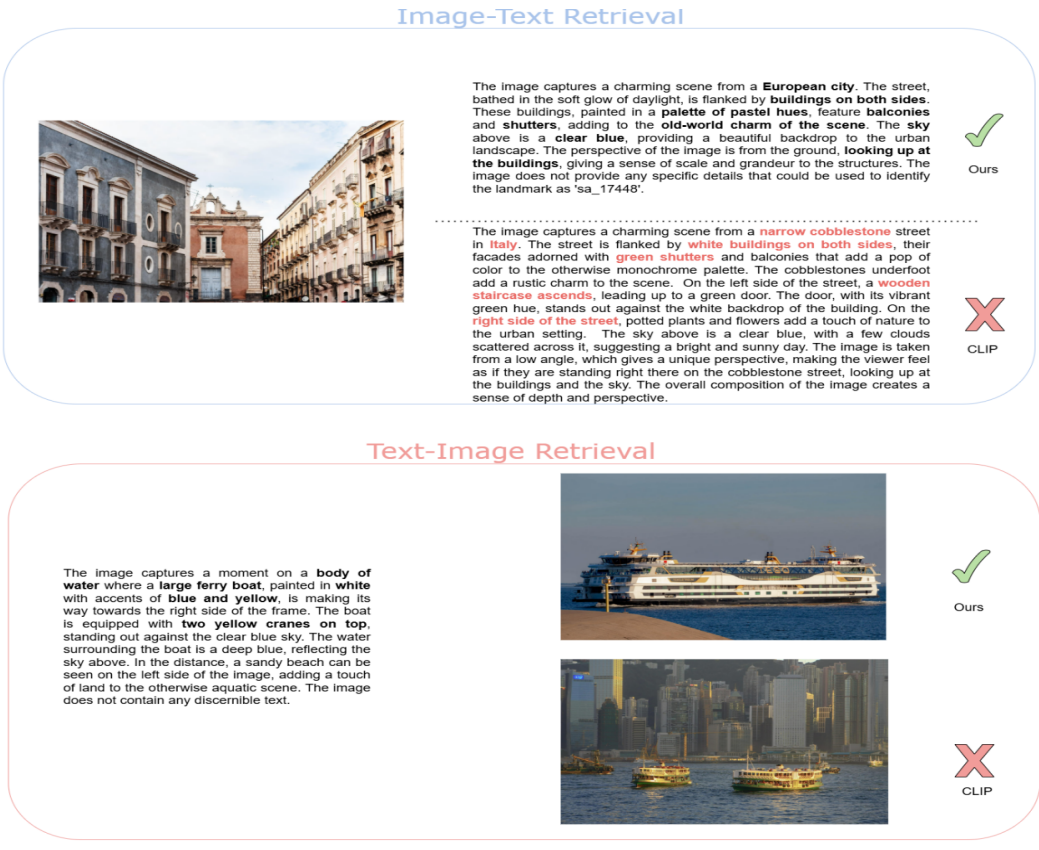

Figure 5.2: **Long Cross-modal Retrieval.** A visual depiction and comparison on the ShareGPT4V validation dataset between our model and original CLIP baseline model using the ViT-B-16 backbone.

When observing the performance of our model on the DCI and Urban-1k benchmarks, it's strength becomes immediately apparent. On DCI , our model decisively outperforms CLIP across all metrics, achieving a remarkable +7.7 point gain in T2I R@1 and a substantial +5.9 point gain in I2T R@1. The improvements in R@5 are even more pronounced, showcasing

our model's superior ability to place the correct item within the top candidates. This trend is emphasized even more on the Urban-1k benchmark, which focuses on complex urban scenes. These dramatic gains strongly suggest that our memory-augmented architecture is particularly effective at disentangling and grounding the many objects, attributes, and relationships present in lengthy and descriptive captions of dense scenes. The consistent, large-margin victories on both DCI and Urban-1k validate our approach for accurate, description-based retrieval. However, the results on the ShareGPT4V benchmark reveal an important nuance about our model's specialization. On this dataset, our model does not surpass the CLIP baseline, showing a particular deficit in the Image-to-Text retrieval task. As highlighted in sec.4, the ShareGPT4V dataset contains more synthetic and AI-generated captions in contrast to the detailed, human-authored descriptions of visual scenes of DCI and Urban-1k datasets. Suggesting that CLIP's broader, more varied pre-training on web-scale data may grant it greater robustness to the more abstract language found in ShareGPT4V captions.

The impact of text encoder & caption length. Aside from the augmented vision encoder that our model enabled through memory layers, we wanted to further examine the impact it would have when working together with a more 'flexible' text encoder. As covered in sec.2.3, the work of TULIP [26] expanded caption capacity and lifted the original CLIP's token limit through the usage of relative instead of fixed positional encodings. Therefore, when swapping the original text encoder with TULIP text encoder, while keeping the vision encoder infused with memory layers, the following impact on cross-modal retrieval can be seen in table 5.3.

	Short Caption								**Long Caption**							
	MSCOCO				IIW				DCI				Urban-1k			
Method	T2I		I2T		T2I		I2T		T2I		I2T		T2I		I2T	
	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5
CLIP [28]	30.24	54.23	**49.42**	**73.66**	**25.76**	43.83	**79.08**	**94.12**	36.46	57.42	38.76	57.98	46.90	69.90	62.10	84.40
Ours	**30.43**	**56.20**	48.00	72.90	25.03	**44.79**	74.8	92.16	44.20	67.63	44.68	66.73	59.60	82.90	64.30	87.20
Ours+TULIP[26]	26.27	51.70	41.12	67.94	19.10	35.03	65.03	86.93	**44.42**	**68.49**	**49.60**	**73.63**	**68.60**	**89.60**	**71.00**	**91.30**

Table 5.3: Cross-modal retrieval comparison validation splits for both standard short caption benchmarks (MSCOCO, IIW) and fine-grained long-caption settings (DCI, Urban-1k). The "Ours+TULIP" row refers to our model, which was further fine-tuned via added memory layers and a TULIP text encoder on the ShareGPT4V dataset. All models are pretrained under identical settings using ViT-B/16 as the vision encoder, and OpenAI dataset. Best scores are in **bold**.

The process of integrating TULIP text encoder involved a strategic "transplant" of the pre-trained TULIP text encoder into our model architecture. Crucially, the TULIP text encoder's weights were frozen during this phase. The objective was to teach the vision encoder to align its feature representations with the rich, pre-existing embedding space of the powerful TULIP text encoder. On Urban-1k, our model achieves a 15.1% relative improvement in T2I R@1 and a 10.4% improvement in I2T R@1 compared to our memory-augmented model with standard text encoder. On DCI, a similar trend is observed, particularly in I2T retrieval, with an 11.0% relative improvement in R@1. The model is better at selecting the most accurate long description for a given image, demonstrating that the vision encoder has successfully learned to produce features that align with the fine-grained details (e.g., specific architectural styles, spatial relationships) encoded by the TULIP text tower. This success can be directly attributed to TULIP's RoPE and NTK-aware scaling, which allow it to process and represent long captions without information loss, a critical limitation of the original baseline CLIP's 77-token encoder. However, due to current approach of integrating an alternative text encoder during fine-tuning phase, a significant performance degradation on the standard short-caption benchmarks is exhibited . Causing it to lead to a form of domain-specific overfitting, which sacrifices generalist

capabilities for more specialized and fine-grained ones. This experiment confirms that our memory-augmented vision encoder is flexible enough to be aligned with different text encoders, but the alignment process must be carefully considered relative to the target application.

The obtained results of these two retrieval tasks demonstrate our model's development of a highly specialized feature space. Where it prioritizes descriptive precision over generalized performance. This is consistently observed in its superior Text-to-Image (T2I) performance across both short and long captions, where it excels at finding the correct image from a textual description. This strength proves to be a decisive advantage in the long-caption setting, where our model excels at grounding the complex, literal narratives found in benchmarks like DCI and Urban-1k. The consequence of this focused approach, however, is evident in the two main datasets (ShareGPT4V[5], DOCCI[27]) which test different capabilities. On DOCCI, which contains groups of highly similar images, and on the ShareGPT4V dataset, our model struggles to isolate the subtle, distinguishing cues needed for a correct match. This indicates our memory module excels at creating a comprehensive scene representation which appears more efficient at matching a whole description than isolating specific elements to generate a caption.

5.1.2 Comparison with State-of-the-Art Models

Situating our method within the current SOTA landscape, we contrast our approach with SOTA baselines (MetaCLIP[40] , SigLIP[43],LongCLIP[44],FLAIR[38]) under identical evaluation protocols and compare their retrieval accuracy across standard and fine-grained retrieval benchmarks.

Short-caption cross-modal retrieval. On the standard retrieval and fine-grained tasks benchmarks, our memory-augmented model demonstrates unique and informative performance. As shown in Table 5.4, our model does not outperform leading SOTA models like SigLIP and MetaCLIP, which achieve dominant results across the board. This performance gap is expected and can be directly attributed to their significant advancements in pre-training paradigms and specific architectural components. SigLIP[43], for example, replaces the standard batch-level softmax loss with a simpler and more scalable pairwise sigmoid loss. Allowing it to train effectively on larger batch sizes and making it more robust to the inherent noise in web-scale data (webli). MetaCLIP[40], on the other hand, achieves its strength by shifting its focus in carefully curating its training data. Its transparent pipeline filters a massive raw data pool using a predefined metadata list and balances the concept distribution. Proving that data quality is one of the key players in dominant and consistent performance. Interestingly, when comparing our model with the remaining baselines several insights emerge. As already established in section 5.1.1, our model achieves slight improvements over the standard CLIP on several key metrics. It also regularly outperforms LongCLIP[44] in I2T retrieval by a significant margin, indicating that LongCLIP's specialization for long captions may have compromised its ability on standard short-caption benchmarks. While our model does not have the same performance behavior w.r.t FLAIR[38], it still manages to display highly competitive values. This establishes that the architectural change of our model (replacing MLP blocks with fine-tuned memory layers) preserves, rather than degrades, CLIP's general-purpose representations.

Long-caption cross-modal retrieval. When evaluating the models on long-caption fine-grained benchmarks, as detailed in Table 5.5, the clear dominance of SigLIP and MetaCLIP remains persistent. Despite this, the unique strengths of specialized and modified architectures (including ours) become visible. FLAIR achieves its SOTA performance on Urban-1k T2I through a novel text-conditioned attention pooling mechanism, allowing the text caption to dynamically query relevant local image regions. Similarly, LongCLIP excels at Urban-1k I2T by directly extending the text encoder's context length to 248 tokens via knowledge-preserved stretching of its positional embeddings. Our model positions itself as a strong and balanced

Method	Standard Retrieval								Fine-grained Retrieval							
	MSCOCO				Flickr30k				DOCCI				IIW			
	T2I		I2T		T2I		I2T		T2I		I2T		T2I		I2T	
	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5
CLIP [28]	30.24	54.23	49.42	73.66	61.22	84.71	79.68	95.27	13.57	27.17	42.18	68.94	25.76	43.83	79.08	94.12
SigLIP [43] *	**47.15**	**72.11**	**65.48**	**86.14**	**75.58**	**92.84**	**89.05**	**98.62**	**20.60**	**35.86**	**57.48**	**82.14**	**33.82**	**53.01**	83.66	**97.71**
MetaCLIP [40] *	39.30	64.84	56.44	80.40	69.88	89.47	85.40	97.63	17.56	32.38	50.64	76.50	30.83	48.79	**85.13**	95.75
Ours	30.43	56.20	48.00	72.90	60.00	85.80	77.00	94.20	12.49	25.42	36.20	63.56	25.03	44.79	74.80	92.16
FLAIR [38] *	37.68	64.45	51.60	77.12	65.72	86.77	78.70	95.17	15.02	30.90	35.70	63.48	31.53	52.98	75.49	92.48
LongCLIP [44] *	36.73	61.91	12.82	28.40	69.98	89.49	52.86	84.42	15.45	30.02	20.36	37.96	27.45	45.55	55.56	80.88

Table 5.4: Extended analysis on Short caption cross-modal retrieval with additional SOTA models. All models contain a ViT-B-16 as their vision encoder. CLIP and our model are pre-trained on OpenAI's proprietary dataset. Baselines marked with (∗) use distinct pre-training data: SigLIP is pre-trained on WebLI, MetaCLIP on a 400M subset of CommonCrawl, FLAIR on CC3M-recap, and LongCLIP results are from their officially released checkpoint. Best results are in **bold**.

competitor to both. It outperforms LongCLIP on T2I (DCI) and FLAIR on I2T(Urban-1k). Additionally, our model manages to also outperform SigLIP and MetaCLIP on both T2I and I2T (Urban-1k). Thereby, indicating that our vision-centered memory augmentation provides a balanced improvement for both encoding and decoding complex scenes. Focusing on the complex dataset of ShareGPT4V, featuring extremely long and descriptive captions, MetaCLIP's dominating performance does not go unnoticed. Yet, our model's performance still validates the efficacy of memory-augmented architectures for long-caption retrieval.

Method	Long-caption Retrieval											
	DCI				Urban-1k				ShareGPT4V			
	T2I		I2T		T2I		I2T		T2I		I2T	
	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5
CLIP [28]	36.46	57.42	38.76	57.98	46.90	69.90	62.10	84.40	74.10	94.60	80.80	95.60
SigLIP [43] *	**56.23**	**76.72**	**57.80**	**77.86**	62.00	82.90	62.70	82.90	83.20	96.20	85.90	97.00
MetaCLIP [40] *	52.94	73.56	52.81	73.70	58.70	80.20	64.00	82.70	**84.50**	**96.30**	**87.20**	**97.40**
Ours	44.20	67.63	44.68	66.73	59.60	82.90	64.30	87.20	69.70	90.70	52.20	87.60
FLAIR [38] *	50.51	73.17	47.32	69.94	**69.50**	**88.90**	63.50	87.10	–	–	–	–
LongCLIP [44] *	43.78	65.75	48.25	69.35	61.30	80.20	**71.50**	**90.10**	–	–	–	–

Table 5.5: Extended analysis on Long cross-modal retrieval with additional SOTA models. Using validation splits of long-caption benchmarks, the table shows Recall@1 and Recall@5 scores for image-to-text (I2T) and text-to-image (T2I) retrieval. All models use ViT-B-16 as vision encoder. CLIP and our model are pre-trained on OpenAI's proprietary dataset. Baselines marked with (∗) use distinct pre-training data: SigLIP is pre-trained on WebLI, MetaCLIP on a 400M subset of CommonCrawl, FLAIR on CC3M-recap, and LongCLIP results are from their officially released checkpoint. The best results are in **bold**.

5.2 Semantic Segmentation

To steer-away from retrieval and instead investigate visual understanding , we aimed at additionally testing our model's capabilities in zero-shot semantic segmentation . As observed in table 5.6, our model achieves a modest drop of roughly 2 to 5 mIoU points across the three semantic datasets (VOC20[13], ADE20K[46], and COCO-Stuff[4]) in comparison to the original

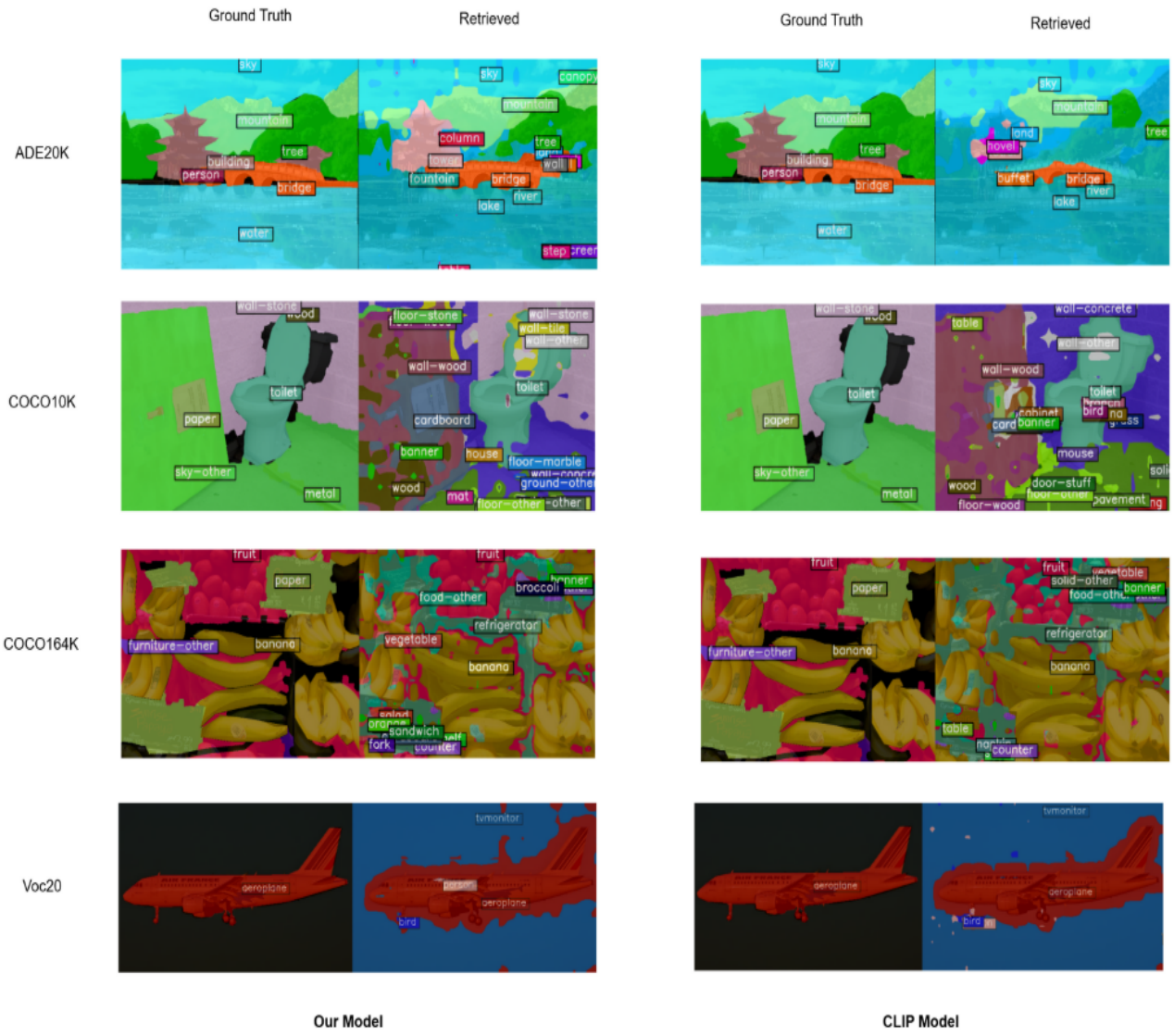

Figure 5.3: **Semantic Segmentation.** For each dataset, the left side shows the ground truth and the memory-augmented model's prediction ("Retrieved"). The right side shows the ground truth and the baseline CLIP's prediction ("Retrieved").

CLIP baseline model. This demonstrates that injecting memory layers in Vision Transformer does not severely compromise the encoder's ability to produce strong pixel-level features in a zero-shot setting. However, the results from our extensive evaluation reveal a consistent and noteworthy pattern.

Our memory-augmented CLIP model *underperforms* the baseline CLIP ViT-B/16 across all four datasets. This outcome, while counterintuitive to the goal of performance enhancement, provides critical insights into the nature of pre-trained vision-language models and the challenges of modifying their architecture. The dataset of VOC 20, characterized by small number of segmentation classes (20), includes exceptionally high mIoU values for both models. The high performance of the original baseline model suggests that the features learned by the standard ViT-B-16 are already highly discriminative and spatially precise for these common object categories. Therefore, as observed in figure 5.3, the introduction of memory layers suggest that its mechanism disrupts the fine-grained feature localization capabilities of the model and instead introduces a level of feature instability.

When analyzing the ADE20K and COCO-Stuff dataset (10k,164k), containing 150, 171, and 182 classes respectively, the mIoU values significantly drop for both of the models. Rea-

Method	VOC20	ADE20K	COCO-Stuff10K	COCO-Stuff164k
CLIP [28]	**81.52**	**16.46**	**25.9**	**22.77**
Ours	74.86	14.64	23.32	20.07

Table 5.6: Mean intersection over union (mIoU) for zero-shot **semantic segmentation** on the VOC20 [13], ADE20K [46], and COCO-Stuff [4] datasets. All models use ViT-B-16 as vision encoder.

sons being, the smaller and complex objects alongside overlapping 'stuff' (e.g sky,grass) and 'things' (e.g car, person). Despite this, our memory-augmented model consistently scores lower than the baseline. When closely inspecting the figures in fig.5.3, the detailed results show that the memory model particularly struggles with unstructured or texture-based classes. The main objects are partially recognized, but the overall scene structure is lost to a noisy collection of smaller, less confident predictions. When observing the nature scene of ADE20K dataset, our model partially recognizes the bridge yet collects a noisy collection of smaller less confident predictions. In the object dense scene of COCO-Stuff164k, the baseline model correctly identifies dominant banana class and smaller hidden classes like 'table', 'fruit', 'banner'. Our model , while correctly identifying dominant object, on the other hand associated features of one object (banana) with semantically related but visually absent concepts (broccoli, orange). The qualitative results of the segmentation task uncovered several insights of our model's memory layers. When replacing the Vision Transformer's blocks with our fine-tuned memory modules, it is evident that our model struggles to hold together large-scale scene context, resulting in more fragmented region predictions. While correctly identifying objects, the model produces noisier and less precise boundaries than the baseline causing it to occasionally hallucinate and over-correlate semantically related concepts. As explained in section 3.3.3, our method replaces certain MLP blocks with Memory layers. During training phase, these layers were trained from scratch while keeping most of vision tower and model components frozen. Potentially, causing insufficient training to learn powerful representations as the baseline mode. The replacement of MLP blocks, therefore, created potential information bottleneck and lost its ability to maintain the coarse and global context necessary for accurate and powerful zero-shot semantic segmentation tasks.

5.3 Ablation Study

Memory Layer Configuration. To better understand the factors that drive and impact the performance of our model, a more granular look on memory-related hyperparameters will be covered below. A detailed explanation of the most important hyperparameters of memory-layers has been previously covered in table 3.1. Therefore, given the great importance and effect that these hyperparameters have in the memory layers, an ablation study consisting of 3 phases was conducted. All of these experimentations of memory layers configurations have been conducted through a constrained subset of ShareGPT4V dataset (one shard - 10k samples) while showing the obtained recall@k metrics after 10 epochs of training. Firstly, the establishment between Capacity vs Richness was examined.

More specfically, the effect of memory capacity (mem_n_keys) and memory richness (mem_v_dim)

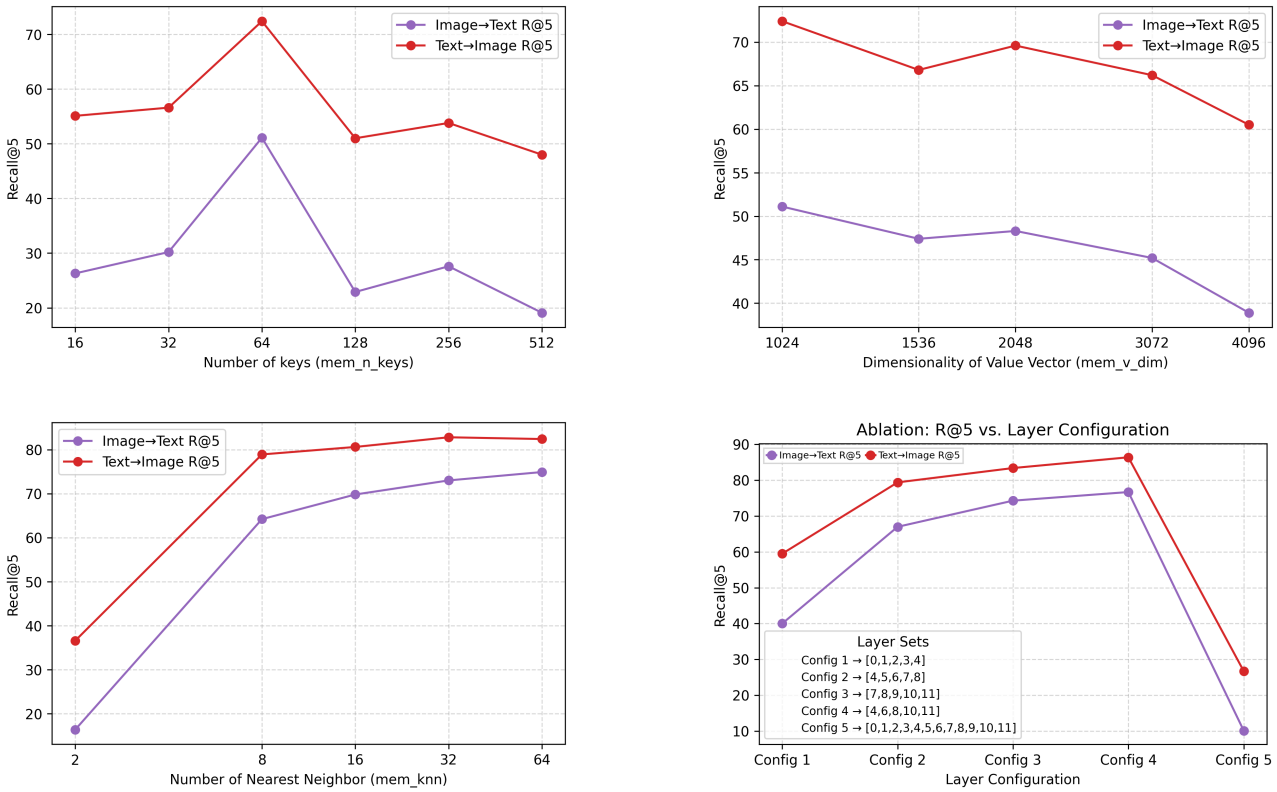

Figure 5.4: Impact of Memory Layers Hyperparameter Configuration on cross-modal retrieval task.

was further analyzed to understand the most optimal values for the knowledge base and expressiveness of each memory slot in it. When observing the top two sub-graphs of figure 5.4, a deeper understanding of whether it is better to have (more & simpler) or (fewer & richer) memory keys is provided. Ultimately, the configuration (mem_n_keys =64, mem_v_dim = 1024) proved to be the most optimal. Implying that the configuration with the higher mem_v_dim (richness) and lower mem_n_keys (capacity) decisively won. The configurations with a low richness consistently performed the worst within their tiers. This implies that for a memory slot to store meaningful, fine-grained concepts, each needs to be a sufficiently high-dimensional, expressive vector with a moderate knowledge base capacity. Next, the retrieval mechanism was investigated. For this experimentation phase, the impact of mem_knn was observed due to its active role in controlling the nature of retrieved information. This investigation therefore aimed at inferring how the model should access the learned knowledge, either by sharply or smoothly retrieving them. It is worth noting that mem_knn is limited by the size of mem_n_keys. The retrieval function splits the query (q1, q2) and the memory keys (keys1, keys2) where the total number of memory slots is $mem\_n\_keys^2$. Each set of keys has mem_n_keys entries. So, keys1 has mem_n_keys different key vectors, and keys2 also has mem_n_keys key vectors. Implying that when the retrieval is trying to find top k largest elements via topk , it is trying to select knn items from a list of size mem_n_keys . Therefore, the results below prove that the model reacts well when it creates a 64x64 = 4096 candidate pool and uses the full number of keys (64) to select top 64 candidates from 4096 pool. Lastly, the optimal memory layer location is examined in order to discover where in the processing hierarchy is memory more effective. In the bottom right plot, the early concentrated layers (”0,1,2,3,4”), mid-level concentrated layers (”4,5,6,7,8”) and all layer mode(”0...11”) seem to not have that big of a positive impact on the model. Ultimately, the best configuration is a late-sparse configuration (”4,6,8,10,11”) which is where the vision encoder retains most of relevant and fine-grained details of an image.

Chapter 6

Conclusion

This thesis investigated memory-augmented vision–language models and demonstrated their impact on cross-modal retrieval and segmentation tasks. It set out to address a critical limitation in fine-grained image understanding in existing Vision-Language models like CLIP [28]. More particularly, it aimed at tackling their inability to capture intricate visual details, handle long and dense captions related to subtle visual patterns. While addressing this shortcoming is an ongoing and growing field of interest, existing fine-grained VLM rely on fragile external captioning pipelines, cause significant computational overhead, and lack mechanisms for persistent storage and recall of localized features. Aiming at overcoming these deficiencies, we proposed a novel framework featuring a memory-augmented Vision Transformer alongside a specialized knowledge distillation pipeline. Our objective was to provide visual backbones with the ability to persistently remember and leverage fine-grained visual details across multiple inferences while reducing reliance on problematic external pipelines or excessive compute.

In response to our first question on architectural design, we found that by strategically replacing the Multi-Layer Perceptron (MLP) sub-layers of a Vision Transformer with trainable, key-value memory modules, we successfully enhanced the model's architectural capacity for fine-grained feature storage. Regarding our second question on efficient knowledge transfer, a teacher-student knowledge distillation regime was then employed to efficiently transfer fine-grained recall capabilities from a pre-trained teacher model (CLIP ViT-B-16) to our leaner, memory-enhanced student model via exclusively updating the memory components all while freezing other parameters. This approach proved to be highly effective, enabling us to create a specialized model without degrading the foundational alignment learned during large-scale pre-training. Finally, addressing our third question on empirical performance and trade-offs, our ten dataset evaluation for retrieval and segmentation establishes a clear and interpretable trend. Our memory-augmented model established a new level of performance on long-caption fine-grained retrieval benchmarks ,like DCI and Urban-1k, decisively outperforming the standard CLIP model (Table 5.1) and remaining competitive with highly specialized SOTA models like FLAIR and LongCLIP (Table 5.5). Confirming that the memory layers effectively learn to ground the complex narratives found in descriptive captions of various benchmarks.

Despite these notable successes, our work also revealed the trade-offs and areas of improvement. A slight decline in cross-modal retrieval performance on certain datasets was present. Suggesting that while our model excels at finding images given precise descriptions, it struggles with broader semantic understanding or isolating subtle cues in scenarios with high similarity. Furthermore, during semantic segmentation tasks, our model underperformed the baseline model of CLIP. Indicating that the introduction of memory layers which are trained from scratch , while most of the vision tower remains frozen, may disrupt the fine-grained feature localization capabilities. Potentially, losing coarse and global context necessary for accurate pixel-level predictions. Our ablation studies , on the other hand, provided important insights regarding

the optimal memory-layer configurations. Revealing that high-dimensional, expressive vectors with a moderate knowledge base which are placed in a late-sparse configuration within the Vision Transformer is where fine-grained details are best stored and extracted. Furthermore, the flexibility and ability of our memory-augmented vision encoder to align with different text encoders for specialized tasks was showcased with the TULIP text encoder.

To this end, the work presented in this thesis successfully introduced a memory-augmented Vision Transformer, leveraging knowledge distillation, to significantly enhance fine-grained image understanding, particularly in text-to-image retrieval for long and detailed captions. While demonstrating a powerful capabilities in this domain, it also highlighted the ongoing challenge of maintaining broad generalization across diverse tasks, especially in image-to-text retrieval and zero-shot semantic segmentation. Future work should focus on mitigating these trade-offs by exploring more adaptive training strategies for memory layers. Exploring a fine-tuning approach where vision encoder layers are gradually made trainable to prevent information bottleneck and developing dynamic memory mechanisms that can adapt to varying task demands and caption lengths without sacrificing overall robustness. By addressing these challenges, our memory-augmented framework can set the scene for more comprehensive and efficient VLMs capable of understanding images at a fine-grained level across a range of applications.

Bibliography

[1] Vincent-Pierre Berges, Barlas Oğuz, Daniel Haziza, Wen tau Yih, Luke Zettlemoyer, and Gargi Ghosh. Memory layers at scale, 2024.

[2] Diego Bonilla-Salvador, Marcelino Martínez-Sober, Joan Vila-Francés, Antonio José Serrano-López, Pablo Rodríguez-Belenguer, and Fernando Mateo. Pixlore: A dataset-driven approach to rich image captioning, 2024.

[3] Florian Bordes, Richard Yuanzhe Pang, Anurag Ajay, Alexander C. Li, Adrien Bardes, Suzanne Petryk, Oscar Mañas, Zhiqiu Lin, Anas Mahmoud, Bargav Jayaraman, Mark Ibrahim, Melissa Hall, Yunyang Xiong, Jonathan Lebensold, Candace Ross, Srihari Jayakumar, Chuan Guo, Diane Bouchacourt, Haider Al-Tahan, Karthik Padthe, Vasu Sharma, Hu Xu, Xiaoqing Ellen Tan, Megan Richards, Samuel Lavoie, Pietro Astolfi, Reyhane Askari Hemmat, Jun Chen, Kushal Tirumala, Rim Assouel, Mazda Moayeri, Arjang Talattof, Kamalika Chaudhuri, Zechun Liu, Xilun Chen, Quentin Garrido, Karen Ullrich, Aishwarya Agrawal, Kate Saenko, Asli Celikyilmaz, and Vikas Chandra. An introduction to vision-language modeling, 2024.

[4] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context, 2018.

[5] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. Sharegpt4v: Improving large multi-modal models with better captions, 2023.

[6] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollar, and C. Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server, 2015.

[7] Yuhan Chen, Ang Lv, Jian Luan, Bin Wang, and Wei Liu. Hope: A novel positional encoding without long-term decay for enhanced context awareness and extrapolation, 2024.

[8] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 2818–2829. IEEE, June 2023.

[9] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023.

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[11] 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, 2021.

[12] Sivan Doveh, Assaf Arbelle, Sivan Harary, Roei Herzig, Donghyun Kim, Paola Cascante-bonilla, Amit Alfassy, Rameswar Panda, Raja Giryes, Rogerio Feris, Shimon Ullman, and Leonid Karlinsky. Dense and aligned captions (dac) promote compositional reasoning in vl models, 2023.

[13] Mark Everingham, Luc Van Gool, Christopher Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88:303–338, 06 2010.

[14] Alex Fang, Gabriel Ilharco, Mitchell Wortsman, Yuhao Wan, Vaishaal Shankar, Achal Dave, and Ludwig Schmidt. Data determines distributional robustness in contrastive language image pre-training (clip), 2022.

[15] Roopal Garg, Andrea Burns, Burcu Karagol Ayan, Yonatan Bitton, Ceslee Montgomery, Yasumasa Onoe, Andrew Bunner, Ranjay Krishna, Jason Baldridge, and Radu Soricut. Imageinwords: Unlocking hyper-detailed image descriptions, 2024.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[17] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015.

[18] Dong Jing, Xiaolong He, Yutian Luo, Nanyi Fei, Guoxing Yang, Wei Wei, Huiwen Zhao, and Zhiwu Lu. Fineclip: Self-distilled region-based clip for better fine-grained understanding. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 27896–27918. Curran Associates, Inc., 2024.

[19] Sanghwan Kim, Rui Xiao, Mariana-Iuliana Georgescu, Stephan Alaniz, and Zeynep Akata. Cosmos: Cross-modality self-distillation for vision language pre-training, 2025.

[20] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Fei-Fei Li. Visual genome: Connecting language and vision using crowdsourced dense image annotations, 2016.

[21] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023.

[22] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation, 2022.

[23] Yangguang Li, Feng Liang, Lichen Zhao, Yufeng Cui, Wanli Ouyang, Jing Shao, Fengwei Yu, and Junjie Yan. Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm, 2022.

[24] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2024.

[25] Norman Mu, Alexander Kirillov, David Wagner, and Saining Xie. Slip: Self-supervision meets language-image pre-training, 2021.

[26] Ivona Najdenkoska, Mohammad Mahdi Derakhshani, Yuki M. Asano, Nanne van Noord, Marcel Worring, and Cees G. M. Snoek. Tulip: Token-length upgraded clip, 2025.

[27] Yasumasa Onoe, Sunayana Rane, Zachary Berger, Yonatan Bitton, Jaemin Cho, Roopal Garg, Alexander Ku, Zarana Parekh, Jordi Pont-Tuset, Garrett Tanzer, Su Wang, and Jason Baldridge. Docci: Descriptions of connected and contrasting images, 2024.

[28] 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, 2021.

[29] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016.

[30] Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. How much can clip benefit vision-and-language tasks?, 2021.

[31] Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. Eva-clip: Improved training techniques for clip at scale, 2023.

[32] Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. Winoground: Probing vision and language models for visio-linguistic compositionality, 2022.

[33] Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms, 2024.

[34] Jack Urbanek, Florian Bordes, Pietro Astolfi, Mary Williamson, Vasu Sharma, and Adriana Romero-Soriano. A picture is worth more than 77 text tokens: Evaluating clip-style models on dense captions, 2024.

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[36] Chien-Yao Wang, I-Hau Yeh, and Hong-Yuan Mark Liao. Yolov9: Learning what you want to learn using programmable gradient information, 2024.

[37] Wei Wu, Kecheng Zheng, Shuailei Ma, Fan Lu, Yuxin Guo, Yifei Zhang, Wei Chen, Qingpei Guo, Yujun Shen, and Zheng-Jun Zha. Lotlip: Improving language-image pre-training for long text understanding, 2024.

[38] Rui Xiao, Sanghwan Kim, Mariana-Iuliana Georgescu, Zeynep Akata, and Stephan Alaniz. Flair: Vlm with fine-grained language-informed image representations, 2024.

[39] Hu Xu, Saining Xie, Xiaoqing Ellen Tan, Po-Yao Huang, Russell Howes, Vasu Sharma, Shang-Wen Li, Gargi Ghosh, Luke Zettlemoyer, and Christoph Feichtenhofer. Demystifying clip data, 2024.

[40] Hu Xu, Saining Xie, Xiaoqing Ellen Tan, Po-Yao Huang, Russell Howes, Vasu Sharma, Shang-Wen Li, Gargi Ghosh, Luke Zettlemoyer, and Christoph Feichtenhofer. Demystifying clip data, 2024.

[41] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014.

[42] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it?, 2023.

[43] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training, 2023.

[44] Beichen Zhang, Pan Zhang, Xiaoyi Dong, Yuhang Zang, and Jiaqi Wang. Long-clip: Unlocking the long-text capability of clip, 2024.

[45] Kecheng Zheng, Yifei Zhang, Wei Wu, Fan Lu, Shuailei Ma, Xin Jin, Wei Chen, and Yujun Shen. Dreamlip: Language-image pre-training with long captions, 2024.

[46] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset, 2018.