POLITECNICO DI TORINO

Master's Degree in Data Science and Engineering

Master's Degree Thesis

Mitigating the Modality Gap in Vision-Language Pre-Trained Models

Supervisors

Prof. Giuseppe RIZZO

Dr. Federico D'ASARO

Candidate

Iman MOROVATIAN

April 2025

Abstract

One of the key challenges in vision-language models is the modality gap, which refers to the misalignment between image and text embeddings when projected into a shared latent space due to the inherent differences between the two modalities. This gap poses significant challenges for tasks that rely on seamless integration of visual and textual information, such as image-text retrieval, caption generation, and cross-modal understanding. While previous research has explored the causes of the modality gap and its effects on various downstream tasks, comprehensive studies on how model architecture influences this gap remain limited.

This thesis investigates the role of model architecture in contributing to the modality gap, with a particular focus on shared-encoder architectures, where both images and text are processed by the same encoder network. Shared-encoder models offer potential benefits in terms of efficiency and parameter sharing, but they also introduce challenges related to modality-specific representations.

Building on prior work, this thesis proposes a novel method to mitigate the modality gap within the shared-encoder architecture. The proposed approach integrates specific loss functions and fine-tuning strategies designed to encourage better alignment between visual and textual embeddings. The effectiveness of this method is evaluated through extensive experiments, demonstrating its impact on reducing the modality gap and improving performance on two critical downstream tasks: image-text retrieval and vector arithmetic-based operations.

Furthermore, the thesis provides a comparative analysis of the shared-encoder architecture against the more traditional dual-encoder architecture, highlighting the strengths and limitations of each in terms of modality alignment, computational efficiency, and downstream task performance. The findings contribute to a deeper understanding of the modality gap in vision-language models and offer insights into architectural choices and training strategies that can enhance cross-modal learning.

Acknowledgements

First of all, I want to thank Prof. Giuseppe Rizzo for all his support throughout this journey. I appreciate his advice and help in overcoming the challenges I faced.

I also want to thank my colleague, Dr. Federico D'Asaro. Without his support, it would have been impossible to complete this journey. It was an honor to work with you.

I am grateful to the LINKS Foundation for allowing me to use their resources during the experiments for this thesis.

A heartfelt thanks to my friends who helped me settle in Turin—Mahyar, Mohammadhossein, Mahdi, Mohammad, and Hossein. The wonderful memories we created together will always bring a smile to my face, and I feel lucky to have had you by my side.

Finally, I want to express my deepest gratitude and love to my family. I know I would not have achieved anything in my life without their unwavering support. My mother's love, my father's encouragement, my brothers' energy, and Fatemeh's motivation gave me the strength to overcome every obstacle. I hope to someday repay even a fraction of their support, though I know words can never fully capture my gratitude.

Table of Contents

List of Tables VI

List of Figures VIII

1 Introduction 1
- 1.1 Thesis Outlines . . . 2
- 1.2 Contributions . . . 2

2 Vision-Language Models 3
- 2.1 Vision-Language Pre-trained models (VLPs) . . . 3
- 2.2 VL Tasks . . . 3
- 2.3 Image-Text Tasks . . . 4
  - 2.3.1 Architectures . . . 6
  - 2.3.2 Pre-Training Objectives . . . 7
- 2.4 CLIP model . . . 11
- 2.5 from Dual-Encoders to Unified-Encoder . . . 12
  - 2.5.1 VISTA . . . 13

3 Modality Gap 15
- 3.1 What is modality gap . . . 15
- 3.2 Causes of modality gap . . . 20
- 3.3 Consequences of modality gap . . . 20

4 Methodology 22
- 4.1 Tailored Loss Functions . . . 22
- 4.2 Fine-Tuning Strategies . . . 24

5 Experiments 26
- 5.1 Downstream Tasks . . . 26
- 5.2 Datasets . . . 26
- 5.3 Experimental Setup . . . 27

5.4 Results 27
5.4.1 Retrieval and Modality Gap 28
5.4.2 Vector Arithmetic and Modality Gap 28

6 Conclusion and Future Work 33

A Retrieval Results of L_{CUA} and L_{CUAXU} 35

B Modality Gap Measured by CD and CMD 38

Bibliography 40

List of Tables

2.1 Classification of VLPs based on training procedure, multimodal fusion, and having decoder. Based on [31]. 9

3.1 Central Moment Discrepancy (CMD) and Centroids Difference (CD) computed for some pre-trained models based on their generated image and text embeddings on the validation split of MSCOCO [74] 19

5.1 Retrieval results of CLIP [10] and VISTA [25] on MSCOCO [74] and Conceptual Captions [82] datasets. Fine-tuned using L_{clip}. ZS^* stands for Zero-Shot, indicating that it has not been fine-tuned. The best performance of each model is bold. 29

5.2 Modality Gap vs. Retrieval: The retrieval performance of the LU strategy across different loss functions is compared to the modality gap to examine the relationship between retrieval effectiveness and the modality gap. 30

5.3 SIMAT [20] scores of CLIP [10] and VISTA [25] fine-tuned on MSCOCO [74] and Conceptual Captions [82] datasets using different loss functions. ZS^* stands for Zero-Shot, indicating that it has not been fine-tuned. The highest score for each model is bold. . 31

5.4 Modality gap of CLIP [10] and VISTA [25] fine-tuned on MSCOCO [74] and Conceptual Captions [82] datasets using different loss functions. ZS^* stands for Zero-Shot, indicating that it has not been fine-tuned. The minimum gap for each model is bold. 32

A.1 Retrieval results of CLIP [10] and VISTA [25] on MSCOCO [74] and Conceptual Captions [82] datasets. Fine-tuned using L_{CUA}. ZS^* stands for Zero-Shot, indicating that it has not been fine-tuned. FT^* stands for Fine-Tuned. The best performance of each model is bold. For the convenience, the results related to models fine-tuned by L_{clip} in LU setting are provided. 36

A.2 Retrieval results of CLIP [10] and VISTA [25] on MSCOCO [74] and Conceptual Captions [82] datasets. Fine-tuned using L_{CUAXU}. ZS^* stands for Zero-Shot, indicating that it has not been fine-tuned. FT^* stands for Fine-Tuned. The best performance of each model is bold. For the convenience, the results related to models fine-tuned by L_{clip} in LU setting are provided. 37

B.1 Modality gap (measured by CD and CMD) of CLIP [10] and VISTA [25] on MSCOCO [74] and Conceptual Captions [82] datasets. Fine-tuned using different loss functions. ZS^* stands for Zero-Shot, indicating that it has not been fine-tuned. The minimum of each model is bold. 39

List of Figures

2.1 Examples of image-text retrieval, visual question answering, visual reasoning, and image captioning. Adopted from [31]. 5
2.2 Example of visual grounding (phrase grounding). Adopted from [31] 5
2.3 Example of visual grounding (referring expression). Adopted from [31] 5
2.4 General architecture of a VLP 6
2.5 Possible categories for the classification of VLPs 8
2.6 Contrastive Pre-Training. Adopted from [10] 12
2.7 The way that the VISTA [25] encodes the input. It is worth mentioning that ViT [38] operates as a tokenizer for the image 14

3.1 UMAP [75] visualization of generated embeddings from pre-trained models on MSCOCO [74] validation split. 17
3.2 Visualization of cosine similarities of the generated embeddings from pre-trained models on MSCOCO [74] validation split. 18

4.1 Different settings for the fine-tuning of the CLIP [10]. The blue color means the component is **L**ocked (frozen), while the red one means the component is (**U**nlocked) going to be fine-tuned. 25
4.2 Different settings for the fine-tuning of the VISTA [25]. The blue color means the component is **L**ocked (frozen), while the red one means the component is (**U**nlocked) going to be fine-tuned. 25

Chapter 1

Introduction

Humans perceive the world through multiple sensory inputs. For instance, our eyes provide visual information about our surroundings, while our ears allow us to localize sounds from various objects. The human brain processes these multimodal inputs simultaneously, efficiently integrating complementary information from different senses to enable us to perform a wide range of tasks [1].

Inspired by the brain's multimodal processing capabilities, the deep learning community has adopted the concept of learning from multiple modalities to tackle diverse tasks effectively. In recent years, there has been a surge in large, pre-trained multimodal foundation models [2], which are trained on web-scale datasets spanning multiple modalities. These models learn versatile data representations that can be transferred to numerous uni-modal and multimodal downstream tasks [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]. This thesis focuses specifically on pre-trained vision-language models, which are designed to learn joint representations of image and text data.

One notable challenge in the field of vision-language models is the modality gap. When images and text are projected into a common latent space, their embeddings often do not align perfectly due to inherent differences in the nature of the two modalities. The modality gap, its roots, and consequences have been studied extensively [1, 16, 17, 18, 19, 20, 21, 22, 23]. Most analyses have focused on CLIP-like (dual-encoder) vision-language models. However, in recent years, a new architecture, the shared-encoder, has been introduced and employed by vision-language models [24, 25].

This thesis aims to investigate and reduce the modality gap in the shared-encoder architecture. Furthermore, the effect of reducing the modality gap on downstream tasks is analyzed, followed by a comparison with the dual-encoder architecture.

1.1 Thesis Outlines

In Chapter 2, vision-language models and the tasks they can solve are introduced. Next, models specifically designed for image-text tasks are categorized based on architectures and pre-training objectives. The shared-encoder architecture is then presented. Additionally, CLIP and VISTA are discussed as examples of dual-encoder and shared-encoder architectures, respectively. Chapter 3 thoroughly examines the modality gap, exploring its causes and its effects on downstream tasks. Chapter 4 introduces the method for reducing the modality gap in the shared-encoder architecture. Experimental results are provided in Chapter 5, including comparisons between dual-encoder and shared-encoder architectures and discussions on the relationship between gap reduction, retrieval, and vector arithmetic. Finally, Chapter 6 concludes the thesis.

1.2 Contributions

To the best of our knowledge, the contributions of this thesis are as follows:

- While previous studies have explored the impact of factors such as random initialization and dataset mismatches on the modality gap, this thesis investigates the role of model architecture in contributing to the modality gap.
- Building upon prior work, this thesis proposes a method to reduce the modality gap in shared-encoder architectures.
- This thesis examines the effect of modality gap reduction on the performance of shared-encoder models in tasks such as retrieval and vector arithmetic, and provides a comparative analysis with dual-encoder architectures.

Chapter 2

Vision-Language Models

In this chapter, an overview of vision-language models is presented, focusing on the tasks they handle, the diverse architectures employed, and the pre-training objectives that guide their learning. Special attention is given to CLIP, one of the most influential models in this domain. Finally, the shared-encoder architecture (unified-encoder) is introduced, with VISTA presented as an example of this type of architecture. Through this discussion, readers are provided with a comprehensive understanding of key advancements in vision-language modeling.

2.1 Vision-Language Pre-trained models (VLPs)

Vision-Language (VL) research exists at the intersection of computer vision and natural language processing (NLP), focusing on developing models that can learn from both images and text simultaneously. Building on the success of language pre-trained models in NLP, such as BERT [26], RoBERTa [27], GPT-3 [28], BART [29], and T5 [30], Vision-Language Pre-trained models (VLPs) have gained significant attention. VLPs are VL models trained on large multimodal datasets and aim to learn representations that can be transferred to a variety of both uni-modal and multi-modal downstream tasks. These models are designed to generalize across tasks, leveraging their ability to understand and process both visual and linguistic information simultaneously [31, 1].

2.2 VL Tasks

Generally, VL tasks can be formulated as $y = f(x; \theta)$, where a VL model f parameterized by θ is trained to generate output y based on input x. From two perspectives, VL tasks can be categorized [31].

1. Based on the modalities of x and y: VL tasks can be grouped into **image-text** and **video-text** tasks.

2. Based on how y is generated: VL tasks can be divided into **understanding** (e.g. image-text retrieval) and **generation** (e.g. image captioning) tasks. In the former, y is selected by f from a candidate list, and in the latter, y is generated by f.

This thesis focuses specifically on understanding image-text tasks, which are detailed in the following section.

2.3 Image-Text Tasks

Image-text tasks can be subdivided into several categories:

- Image-to-Text Retrieval: retrieving textual description(s) given an image as the query. (Figure 2.1)

- Text-to-Image Retrieval: retrieving image(s) given a text as the query. (Figure 2.1)

- Vector Arithmetic in Multimodal Embeddings: inspired by the well-known analogy properties of word embeddings (e.g., king - man + woman = queen), this task explores algebraic transformations within the shared latent space of vision-language models, specifically examining how textual modifications can be applied to image embeddings for retrieval [20]. These transformations are represented as delta vectors in the multimodal space. Given an image and a corresponding text transformation query (e.g., cat to dog), the transformation is computed as the difference between the embeddings of the two words and subsequently added to the image embedding. The resulting modified embedding is then used to retrieve the most similar image from a database, ideally capturing the intended change while maintaining other visual elements.

- Visual Question Answering (VQA): providing a correct answer to a question based on an image given the question and image. The answer can be chosen from a candidate list (multiple-choice) or it can be open-ended [32]. Figure (2.1)

- Image Captioning: generating a (single-sentence or multiple-sentence) caption for a given image. (Figure 2.1)

- Visual Reasoning: evaluating specific reasoning capabilities (e.g. spatial understanding [33], logical reasoning [34], and commonsense reasoning [35]). Most visual reasoning tasks are formulated as VQA [31]. (Figure 2.1)

- Visual Grounding: Aligning a text query with the relevant object in an image and predicting its bounding box. In phrase grounding (Figure 2.2), multiple entities in the text are mapped to corresponding regions in the image, while in referring expression comprehension (Figure 2.3), specific objects mentioned in the text are localized with bounding boxes [31].

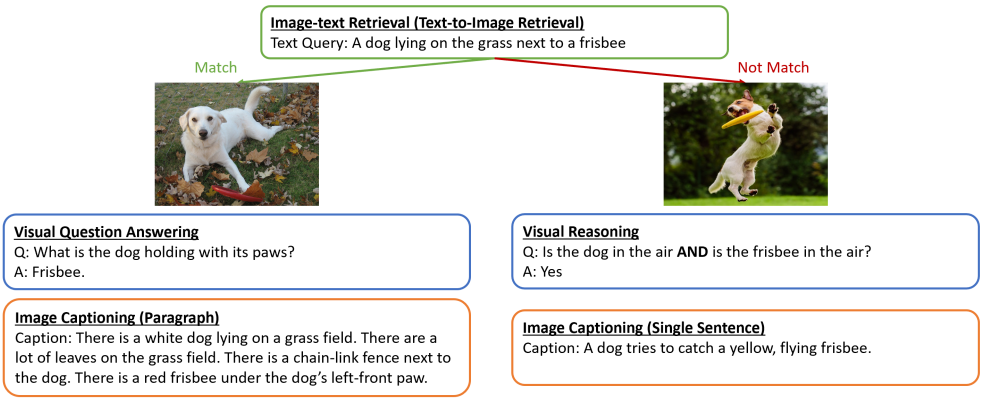


Figure 2.1: Examples of image-text retrieval, visual question answering, visual reasoning, and image captioning. Adopted from [31].

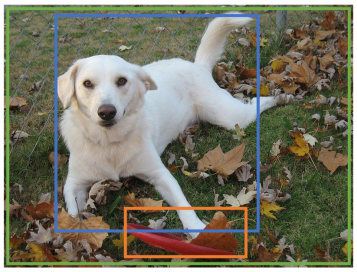


Figure 2.2: Example of visual grounding (phrase grounding). Adopted from [31]

Figure 2.3: Example of visual grounding (referring expression). Adopted from [31]

The tasks mentioned above have been extensively studied, leading to the development of various models to address them. In the remainder of this section, an

in-depth analysis of VLPs designed for image-text tasks is presented, focusing on their architecture and pre-training objectives.

2.3.1 Architectures

Typically, VLPs have an architecture where image and text features are extracted by separate encoders, an image encoder for visual features and a text encoder for linguistic features. After extraction, these features are passed to a multimodal fusion phase to generate a cross-modality representation. This representation is then either passed directly to the output layer or first processed through a decoder layer before reaching the output layer (Figure 2.4).

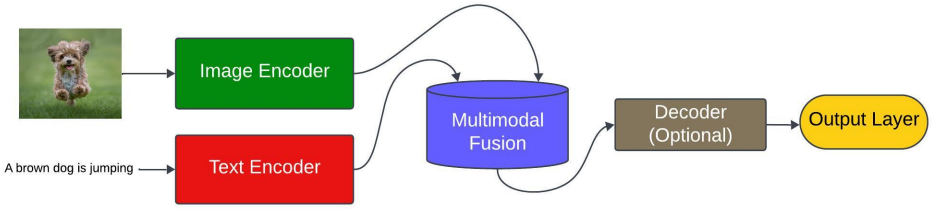


Figure 2.4: General architecture of a VLP

Considering the *multimodal fusion*, the architectures of VLPs can be categorized into two main types [31]:

- Dual Encoder: In this architecture, images and text are encoded separately, and a simple cosine similarity is used to measure the intersection between the encoded images and text, facilitating modality interaction or multimodal fusion [31].
- Fusion Encoder: Similar to the dual encoder, images and text are encoded separately; however, multimodal fusion is achieved through Transformer [36] layers [31].

The dual encoder is effective for image retrieval and can produce a robust image encoder trained from scratch. However, it is less suitable for VQA and visual reasoning due to its limited deep multimodal fusion capabilities [31]. Conversely, the fusion encoder demonstrates superior performance in VQA and visual reasoning but is less effective for image retrieval, as it encodes all possible image-text pairs to compute similarity scores [31]. See Table 2.1 for the examples of each type.

In the context of *fusion-encoder* methods, two key perspectives for deeper categorization are the **training procedure** and the **application of Transformer [36] layers** for multimodal fusion [31]. Considering the *training procedure*, there are two main approaches:

- Two-Stage Pre-training: Earlier VLP methods typically followed this approach, where the model first relied on a pre-trained object detector to extract image region features. In this stage, the object detector was used to identify and describe important parts of an image, and only after this step were these features passed on to the main model to learn the relationship between the image and the corresponding text. This approach separates the process of feature extraction and model learning into two distinct phases [31].

- End-to-End Pre-Training: More recent methods have adopted this approach, where the model learns to extract image features and understand their relationship with text in a single, unified process. Instead of relying on a pre-trained object detector, these models use convolutional neural networks (CNNs) [37], vision transformers (ViTs) [38], or image patch embeddings to directly process the images. Since the gradients of the model can flow back through the entire system, including the vision backbone, the model can improve its feature extraction and relationship learning simultaneously. This end-to-end method has led to state-of-the-art performance across major vision-language tasks, as it enables more efficient and integrated learning compared to the two-stage process [31]. See Table 2.1 for the examples of each approach.

Considering the *application of Transformer [36] layers* for multimodal fusion, there are two main types:

- Merged Attention: Text and visual featuLU res are concatenated and fed into a single Transformer [36] block for joint processing [31].

- Co-Attention: Text and visual features are processed separately through distinct Transformer blocks, with cross-attention mechanisms facilitating cross-modal interaction. Also, it is possible to use only image-to-text cross-attention modules [31]. See Table 2.1 for the examples of each type.

2.3.2 Pre-Training Objectives

Various pre-training objectives are used by VLPs, with the following being four of the most popular [31].

- Masked Language Modeling (MLM): Given an image-text pair $(\tilde{v}, \tilde{w})$, some tokens (e.g. m) of the text are randomly masked ($\tilde{w}_m$) and then it is tried to predict them based on the image ($\tilde{v}$) and the unmasked tokens ($\tilde{w}_{\bar{m}}$) by minimizing the negative log-likelihood:

$$\mathcal{L}_{MLM}(\theta) = -\mathbb{E}_{(\tilde{w},\tilde{v})\sim D} \log P_\theta(\tilde{w}_m|\tilde{w}_{\bar{m}}, \tilde{v}) \tag{2.1}$$

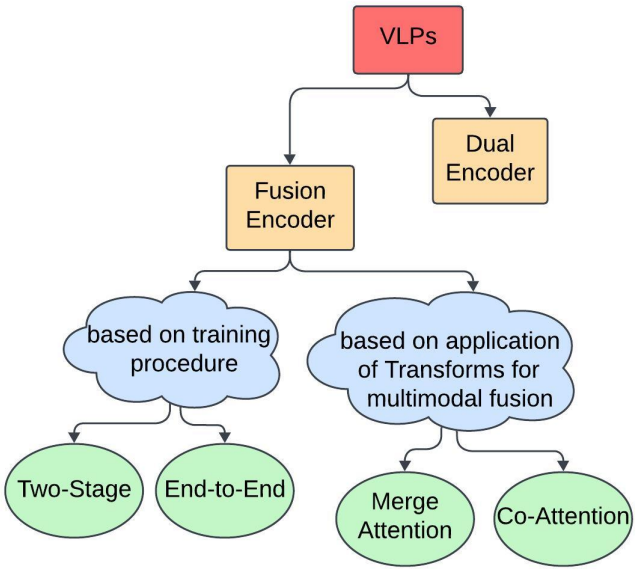


Figure 2.5: Possible categories for the classification of VLPs

where θ shows the learnable parameters and D indicates the training data [31].

There are three variants of MLM. One of them is **Seq-MLM**, which modifies the standard MLM objective by introducing a causal attention mechanism. In Seq-MLM, when predicting a masked token, the model can only use the tokens that precede it in the sequence. This mirrors the process of language generation during tasks such as image captioning, where the model generates words sequentially without knowledge of future words. By restricting the model's attention in this way, Seq-MLM enhances its ability to perform tasks that require a step-by-step generation of language, aligning more closely with real-world applications [31].

Another variant is **Direct Language Modeling (LM)**, which differs from MLM by focusing on generating entire sentences from scratch. In this approach, the model is trained to produce captions by predicting each token one by one, relying on the image and previously generated tokens. There are no masked tokens in this process; instead, the model learns to create coherent and contextually relevant sentences based on its training. This method effectively teaches the model to produce new language outputs by leveraging both visual and linguistic information [31].

Finally, **Prefix-LM** introduces a hybrid strategy that combines elements of MLM and autoregressive generation. In this approach, the input sentence is

Model	Dual or Fusion Encoder	Multimodal Fusion	Having Decoder
ViLBERT [39]	Fusion Encoder (Two-Stage)	Co-Attention	No
LXMERT [40]			
VisualBERT [41]		Merged Attention	
VL-BERT [42]			
UNITER [43]			
OSCAR [44]			
VILLA [45]			
VinVL [46]			
UNIMO [47]			
VL-T5 [48]	Fusion Encoder (End-to-End)		Yes
SimVLM [12]			
MDETR [49]			
UniTAB [50]			
OFA [51]			
PixelBERT [52]			No
SOHO [53]			
CLIP-ViL [54]			
ViLT [55]			
Visual Parsing [56]			
GIT [57]			
VLMo [58]			
BEiT-3 [59]			
Flamingo [3]		Co-Attention (Only Image-to-Text)	
ALBEF [60]			
BLIP [7]			
CoCa [15]			
METER [4]		Co-Attention	
FIBER [61]			
CLIP [10]	Dual Encoder	Cosine Similarity	
ALIGN [5]			

Table 2.1: Classification of VLPs based on training procedure, multimodal fusion, and having decoder. Based on [31].

split into two parts: a prefix and a remaining sequence. The model utilizes bi-directional attention for the prefix, allowing it to incorporate context from both the prefix and the accompanying image. However, when generating the remaining tokens, it adopts a causal attention mechanism, ensuring that each

word is predicted based only on the preceding words. This innovative structure enhances the model's ability to understand and generate natural language in a way that reflects realistic writing and speaking patterns [31].

- Image-Text Matching (ITM): Involves determining whether a given image and caption correspond to each other. The model is trained to analyze pairs of images and captions, some of which are correct matches and others are not, and then predict which pairs go together. Many VLPs approach this by treating the task as a simple classification problem, where the model decides whether the pair is a match or not. To help the model understand both the image and the text together, a special token (*[CLS]*) is added to the input, which allows the model to capture a joint representation. During training, the model is presented with both matching and non-matching pairs and learns to predict whether they are correct matches. The model assigns a score to each pair, indicating the likelihood of a match (*p(y)*), and this score is optimized using *cross-entropy* loss [31]:

$$\mathcal{L}_{ITM}(\theta) = -\mathbb{E}_{(\tilde{w},\tilde{v})\sim D}[y \log p(y) + (1-y)\log(1-p(y))] \tag{2.2}$$

- Image-Text Contrastive Learning (ITC): Given a batch of N pairs of image-text, ITC aims to predict the N matched pairs from all possible ones (N^2). It is worth mentioning that ITC is used on top of the embeddings generated by the image and text encoders, before the multimodal fusion (i.e., the use of (w, v) instead of $(\tilde{w}, \tilde{v})$). LTC is usually the average of image-to-text ($\mathcal{L}^{i2t}_{LTC}$) and text-to-image ($\mathcal{L}^{t2i}_{LTC}$) contrastive losses [31]:

$$\begin{aligned} s^{i2t}_{i,j} &= v_i^T w_j, s^{t2i}_{i,j} = w_i^T v_j \\ \mathcal{L}^{i2t}_{LTC} &= -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp(s^{i2t}_{i,i}/\sigma)}{\sum_{j=1}^{N}\exp(s^{i2t}_{i,j}/\sigma)} \\ \mathcal{L}^{t2i}_{LTC} &= -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp(s^{t2i}_{i,i}/\sigma)}{\sum_{j=1}^{N}\exp(s^{t2i}_{i,j}/\sigma)} \end{aligned} \tag{2.3}$$

  where σ is a learned temperature hyper-parameter. Also, $\{v\}_{i=1}^{N}$ and $\{w\}_{i=1}^{N}$ indicate the normalized image vectors and text vectors in a batch, respectively.

- Masked Image Modeling (MIM): Given an image-text pair $(\tilde{v}, \tilde{w})$, some parts of the image (e.g. m) are randomly masked ($\tilde{v}_m$) and then it is tried to predict them based on the text ($\tilde{w}$) and the remaining parts of the image ($\tilde{v}_{\bar{m}}$) by minimizing the negative log-likelihood [31]:

$$\mathcal{L}_{MLM}(\theta) = -\mathbb{E}_{(\tilde{w},\tilde{v})\sim D} \log P_\theta(\tilde{v}_m|\tilde{v}_{\bar{m}}, \tilde{w}) \tag{2.4}$$

There are different approaches for masking parts of an image in VLPs. Models that rely on object detectors for image feature extraction divide the image into regions, and the features from some random regions are masked by replacing them with zeros. These models are then trained to reconstruct the original features by minimizing the mean squared error. In some cases, labels are generated for the regions, and the models are trained to predict the labels rather than reconstruct the original region features [31].

On the other hand, end-to-end VLP models work with image patches for masking. For example, DALL-E [62] uses VQ-VAE [63] to convert image patches into discrete tokens. First, the image is divided into patches, and each patch is assigned a discrete token. Some tokens are randomly masked, and the model is trained to predict the missing tokens. Additionally, some models use in-batch negatives for masked patch reconstruction. In this approach, each image in a batch is divided into patches, creating a pool of candidate patches. The model is trained to select the correct patches for each masked region from this candidate pool [31].

In this thesis, the retrieval task is addressed through the use of Dual Encoder architectures. Specifically, well-established CLIP-like architectures are utilized, where contrastive pre-training is employed to align visual and textual representations in a shared embedding space.

2.4 CLIP model

Contrastive Language-Image Pre-training (CLIP) [10] is a dual-encoder vision-language model that leverages image-text contrastive learning (ITC) to maximize the similarity between an image and its corresponding caption while minimizing the similarity between unrelated images and captions. Given N image-text pairs, CLIP constructs an $N \times N$ matrix, where *(i,j)-th* element represents the similarity between the *i-th* image and the *j-th* caption. The model is trained to identify true image-text pairs from this matrix. CLIP jointly trains image and text encoders to learn a shared image-text embedding space. It uses a modified Transformer [36] network as the text encoder and supports different variants of ResNet [37] or Vision-Transformer [38] as the image encoder. Each encoder is equipped with a linear projection head that maps image and text representations into the shared embedding space. Notably, the embeddings of both images and text are l_2-normalized confining the joint embedding space to a unit hypersphere.

CLIP has redefined the possibilities of AI by demonstrating exceptional performance in a vast array of visual tasks. Its robust embedding space not only aligns text and image modalities seamlessly but also empowers it to excel in diverse downstream applications. From accurately classifying images into unseen categories

[64] and retrieving images based on textual descriptions [65] to precisely segmenting objects [66] within scenes, CLIP's versatility is truly remarkable. It has even ventured into the realm of object detection [67], video understanding [68], depth estimation [69], and creative tasks like image captioning [70] and visual question answering [54]. This groundbreaking model has reshaped the landscape of computer vision, opening up new avenues for innovation and problem-solving.

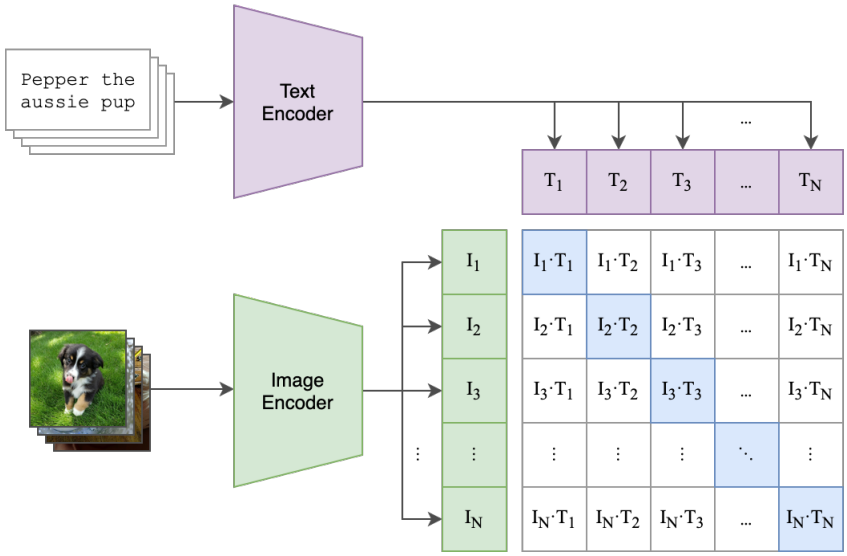


Figure 2.6: Contrastive Pre-Training. Adopted from [10]

2.5 From Dual-Encoders to Unified-Encoder

The human brain, a key inspiration for neural networks, exhibits remarkable parallel processing capabilities. It seamlessly integrates information from various senses (vision, hearing, touch, etc.) concurrently. Moreover, knowledge acquired in one domain can significantly improve understanding in others. In contrast, current deep-learning perception models are constrained by inductive biases and specific assumptions. They are typically designed for individual modalities and lack generalizability across different sensory inputs. For instance, image processing models effectively leverage the 2D grid structure of images, employing 2D convolutional operations. However, this specialized architecture cannot be readily applied to process text data, which exhibits a fundamentally different structure.

Recent advancements have seen the emergence of models capable of processing diverse modalities with a unified approach. These models typically involve a preprocessing step to adapt data from different sources (vision, text, audio) into

a suitable format, followed by a shared architecture for representation learning. For example, Perceiver [71] initially converts data into a byte array through a preprocessing step. Subsequently, a Transformer-like [36] architecture learns a latent representation of this processed data. Data2Vec [72] utilizes a modality-specific preprocessing step (such as converting images into a sequence of patches) and then employs a teacher-student framework based on Transformers [36]. In this framework, a teacher model generates representations of the complete input, while a student model learns to predict these representations using a masked portion of the input. Meta-Transformer [73] also adopts a modality-specific data-to-sequence tokenization step. This tokenized data is then fed into a shared Transformer-based [36] encoder for learning a latent representation across different modalities.

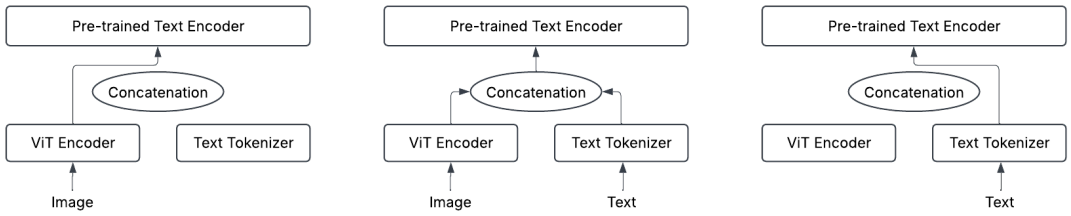
Vision-Language Pre-trained models (VLPs) typically employ separate encoders for vision and text, often with an additional encoder for multimodal fusion (for details, see Section 2.3.1). This conventional architecture presents two key limitations. Firstly, it necessitates extensive training data. For instance, CLIP [10] was trained on 400 million image-text pairs, while ALIGN [5] and SimVLM [12] required approximately 1 billion pairs. Secondly, these models often exhibit a substantial number of parameters. FLAVA [11], for example, has 437 million parameters, while FLAMINGO [3] boasts 4.6 billion. These limitations result in high computational resource demands and prolonged training times. Recent research has explored the use of unified encoders to address these challenges. MoMo [24], for instance, converts both text and images into tokens and patches, subsequently feeding them into a shared encoder. It incorporates a unique three-stage training procedure, enabling it to learn representations from text-only, image-only, and combined text-image inputs. VISTA [25] represents another approach, utilizing a unified encoder to process text tokens alongside hidden states extracted from image tokens. This architecture allows for the generation of representations for text-only, image-only, and text-image inputs.

2.5.1 VISTA

The VISTA model is designed to integrate a Vision Transformer (ViT) [38] encoder as an image tokenizer into a pre-trained text encoder, enabling it to process text, images, and multi-modal data efficiently. The text encoder, based on a pre-trained BERT model, serves as the foundation for text embedding and remains frozen during the training process to maintain its strong text retrieval capabilities. Meanwhile, ViT acts as an image tokenizer, converting images into sequences of visual tokens, which are then fed into the text encoder. This design allows the model to incorporate image representations while preserving the structure of text embeddings.

The encoding process in VISTA follows a structured approach for different types

of data. For text encoding, the model directly tokenizes a given text sequence and passes it through the pre-trained BERT encoder, generating a corresponding text embedding (Figure 2.7c). For image encoding, the input image is divided into patches, which are processed by the ViT encoder to produce image tokens. These tokens are then fed into the text encoder, ensuring compatibility between text and image representations (Figure 2.7a). When encoding composed image-text data, the image tokens and text tokens are concatenated into a single interleaved sequence and jointly processed by the text encoder, producing a unified multi-modal embedding (Figure 2.7b). This method ensures seamless integration of text and visual information within a common embedding space.



(a) The input is just image **(b)** The input is image-text pair **(c)** The input is just text

Figure 2.7: The way that the VISTA [25] encodes the input. It is worth mentioning that ViT [38] operates as a tokenizer for the image

Chapter 3

Modality Gap

In this chapter, the concept of the modality gap in vision-language models is explored, with a focus on its causes, consequences, and implications. The causes of the modality gap are discussed, stemming from fundamental differences in how visual and linguistic data are represented in a shared latent space. Its consequences are then examined, emphasizing the impacts it creates for downstream tasks such as retrieval, classification, and vector arithmetic. Through this discussion, a deeper understanding of the modality gap and its significance in multimodal learning is provided.

3.1 What is modality gap

The modality gap refers to the significant disparity in the way that vision language models (VLMs) represent information from different modalities (images and text). Essentially, the models struggle to align the semantic meanings of visual and textual data within a shared embedding space. Visual and textual representations often occupy separate clusters or regions within the model's embedding space, hindering direct comparisons and cross-modal interactions [1, 18, 17, 23, 22].

To demonstrate the modality gap, the embeddings of images and their corresponding captions from the MSCOCO [74] validation set, generated by various models are visualized in Figure 3.1 using UMAP [75]. The models include:

- ALBEF: it employs a dual-encoder architecture that aligns image and text representations before fusing them. It utilizes three main pre-training objectives including an image-text contrastive loss to ensure alignment, a matching loss for better representation of paired data, and a masked language modeling task to enhance text understanding [60].
- ALIGN: it features a dual-encoder setup consisting of EfficientNet [76] for

image processing and BERT [26] for text representation. It is trained on a massive dataset of noisy image-text pairs, leveraging self-supervised learning. The primary pre-training objective focuses on contrastive learning, which allows the model to learn robust embeddings [5].

- CLIP: it is a dual-encoder vision-language model that leverages image-text contrastive learning to maximize the similarity between an image and its corresponding caption (more details are provided in Section 2.4).

- CyCLIP: it enhances the original CLIP [10] model by introducing cyclic contrastive learning. Its architecture includes separate encoders for images and text, similar to CLIP, but adds cycle consistency constraints that enforce geometric alignment between embeddings across modalities [77].

- FLAVA: it follows a tri-branch transformer architecture integrating both unimodal and multimodal encoders—specifically Vision Transformers [38] for images and BERT [26] for text. Its pre-training objectives are diverse, combining contrastive learning for alignment and generative tasks to enhance understanding across modalities [11].

- ImageBind: it is designed to create a shared embedding space for six modalities: image, text, audio, depth, thermal, and IMU data. Its architecture revolves around using images as a pivot for binding these diverse modalities together. The pre-training objective focuses on cross-modal retrieval without requiring paired data beyond images [78].

- VISTA: The VISTA model is designed to integrate a Vision Transformer (ViT) [38] encoder as an image tokenizer into a pre-trained text encoder, enabling it to process text, images, and multi-modal data efficiently (more details are provided in Section 2.5.1).

Image and text embeddings are found to be distinctly separated in the shared multi-modal space. This finding is unexpected given the typical behavior of VLMs trained with contrastive loss. Contrastive loss aims to bring embeddings of paired images and texts closer together in the embedding space, while simultaneously pushing embeddings of different image-text pairs further apart. However, the observed behavior deviates from this expected pattern.

Moreover, Figure 3.2 highlights the modality gap. Since MSCOCO [74] does not provide positive examples for images, the images are grouped into classes based on shared objects—i.e., images containing the same objects are placed in the same class. This process results in 438 classes, each containing at least two images. Two images and their corresponding captions are selected from each class, and the cosine similarities among text-text, image-image, and image-text pairs

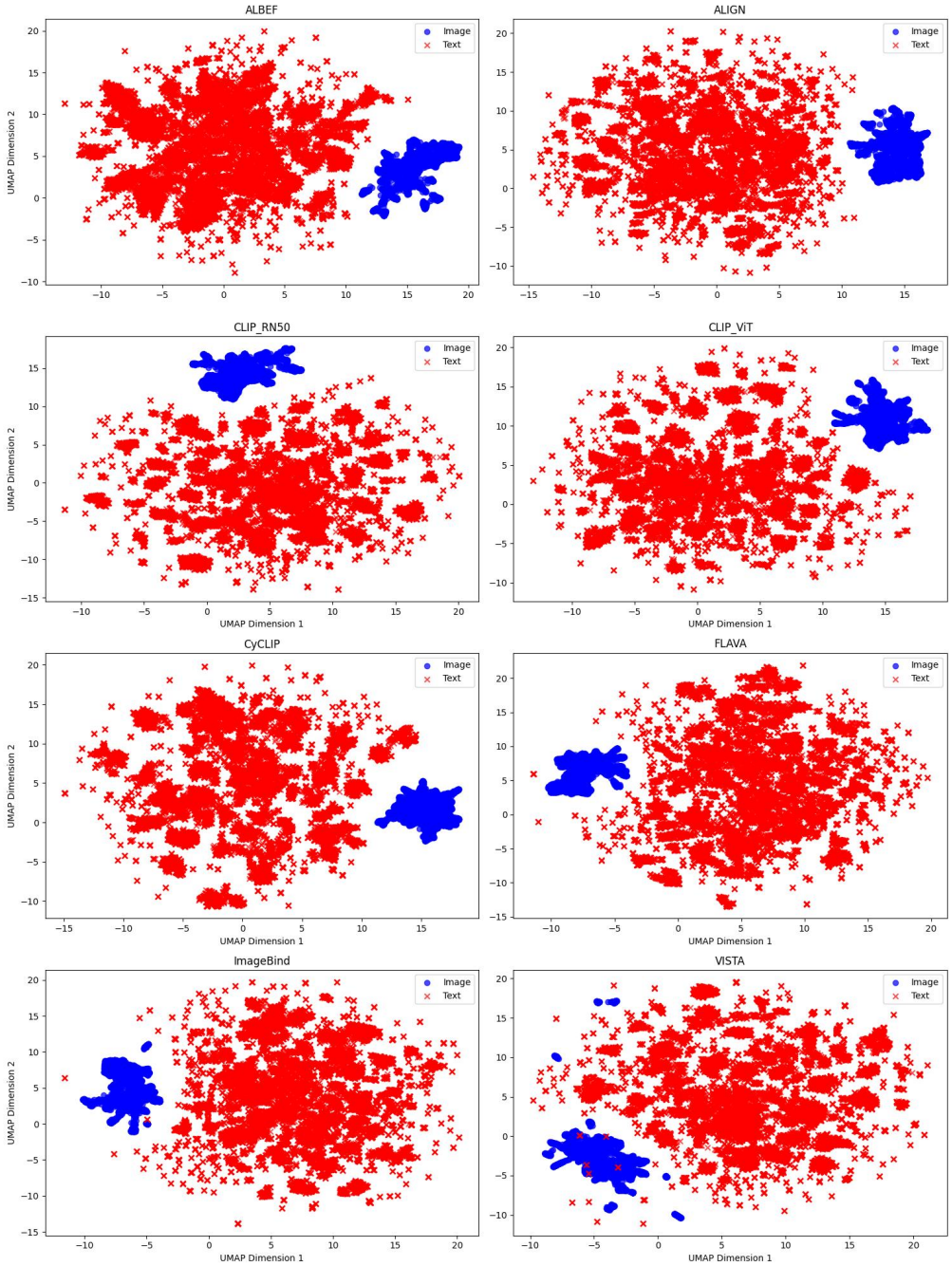


Figure 3.1: UMAP [75] visualization of generated embeddings from pre-trained models on MSCOCO [74] validation split.

are analyzed. Despite the models being explicitly trained to maximize paired

image-text cosine similarities, the image-text similarities remain significantly lower than the text-text and image-image similarities (VISTA [25] is an exception). This observation indicates that the image and text embeddings reside in distinct regions of the shared multi-modal embedding space.

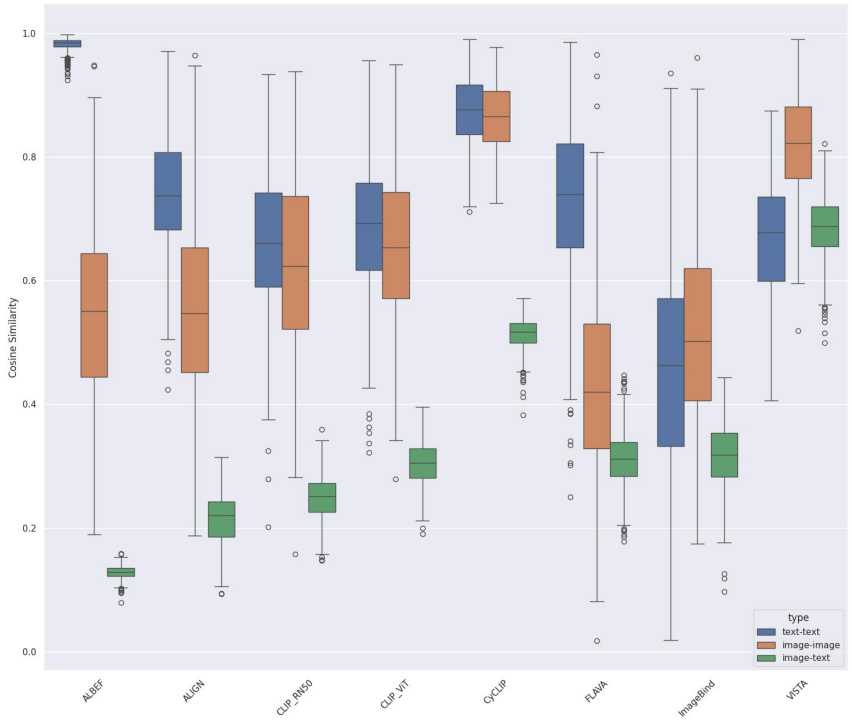


Figure 3.2: Visualization of cosine similarities of the generated embeddings from pre-trained models on MSCOCO [74] validation split.

In addition to the charts, quantitative metrics can also be used to measure the modality gap. One of the metrics introduced by [17] defines the modality gap as the difference between the center of image embeddings and text embeddings

$$\frac{1}{n}\sum_{i=1}^{n} x_i - \frac{1}{n}\sum_{i=1}^{n} y_i \tag{3.1}$$

where x_i and y_i are the L2-normalized image embedding and text embedding. This metric is named *Centroids Difference (CD)* in the rest of the thesis.

Another metric is *Central Moment Discrepancy (CMD)* [79], which measures the distributional difference between two feature sets by comparing their higher-order

moments. CMD is widely used in domain adaptation, where the objective is to align the feature distributions of a source domain and a target domain. It iteratively computes discrepancies for each moment, starting from the first moment (mean) and proceeding to higher-order moments such as variance, skewness, and kurtosis. Formally, given two feature sets—source *(S)* and target *(T)*—CMD calculates the difference between their corresponding central moments up to a specified order and sums these differences to obtain the final value.

$$\sum_{n=1}^{k} \|\mu^n(S) - \mu^n(T)\|_2 \tag{3.2}$$

where μ^n indicates the *n-th* order central moment. By summing the discrepancies across multiple moments, CMD captures differences in both the central tendency and the shape of the distributions.

In vision-language models, the embeddings of images and text are mapped into a shared latent space. There is usually a modality gap in the latent space, and the resulting distributions of image embeddings and text embeddings often do not align perfectly. CMD can be adapted to quantify the modality gap by treating the image embeddings as one domain and the text embeddings as another. CMD measures the discrepancy between the distributions of the embedding sets, providing a metric for how well-aligned the two modalities are in the shared space.

The two metrics are computed for some pre-trained models based on their generated image and text embeddings on the validation split of MSCOCO [74] and are shown in Table 3.1.

Model	**CMD**	**CD**
ALBEF	1.1	1.05
ALIGN	0.98	0.96
CLIP_RN50	0.87	0.82
CLIP_ViT	0.86	0.82
CyCLIP	0.87	0.87
FLAVA	0.83	0.81
ImageBind	0.71	0.7
VISTA	0.44	0.44

Table 3.1: Central Moment Discrepancy (CMD) and Centroids Difference (CD) computed for some pre-trained models based on their generated image and text embeddings on the validation split of MSCOCO [74]

3.2 Causes of modality gap

The causes of the modality gap have been a subject of considerable debate in the literature. According to [17], the modality gap exists at the model's initialization due to the *Narrow Cone Effect*, and the contrastive loss commonly employed by vision-language models maintains this gap throughout training. The narrow cone effect occurs because the encoder's effective embedding space is confined to a cone-shaped subregion of the overall embedding space, a consequence of the model's random initialization and non-linear activation functions. Furthermore, [17] identifies two key factors influencing the modality gap in relation to contrastive loss: temperature settings and data mismatches. High temperatures facilitate a more effective reduction of the gap, whereas low temperatures, combined with data mismatches, contribute to sustaining the gap. While [1] supports the notion that higher temperatures reduce the modality gap, it argues that the impact of data mismatches does not hold under all circumstances.

In contrast, [18] attributes the modality gap primarily to the contrastive loss function's dual objectives: alignment and uniformity. The alignment objective aims to increase the similarity between paired image-text embeddings by bringing them closer in the latent space, whereas the uniformity objective pushes apart negative pairs from different modalities without ensuring consistent spacing within each modality. The inherent tension between these two goals creates local minima, leading to the separation of modalities. Their experiments confirm that the contrastive loss naturally induces such local minima, making it difficult for the model to bridge the gap during training.

On a different note, [16] contends that even if the modality gap does not exist at initialization, contrastive loss can still induce it. They propose modifying the contrastive loss by incorporating alignment and uniformity properties from unimodal contrastive frameworks into the multimodal setting. By adding these enhanced terms, the embeddings are distributed more evenly across the representational space, effectively mitigating the modality gap. Additionally, [31] suggests that achieving high alignment and uniformity is essential for optimal loss function performance.

3.3 Consequences of modality gap

Intuitively, it might be expected that minimizing the distance between image and text embeddings (the modality gap) in multimodal models would enhance performance on downstream tasks. However, research has shown a more nuanced picture. While modifying the modality gap can improve performance on downstream tasks (e.g. classification), as demonstrated by [17], the optimal gap size and the direction of change are not always clear. [19] observed that CLIP's embedding

space displays a noticeable separation between image and text representations, resulting in suboptimal alignment and uniformity. Although the study does not explicitly address the modality gap, it demonstrates that improving alignment and uniformity enhances the model's capacity to transfer knowledge across modalities, thereby boosting performance in downstream tasks such as retrieval, classification, and captioning. Further, [16] discusses how the modality gap negatively impacts downstream tasks like image classification, retrieval, and vector arithmetic [20]. Meanwhile, [21] highlights that reducing the modality gap can enhance cross-modal transferability. However, several studies, including [22] and [1] have highlighted that simply closing the modality gap may not consistently improve downstream performance (e.g. classification, retrieval, and vector arithmetic) and can even be detrimental in certain cases. These findings suggest that the relationship between the modality gap and downstream performance is complex and requires further investigation.

Figure 3.1 illustrates that VISTA [25], as a single-encoder architecture with shared parameters across modalities, exhibits distinct behavior. Meanwhile, Figure 3.2 shows that the cosine similarities between image-text pairs in VISTA are nearly identical to those of text-text pairs. Additionally, Table 3.1 indicates that VISTA demonstrates a smaller modality gap compared to dual-encoder models. These observations suggest that modality gap reduction may have different implications for shared-encoder architectures than for dual-encoder ones. Therefore, the extent to which modality gap reduction benefits these two types of vision-language pretraining (VLP) models is analyzed in this thesis, with a particular focus on cross-modal retrieval and vector arithmetic [20].

Chapter 4

Methodology

In this chapter, the methodology designed to address the modality gap in vision-language models is presented, with a focus on its impact on downstream tasks. Building upon prior works, where specific loss functions [16] or fine-tuning strategies [80] were explored individually on dual encoder architecture i.e. CLIP [10], an innovative combination of these techniques is proposed and applied on the unified (shared-encoder) architecture. Specifically, tailored loss functions are integrated with fine-tuning strategies and applied to VISTA [25]. The impact of combining these approaches on the modality gap, and downstream tasks, including retrieval, and vector arithmetic [20], is evaluated. Additionally, to ensure a fair comparison between dual-encoder and shared-encoder architectures, the same pipeline is applied to CLIP [10].

4.1 Tailored Loss Functions

Uniformity refers to the property of embeddings being evenly distributed across the contrastive latent space. In contrast, alignment describes the closeness of positive pairs within the latent space [16].

The authors of [16] conducted a controlled experiment to indicate the modality gap arises not from differences between modalities but as an inherent consequence of contrastive learning itself. In another experiment, they reduced CLIP's embedding space to three dimensions to visualize how embeddings evolve during training. Initially, the embeddings form distinct clusters within separate cones. As training progresses, they transition into arcs, then rings, and eventually disperse across the sphere. The authors argue that in higher-dimensional spaces, embeddings fail to distribute uniformly and remain misaligned, contributing to the contrastive gap.

To address this, the authors fine-tune CLIP by incorporating explicit uniformity and alignment terms into the contrastive loss. Their results demonstrate that this

fine-tuning significantly mitigates the contrastive gap.

According to [81], uniformity and alignment are desirable characteristics in unimodal contrastive representational spaces, and models exhibiting high uniformity and alignment tend to perform better on downstream tasks. [16] extends the concepts of uniformity and alignment to the multimodal contrastive space. Specifically, it enforces uniformity through two terms: in-modality uniformity ($L_{Uniform}$) and cross-modality uniformity ($L_{XUniform}$).

The in-modality uniformity term encourages uniformity among embeddings within each modality, while the cross-modality uniformity term enforces uniformity among negative image and text samples. The in-modality uniformity loss is defined as follows:

$$L_{\text{Uniform}} = \frac{1}{2}(L^I_{\text{Uniform}} + L^T_{\text{Uniform}}) \tag{4.1}$$

where L^I_{Uniform} promotes uniformity among image embeddings and is given by:

$$L^I_{\text{Uniform}} = \log\left(\frac{1}{N}\sum_{j=1}^{N}\sum_{k=1}^{N}\exp\left(-2\|E^I_j - E^I_k\|^2\right)\right)$$

Where $E_j{}^I$ is the image embedding of the j^{th} pair and N indicate the size of a batch. A similar term can be defined for text embeddings. The cross-modality uniformity term is expressed as:

$$L_{\text{XUniform}} = \log\left(\frac{1}{N}\sum_{j=1}^{N}\sum_{\substack{k=1\\k\neq j}}^{N}\exp\left(-2\left\|E^I_j - E^T_k\right\|^2\right)\right) \tag{4.2}$$

Where $E_k{}^T$ is the text embedding of the k^{th} pair. Additionally, [16] introduces a term to improve the alignment of positive image-text pairs:

$$L_{\text{Align}} = \frac{1}{N}\sum_{j=1}^{N}\left(\left\|E^I_j - E^T_j\right\|^2\right) \tag{4.3}$$

Using the three terms, [16] proposes two new loss functions:

$$L_{CUA} = L_{CLIP} + L_{Uniform} + L_{Align} \tag{4.4}$$

$$L_{CUAXU} = L_{CUA} + L_{XUniform} \tag{4.5}$$

Here, L_{CLIP} corresponds to the same objective defined in Formula 2.3. In this thesis, these two loss functions are employed along with fine-tuning strategies (explained in Section 4.2) and applied on the unified architecture to examine their effects on the shared embedding space and, consequently, on downstream tasks.

4.2 Fine-Tuning Strategies

A common strategy for learning image embeddings involves leveraging a large, well-curated dataset of labeled datasets. The combination of extensive scale and high-quality annotations enables the development of state-of-the-art image embeddings. However, this approach has a fundamental limitation: it is restricted to a predefined set of categories, meaning the resulting models can only reason about those specific classes.

In contrast, image-text data does not suffer from this constraint, as it learns from free-form text that encompasses a much wider range of real-world concepts. Nonetheless, the quality of available image-text data may be lower for training image embeddings compared to meticulously curated datasets.

[80] introduce *contrastive tuning* to leverage both labeled image data and image-text data by initializing contrastive pre-training with an image model already trained on cleaner, labeled data. This allows image-text alignment to be learned independently of image embeddings, benefiting from both data sources. The approach is flexible enough to integrate various pre-trained models, including self-supervised ones, to produce meaningful representations. A similar strategy can also be applied to text encoders, utilizing powerful pre-trained models that rely on text-specific data and learning techniques.

Contrastive tuning requires several key design decisions. First, the model components handling different modalities (e.g., image and text) can either be randomly initialized or derived from a pre-trained model. For pre-trained models, there are at least two main approaches: keeping them entirely frozen or allowing full fine-tuning. Additionally, various intermediate strategies exist, such as selectively freezing specific layers or applying customized learning rates.

Following [80], this thesis adopts a two-character notation to represent design choices. Each character corresponds to the configuration of the image model and text model, respectively. There are two settings: L (locked/frozen weights, initialized from a pre-trained model) and U (unlocked/trainable weights, also initialized from a pre-trained model).

For CLIP [10], four distinct fine-tuning strategies are explored:

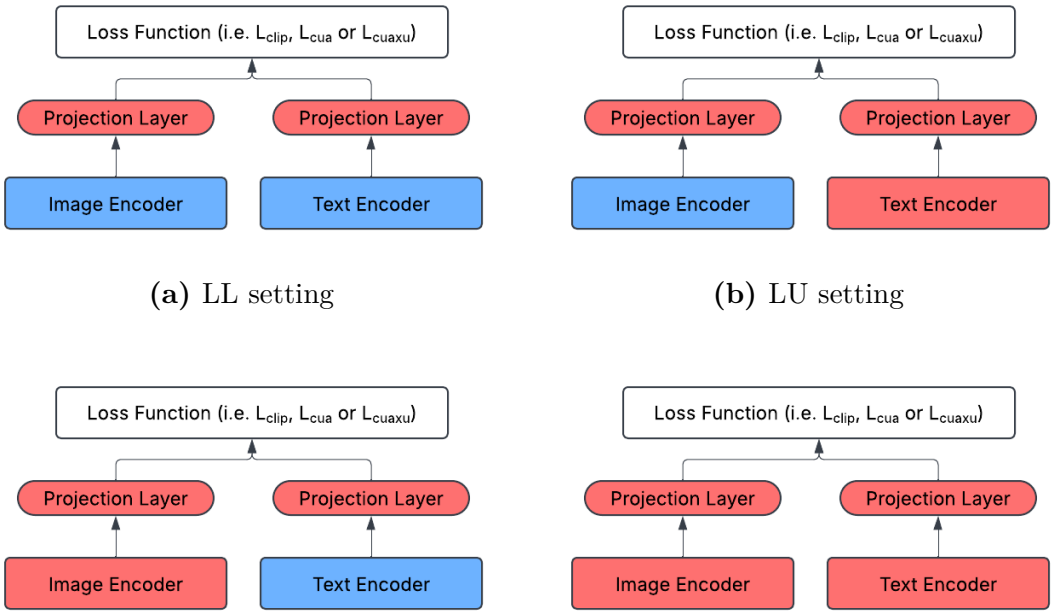
1. LL: Both the image and text encoders remain frozen (Figure 4.1a).

2. LU: The image encoder is frozen, while the text encoder is fine-tuned (Figure 4.1b).

3. UL: The text encoder remains frozen, whereas the image encoder is fine-tuned (Figure 4.1c).

4. UU: Both the image and text encoders are fine-tuned (Figure 4.1d).

In all configurations, models are initialized with pre-trained weights, and the projection layers are always fine-tuned.

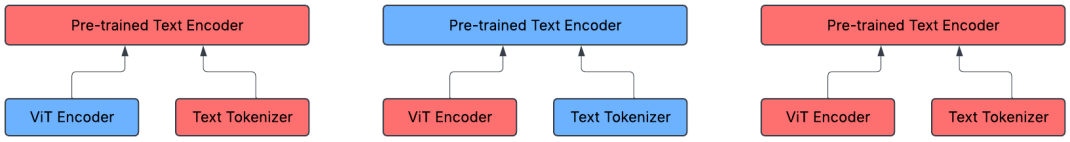


(a) LL setting **(b)** LU setting

(c) UL setting **(d)** UU setting

Figure 4.1: Different settings for the fine-tuning of the CLIP [10]. The blue color means the component is **L**ocked (frozen), while the red one means the component is (**U**nlocked) going to be fine-tuned.

Similarly, fine-tuning settings are defined for VISTA [25]. However, the LL configuration is not applicable, as the architecture lacks projection layers (Figure 4.2).



(a) LU setting **(b)** UL setting **(c)** UU setting

Figure 4.2: Different settings for the fine-tuning of the VISTA [25]. The blue color means the component is **L**ocked (frozen), while the red one means the component is (**U**nlocked) going to be fine-tuned.

Chapter 5

Experiments

The experiments conducted in this work aim to evaluate the impact of the modality gap in shared-encoder architectures across downstream tasks, including retrieval and vector arithmetic, using well-established datasets. This chapter provides an overview of the downstream tasks, the datasets employed, the experimental setup, and the results obtained.

5.1 Downstream Tasks

This thesis evaluates and compares the impact of different fine-tuning strategies and loss functions on dual-encoder and shared-encoder architectures through observing their performances on two downstream tasks: image retrieval and vector arithmetic.

- Image Retrieval: is a fundamental vision-language task that assesses the alignment between visual and textual representations. This study considers both image-to-text and text-to-image retrieval scenarios, evaluating retrieval performance at ranks 1, 5, and 10.

- Vector Arithmetic in Multimodal Embeddings: applies textual transformations to image embeddings for retrieval by adding a delta vector derived from word differences (e.g., cat to dog). Key parameters include λ for scaling the delta vector and N for the number of nearest neighbors in retrieval, both set to 1 in the experiments.

5.2 Datasets

In this thesis, three datasets—Microsoft Common Objects in Context (MSCOCO) [74], Conceptual Captions [82], and SIMAT [20]—were utilized for experimental

purposes. The first two datasets supported contrastive fine-tuning and image retrieval tasks, whereas SIMAT was solely employed for vector arithmetic analysis.

- MSCOCO: is a large-scale dataset designed for multiple computer vision applications, including object detection, segmentation, and image captioning. Each image in this dataset is accompanied by five descriptive captions. The 2017 version of MSCOCO was used in this study.
- Conceptual Captions: consists of over 3.3 million image-caption pairs sourced from the web. Unlike MSCOCO, where captions are manually annotated, this dataset features automatically generated captions derived from image metadata, with a single caption assigned to each image. Due to its scale and diverse linguistic variations, it is particularly beneficial for large-scale vision-language pre-training. For this research, a random subset of 100,000 images with their corresponding captions was selected as a benchmark dataset.
- SIMAT: is specifically designed for examining vector arithmetic in multimodal embeddings. It provides a structured collection of image and text embeddings, facilitating an in-depth study of relationships between these representations through algebraic operations. This dataset enables the investigation of how the modality gap influences vector arithmetic and allows for the evaluation of different loss functions in improving cross-modal consistency. By leveraging SIMAT, this thesis explores whether enhanced alignment leads to more semantically meaningful vector transformations within multimodal spaces.

5.3 Experimental Setup

In all experiments, the ViT-B/32 version of CLIP [10] and the BAAI/bge-base-en-v1.5 version of VISTA (with mean pooling) were fine-tuned for 10 epochs with a batch size of 128 using the AdamW optimizer [83] (learning rate = 5×10^{-5}, eps = 1×10^{-8}, and weight decay = 0.1) and a cosine learning rate scheduler [84]. The experiment code was implemented in PyTorch, and all experiments were conducted on an NVIDIA RTX 2080 Ti GPU.

5.4 Results

This section examines the impact of fine-tuning strategies and loss functions on downstream task performance. First, it analyzes the effectiveness of various strategies and loss functions in enhancing retrieval performance and reducing the modality gap. Next, it explores the relationship between retrieval performance and the modality gap. Additionally, a similar analysis is conducted to investigate the relationship between the modality gap and performance on the vector arithmetic.

5.4.1 Retrieval and Modality Gap

The retrieval results of fine-tuned models with L_{clip} using different strategies are reported in Table 5.1. The findings indicate that **LU** is the most effective strategy when combined with L_{clip} for improving retrieval performance. The difference between LL and LU in CLIP is minimal in terms of T→I (R@1 and R@5) on the Conceptual Captions, making their performance nearly equivalent.

This is particularly interesting because [80] previously suggested that locking the image encoder while tuning the text encoder is the best strategy for CLIP (a dual-encoder architecture). Here, it is observed that the same strategy is also optimal for VISTA (a shared-encoder architecture).

Another notable finding is that in the text-to-image zero-shot setting, VISTA slightly outperforms CLIP (except in R@10 on MSCOCO), whereas CLIP demonstrates significantly better performance in the image-to-text zero-shot setting. This discrepancy can be attributed to VISTA's use of a text encoder as its backbone, which may be less effective at processing images compared to CLIP's more robust visual encoder, potentially leading to difficulties in capturing all relevant features.

A similar analysis was conducted for L_{CUA} and L_{CUAXU}, with results provided in Appendix A. The results for L_{clip} are presented here because models fine-tuned with this loss function outperform those fine-tuned with other loss functions.

To study the effects of fine-tuning on the modality gap and its relationship with retrieval performance, the retrieval performance of the LU strategy (identified as the best strategy) across different loss functions is compared to the modality gap in Table 5.2. The results indicate that L_{CUA} and L_{CUAXU} are both highly effective in reducing the modality gap. Models fine-tuned using these two loss functions exhibit a significantly lower modality gap compared to those fine-tuned with L_{clip}, with L_{CUA} consistently achieving the minimal gap. However, the best retrieval performance is obtained with L_{clip}. This finding suggests that reducing the modality gap and retrieval performance are inversely related, regardless of the architecture. One possible explanation for this inverse relationship is that modality-specific features in the image and text embeddings contribute to retrieval performance. When the modality gap is reduced by bringing image and text embeddings closer together, these modality-specific features may be eliminated, potentially affecting retrieval effectiveness.

5.4.2 Vector Arithmetic and Modality Gap

The SIMAT scores of fine-tuned models using different loss functions and strategies and their corresponding modality gaps are presented in Table 5.3 and 5.4, respectively.

The findings reveal a positive correlation between reducing the modality gap and improving SIMAT scores. Notably, L_{CUA} and L_{CUAXU} effectively minimize the

		MSCOCO			Conceptual Captions		
		R@1	R@5	R@10	R@1	R@5	R@10
Text to Image							
CLIP	LL	0.4038	0.6797	0.7860	**0.3895**	**0.6374**	0.7276
	LU	**0.4175**	**0.7012**	**0.8030**	0.3856	0.6349	**0.7322**
	UL	0.3799	0.6603	0.7680	0.3028	0.5419	0.6393
	UU	0.3910	0.6816	0.7898	0.3012	0.5466	0.6449
	ZS*	0.3082	0.5504	0.6620	0.3255	0.5527	0.6454
VISTA	LU	**0.4725**	**0.7482**	**0.8360**	**0.4273**	**0.6676**	**0.7460**
	UL	0.4001	0.6852	0.7885	0.2910	0.5283	0.6216
	UU	0.3954	0.6869	0.7954	0.2906	0.5240	0.6254
	ZS*	0.3201	0.5743	0.6048	0.3547	0.5673	0.65
Image to Text							
CLIP	LL	0.5334	0.7348	0.8168	0.3937	0.6370	0.7265
	LU	**0.5608**	**0.7696**	**0.8490**	**0.3954**	**0.6502**	**0.7392**
	UL	0.4682	0.6888	0.7896	0.3024	0.5391	0.6357
	UU	0.5068	0.7144	0.8094	0.3014	0.5469	0.6505
	ZS*	0.4674	0.6662	0.7554	0.3517	0.5771	0.6688
VISTA	LU	**0.5842**	**0.7816**	**0.8608**	**0.4043**	**0.6480**	**0.7339**
	UL	0.4734	0.6938	0.7948	0.2754	0.5025	0.6043
	UU	0.4954	0.7188	0.8180	0.2727	0.5161	0.6161
	ZS*	0.3054	0.495	0.6048	0.2938	0.5014	0.5820

Table 5.1: Retrieval results of CLIP [10] and VISTA [25] on MSCOCO [74] and Conceptual Captions [82] datasets. Fine-tuned using L_{clip}. ZS* stands for Zero-Shot, indicating that it has not been fine-tuned. The best performance of each model is bold.

gap while enhancing the SIMAT score, particularly for CLIP, which improves from a zero-shot score of approximately 16 to around 45. This improvement may be attributed to the inherently high modality gap in CLIP due to its dual-encoder architecture (see Appendix B.1). Thus, reducing the gap significantly improves its SIMAT score.

In most strategies (except for UL of VISTA on Conceptual Captions), the best score and gap are achieved with these two loss functions. However, the highest score and the smallest gap do not coincide, indicating a limit to gap reduction. For instance, CLIP's minimum gap of 0.02 (LU strategy on Conceptual Captions) results in a score of 36.7420, significantly lower than its highest score of 45.4611 (UU strategy on MSCOCO). This disparity is even more notable for VISTA, where the minimum gap of 0.05 (UU strategy on Conceptual Captions) corresponds to

		MSCOCO			
		L_{clip}	L_{CUA}	L_{CUAXU}	ZS^*
CLIP (LU)	T→I@1	**0.4175**	0.3830	0.3954	0.3082
	I→T@1	**0.5608**	0.5188	0.5384	0.4674
	Modality Gap	0.80	**0.05**	0.14	0.82
VISTA (LU)	T→I@1	**0.4725**	0.4362	0.4467	0.3201
	I→T@1	**0.5842**	0.5416	0.5682	0.3054
	Modality Gap	0.30	**0.18**	0.22	0.44
		Conceptual Captions			
CLIP (LU)	T→I@1	**0.3856**	0.3394	0.3470	0.3255
	I→T@1	**0.3954**	0.3780	0.3807	0.3517
	Modality Gap	0.60	**0.02**	0.03	0.81
VISTA (LU)	T→I@1	0.3954	0.3872	**0.4028**	0.3547
	I→T@1	**0.4043**	0.3881	0.3956	0.2938
	Modality Gap	0.24	**0.14**	0.16	0.35

Table 5.2: Modality Gap vs. Retrieval: The retrieval performance of the LU strategy across different loss functions is compared to the modality gap to examine the relationship between retrieval effectiveness and the modality gap.

scores of 12.3750 and 34.1871, both below the zero-shot score (44.27793).

The results suggest that the optimal fine-tuning strategy for CLIP in the vector arithmetic task is **UU**, offering a near-optimal balance with a score of 45.4273 and a gap of 0.03 compared to the highest score (45.4611) and lowest gap (0.02). For VISTA, the **LU** strategy (similar to retrieval) yields the highest score (48.9935) with an acceptable gap of 0.18, relative to the minimum possible gap of 0.05. In both cases, $\mathbf{L_{CUA}}$ emerges as the most effective loss function. Additionally, dataset selection is crucial in fine-tuning, as Conceptual Captions, with fewer and less diverse captions compared to MSCOCO (one caption per image versus five), enables greater modality gap reduction that is less favorable for SIMAT scores.

Interestingly, in the zero-shot setting, VISTA's SIMAT score is more than double that of CLIP. One possible explanation is that the SIMAT score is correlated with the modality gap. VISTA's architecture (see Appendix B.1) inherently results in a lower modality gap, which contributes to its superior SIMAT score in the zero-shot setting.

		fine-tuned on MSCOCO		
		L_{clip}	L_{CUA}	L_{CUAXU}
CLIP	LL	20.7555	39.7298	35.9161
	LU	23.1801	42.7115	40.8420
	UL	20.7623	41.7190	38.4732
	UU	24.9087	45.4273	**45.4611**
VISTA	LU	44.8295	**48.9935**	48.2509
	UL	37.0861	39.3065	38.3298
	UU	42.2497	1.0530	44.0940
		fine-tuned on Conceptual Captions		
CLIP	LL	21.3792	28.6908	27.8886
	LU	25.7402	36.7420	35.7245
	UL	14.4948	32.2853	33.8313
	UU	20.5068	34.9658	37.2335
VISTA	LU	42.2328	40.7463	43.6419
	UL	26.8703	29.4735	27.5182
	UU	30.9459	12.3750	34.1871
		ZS^*		
CLIP		16.3259		
VISTA		44.27793		

Table 5.3: SIMAT [20] scores of CLIP [10] and VISTA [25] fine-tuned on MSCOCO [74] and Conceptual Captions [82] datasets using different loss functions. ZS^* stands for Zero-Shot, indicating that it has not been fine-tuned. The highest score for each model is bold.

		fine-tuned on MSCOCO		
		L_{clip}	L_{CUA}	L_{CUAXU}
CLIP	LL	0.83	0.09	0.20
	LU	0.80	0.05	0.14
	UL	0.83	0.07	0.15
	UU	0.90	0.03	0.10
	ZS*	0.82		
VISTA	LU	0.30	0.18	0.22
	UL	0.38	0.36	0.72
	UU	0.24	0.87	0.05
	ZS*	0.44		
		fine-tuned on Conceptual Captions		
CLIP	LL	0.67	0.08	0.16
	LU	0.60	**0.02**	0.03
	UL	0.59	0.04	0.6
	UU	0.67	0.03	0.03
	ZS*	0.81		
VISTA	LU	0.24	0.14	0.16
	UL	0.29	0.33	0.62
	UU	0.14	**0.05**	**0.05**
	ZS*	0.35		

Table 5.4: Modality gap of CLIP [10] and VISTA [25] fine-tuned on MSCOCO [74] and Conceptual Captions [82] datasets using different loss functions. ZS* stands for Zero-Shot, indicating that it has not been fine-tuned. The minimum gap for each model is bold.

Chapter 6

Conclusion and Future Work

In this thesis, an analysis was conducted to investigate the effect of model architecture on the modality gap. Building upon previous work, a novel combination of fine-tuning strategies and tailored loss functions was proposed to mitigate the modality gap in the shared-encoder architecture. Similar to the dual-encoder architecture, it was observed that there is a negative relationship between modality gap reduction and retrieval performance. Conversely, a positive relationship was found between modality gap reduction (to some extent) and vector arithmetic performance.

Additionally, it was demonstrated that in a zero-shot setting, the shared-encoder architecture exhibits a lower modality gap and outperforms the dual-encoder architecture in text-to-image retrieval and vector arithmetic, while the dual-encoder architecture shows better performance in image-to-text retrieval. Beyond reducing the modality gap, the proposed method also proved effective in enhancing performance in both retrieval and vector arithmetic tasks.

The LU strategy, which involves locking the image-processing components while unlocking the text-processing components, combined with the L_{clip} loss function, was identified as the most effective approach to improve retrieval performance in both architectures. Moreover, this strategy, when combined with the L_{CUA} loss function, also yielded the best performance improvements for the shared-encoder architecture in vector arithmetic tasks. However, for the dual-encoder architecture, the UU (Unlocking-Unlocking) strategy combined with the same loss function exhibited the best results for enhancing vector arithmetic performance.

Despite the interesting findings, there is still work to be done for further improvement:

- Considering more downstream tasks such as image captioning and visual question answering. It was observed that reducing the modality gap has

different effects on retrieval and vector arithmetic. Moreover, different fine-tuning strategies and loss functions were found to be effective in improving performance on these downstream tasks. Thus, it is worth exploring more downstream tasks.

- Considering more datasets. As mentioned in the vector arithmetic analysis, the dataset selected for fine-tuning plays an important role. Testing on additional datasets may lead to interesting findings.
- VISTA uses a text encoder as its backbone, which may introduce bias to the model. An interesting direction could be to use modality-agnostic models such as Perceiver, Data2Vec, etc., as the backbone in the shared-encoder architecture.

Appendix A

Retrieval Results of L_{CUA} and L_{CUAXU}

The retrieval results related to fine-tuning with L_{CUA} and L_{CUAXU} are provided in Tables A.1 and A.2. As mentioned in Section 5.4.1, the models fine-tuned with L_{clip} achieve the best performance compared to the other two loss functions, and the results presented here support this finding. While the overall pattern aligns with the explanation in Section 5.4.1, there is a notable difference: when fine-tuning CLIP with L_{CUA} and L_{CUAXU}, the best strategy is LL rather than LU.

		MSCOCO			Conceptual Captions		
		R@1	R@5	R@10	R@1	R@5	R@10
Text to Image							
CLIP FT* by L_{clip}	LU	0.4175	0.7012	0.8030	0.3856	0.6349	0.7322
CLIP	LL	0.3796	0.6534	0.7636	**0.3777**	**0.6285**	**0.7178**
	LU	**0.3830**	**0.6653**	**0.7663**	0.3394	0.5970	0.6965
	UL	0.3566	0.6284	0.7368	0.3005	0.5415	0.6420
	UU	0.3499	0.6319	0.7440	0.2706	0.5163	0.6215
	ZS*	0.3082	0.5504	0.6620	0.3255	0.5527	0.6454
VISTA FT* by L_{clip}	LU	0.4725	0.7482	0.8360	0.4273	0.6676	0.7460
VISTA	LU	**0.4362**	**0.7116**	**0.8073**	**0.3872**	**0.6264**	**0.7164**
	UL	0.3875	0.6668	0.7769	0.2978	0.5276	0.6244
	UU	0.0004	0.0026	0.0065	0.0810	0.2392	0.3416
	ZS*	0.3201	0.5743	0.6048	0.3547	0.5673	0.65
Image to Text							
CLIP FT* by L_{clip}	LU	0.5608	0.7696	0.8490	0.3954	0.6502	0.7392
CLIP	LL	0.4828	0.6986	0.7998	**0.3840**	**0.6373**	**0.7279**
	LU	**0.5188**	**0.7380**	**0.8304**	0.3780	0.6296	0.7246
	UL	0.4222	0.6364	0.7484	0.2776	0.5190	0.6228
	UU	0.4458	0.6722	0.7726	0.2752	0.5211	0.6252
	ZS*	0.4674	0.6662	0.7554	0.3517	0.5771	0.6688
VISTA FT* by L_{clip}	LU	0.5842	0.7816	0.8608	0.4043	0.6480	0.7339
VISTA	LU	**0.5416**	**0.7526**	**0.8344**	**0.3881**	**0.6315**	**0.7248**
	UL	0.4438	0.6668	0.7756	0.2690	0.4986	0.6003
	UU	0.0002	0.0020	0.0044	0.0802	0.2307	0.3318
	ZS*	0.3054	0.495	0.6048	0.2938	0.5014	0.5820

Table A.1: Retrieval results of CLIP [10] and VISTA [25] on MSCOCO [74] and Conceptual Captions [82] datasets. Fine-tuned using L_{CUA}. ZS* stands for Zero-Shot, indicating that it has not been fine-tuned. FT* stands for Fine-Tuned. The best performance of each model is bold. For the convenience, the results related to models fine-tuned by L_{clip} in LU setting are provided.

		MSCOCO			Conceptual Captions		
		R@1	R@5	R@10	R@1	R@5	R@10
Text to Image							
CLIP FT* by L_{clip}	LU	0.4175	0.7012	0.8030	0.3856	0.6349	0.7322
CLIP	LL	0.3831	0.6596	0.7675	**0.3778**	**0.6285**	**0.7186**
	LU	**0.3954**	**0.6788**	**0.7846**	0.3470	0.6111	0.7070
	UL	0.3644	0.6381	0.7440	0.3038	0.5437	0.6444
	UU	0.3623	0.6429	0.7496	0.2842	0.5270	0.6303
	ZS*	0.3082	0.5504	0.6620	0.3255	0.5527	0.6454
VISTA FT* by L_{clip}	LU	0.4725	0.7482	0.8360	0.4273	0.6676	0.7460
VISTA	LU	**0.4467**	**0.7217**	**0.8134**	**0.4028**	**0.6394**	**0.7278**
	UL	0.3064	0.5838	0.7029	0.2515	0.4757	0.5734
	UU	0.3587	0.6477	0.7630	0.2722	0.5074	0.6054
	ZS*	0.3201	0.5743	0.6048	0.3547	0.5673	0.65
Image to Text							
CLIP FT* by L_{clip}	LU	0.5608	0.7696	0.8490	0.3954	0.6502	0.7392
CLIP	LL	0.5028	0.7102	0.8018	**0.3888**	**0.6383**	**0.7310**
	LU	**0.5384**	**0.7528**	**0.8344**	0.3807	0.6374	0.7261
	UL	0.4308	0.6548	0.7534	0.2861	0.5264	0.6324
	UU	0.4530	0.6764	0.7794	0.2869	0.5361	0.6350
	ZS*	0.4674	0.6662	0.7554	0.3517	0.5771	0.6688
VISTA FT* by L_{clip}	LU	0.5842	0.7816	0.8608	0.4043	0.6480	0.7339
VISTA	LU	**0.5682**	**0.7622**	**0.8408**	**0.3956**	**0.6397**	**0.7293**
	UL	0.3714	0.5882	0.6970	0.2313	0.4541	0.5531
	UU	0.4436	0.6804	0.7822	0.2578	0.4978	0.6036
	ZS*	0.3054	0.495	0.6048	0.2938	0.5014	0.5820

Table A.2: Retrieval results of CLIP [10] and VISTA [25] on MSCOCO [74] and Conceptual Captions [82] datasets. Fine-tuned using L_{CUAXU}. ZS* stands for Zero-Shot, indicating that it has not been fine-tuned. FT* stands for Fine-Tuned. The best performance of each model is bold. For the convenience, the results related to models fine-tuned by L_{clip} in LU setting are provided.

Appendix B

Modality Gap Measured by CD and CMD

		MSCOCO					
		L_{clip}		L_{CUA}		L_{CUAXU}	
		CD	CMD	CD	CMD	CD	CMD
CLIP	LL	0.83	0.86	0.09	0.10	0.20	0.22
	LU	0.80	0.83	0.05	0.06	0.14	0.15
	UL	0.83	0.85	0.07	0.08	0.15	0.17
	UU	0.90	0.92	0.03	0.04	0.10	0.11
	ZS*		0.82			0.86	
VISTA	LU	0.30	0.30	0.18	0.18	0.22	0.22
	UL	0.38	0.39	0.36	0.37	0.72	0.72
	UU	0.24	0.24	0.87	0.90	0.05	0.05
	ZS*		0.44			0.44	
		Conceptual Captions					
CLIP	LL	0.67	0.72	0.08	0.10	0.16	0.18
	LU	0.60	0.64	0.02	0.03	0.03	0.04
	UL	0.59	0.63	0.04	0.05	0.06	0.07
	UU	0.67	0.71	0.03	0.03	0.03	0.04
	ZS*		0.81			0.85	
VISTA	LU	0.24	0.24	0.14	0.14	0.16	0.16
	UL	0.29	0.30	0.33	0.34	0.62	0.63
	UU	0.14	0.14	0.05	0.05	0.05	0.05
	ZS*		0.35			0.36	

Table B.1: Modality gap (measured by CD and CMD) of CLIP [10] and VISTA [25] on MSCOCO [74] and Conceptual Captions [82] datasets. Fine-tuned using different loss functions. ZS* stands for Zero-Shot, indicating that it has not been fine-tuned. The minimum of each model is bold.

Bibliography

[1] Vishaal Udandarao. «Understanding and fixing the modality gap in vision-language models». In: *Master's thesis, University of Cambridge* (2022) (cit. on pp. 1, 3, 15, 20, 21).

[2] Rishi Bommasani et al. «On the opportunities and risks of foundation models». In: *arXiv preprint arXiv:2108.07258* (2021) (cit. on p. 1).

[3] Jean-Baptiste Alayrac et al. «Flamingo: a visual language model for few-shot learning». In: *Advances in neural information processing systems* 35 (2022), pp. 23716–23736 (cit. on pp. 1, 9, 13).

[4] Zi-Yi Dou et al. «An empirical study of training end-to-end vision-and-language transformers». In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 18166–18176 (cit. on pp. 1, 9).

[5] Chao Jia et al. «Scaling up visual and vision-language representation learning with noisy text supervision». In: *International conference on machine learning*. PMLR. 2021, pp. 4904–4916 (cit. on pp. 1, 9, 13, 16).

[6] Gukyeong Kwon, Zhaowei Cai, Avinash Ravichandran, Erhan Bas, Rahul Bhotika, and Stefano Soatto. «Masked vision and language modeling for multi-modal representation learning». In: *arXiv preprint arXiv:2208.02131* (2022) (cit. on p. 1).

[7] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. «Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation». In: *International conference on machine learning*. PMLR. 2022, pp. 12888–12900 (cit. on pp. 1, 9).

[8] Liunian Harold Li et al. «Grounded language-image pre-training». In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 10965–10975 (cit. on p. 1).

[9] Hieu Pham et al. «Combined scaling for zero-shot transfer learning». In: *Neurocomputing* 555 (2023), p. 126658 (cit. on p. 1).

[10] Alec Radford et al. «Learning transferable visual models from natural language supervision». In: *International conference on machine learning*. PMLR. 2021, pp. 8748–8763 (cit. on pp. 1, 9, 11–13, 16, 22, 24, 25, 27, 29, 31, 32, 36, 37, 39).

[11] Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. «Flava: A foundational language and vision alignment model». In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 15638–15650 (cit. on pp. 1, 13, 16).

[12] Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. «Simvlm: Simple visual language model pretraining with weak supervision». In: *arXiv preprint arXiv:2108.10904* (2021) (cit. on pp. 1, 9, 13).

[13] Jinyu Yang, Jiali Duan, Son Tran, Yi Xu, Sampath Chanda, Liqun Chen, Belinda Zeng, Trishul Chilimbi, and Junzhou Huang. «Vision-language pre-training with triple contrastive learning». In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 15671–15680 (cit. on p. 1).

[14] Haoxuan You, Luowei Zhou, Bin Xiao, Noel Codella, Yu Cheng, Ruochen Xu, Shih-Fu Chang, and Lu Yuan. «Learning visual representation from modality-shared contrastive language-image pre-training». In: *European Conference on Computer Vision*. Springer. 2022, pp. 69–87 (cit. on p. 1).

[15] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. «Coca: Contrastive captioners are image-text foundation models». In: *arXiv preprint arXiv:2205.01917* (2022) (cit. on pp. 1, 9).

[16] Abrar Fahim, Alex Murphy, and Alona Fyshe. «Its Not a Modality Gap: Characterizing and Addressing the Contrastive Gap». In: *arXiv preprint arXiv:2405.18570* (2024) (cit. on pp. 1, 20–23).

[17] Victor Weixin Liang, Yuhui Zhang, Yongchan Kwon, Serena Yeung, and James Y Zou. «Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning». In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 17612–17625 (cit. on pp. 1, 15, 18, 20).

[18] Peiyang Shi, Michael C Welle, Mårten Björkman, and Danica Kragic. «Towards understanding the modality gap in CLIP». In: *ICLR 2023 Workshop on Multimodal Representation Learning: Perks and Pitfalls*. 2023 (cit. on pp. 1, 15, 20).

[19] Changdae Oh, Junhyuk So, Hoyoon Byun, YongTaek Lim, Minchul Shin, Jong-June Jeon, and Kyungwoo Song. «Geodesic multi-modal mixup for robust fine-tuning». In: *Advances in Neural Information Processing Systems* 36 (2024) (cit. on pp. 1, 20).

[20] Guillaume Couairon, Matthijs Douze, Matthieu Cord, and Holger Schwenk. «Embedding arithmetic of multimodal queries for image retrieval». In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 4950–4958 (cit. on pp. 1, 4, 21, 22, 26, 31).

[21] Yuhui Zhang, Jeff Z HaoChen, Shih-Cheng Huang, Kuan-Chieh Wang, James Zou, and Serena Yeung. «Diagnosing and rectifying vision models using language». In: *arXiv preprint arXiv:2302.04269* (2023) (cit. on pp. 1, 21).

[22] Simon Schrodi, David T Hoffmann, Max Argus, Volker Fischer, and Thomas Brox. «Two Effects, One Trigger: On the Modality Gap, Object Bias, and Information Imbalance in Contrastive Vision-Language Representation Learning». In: *arXiv preprint arXiv:2404.07983* (2024) (cit. on pp. 1, 15, 21).

[23] Chao Yi, De-Chuan Zhan, and Han-Jia Ye. «Bridge the modality and capacity gaps in vision-language model selection». In: *arXiv preprint arXiv:2403.13797* (2024) (cit. on pp. 1, 15).

[24] Rakesh Chada, Zhaoheng Zheng, and Pradeep Natarajan. «Momo: A shared encoder model for text, image and multi-modal representations». In: *arXiv preprint arXiv:2304.05523* (2023) (cit. on pp. 1, 13).

[25] Junjie Zhou, Zheng Liu, Shitao Xiao, Bo Zhao, and Yongping Xiong. «VISTA: Visualized Text Embedding For Universal Multi-Modal Retrieval». In: *arXiv preprint arXiv:2406.04292* (2024) (cit. on pp. 1, 13, 14, 18, 21, 22, 25, 29, 31, 32, 36, 37, 39).

[26] Jacob Devlin. «Bert: Pre-training of deep bidirectional transformers for language understanding». In: *arXiv preprint arXiv:1810.04805* (2018) (cit. on pp. 3, 16).

[27] Yinhan Liu. «Roberta: A robustly optimized bert pretraining approach». In: *arXiv preprint arXiv:1907.11692* (2019) (cit. on p. 3).

[28] Tom B Brown. «Language models are few-shot learners». In: *arXiv preprint arXiv:2005.14165* (2020) (cit. on p. 3).

[29] M Lewis. «Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension». In: *arXiv preprint arXiv:1910.13461* (2019) (cit. on p. 3).

[30] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. «Exploring the limits of transfer learning with a unified text-to-text transformer». In: *Journal of machine learning research* 21.140 (2020), pp. 1–67 (cit. on p. 3).

[31] Zhe Gan, Linjie Li, Chunyuan Li, Lijuan Wang, Zicheng Liu, Jianfeng Gao, et al. «Vision-language pre-training: Basics, recent advances, and future trends». In: *Foundations and Trends® in Computer Graphics and Vision* 14.3–4 (2022), pp. 163–352 (cit. on pp. 3–11, 20).

[32] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. «Vqa: Visual question answering». In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 2425–2433 (cit. on p. 4).

[33] Drew A Hudson and Christopher D Manning. «Gqa: A new dataset for real-world visual reasoning and compositional question answering». In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 6700–6709 (cit. on p. 4).

[34] Tejas Gokhale, Pratyay Banerjee, Chitta Baral, and Yezhou Yang. «Vqa-lol: Visual question answering under the lens of logic». In: *European conference on computer vision*. Springer. 2020, pp. 379–396 (cit. on p. 4).

[35] Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. «From recognition to cognition: Visual commonsense reasoning». In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 6720–6731 (cit. on p. 4).

[36] A Vaswani. «Attention is all you need». In: *Advances in Neural Information Processing Systems* (2017) (cit. on pp. 6, 7, 11, 13).

[37] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. «Deep residual learning for image recognition». In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778 (cit. on pp. 7, 11).

[38] Alexey Dosovitskiy et al. «An image is worth 16x16 words». In: *arXiv preprint arXiv:2010.11929* 7 (2020) (cit. on pp. 7, 11, 13, 14, 16).

[39] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. «Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks». In: *Advances in neural information processing systems* 32 (2019) (cit. on p. 9).

[40] Hao Tan and Mohit Bansal. «Lxmert: Learning cross-modality encoder representations from transformers». In: *arXiv preprint arXiv:1908.07490* (2019) (cit. on p. 9).

[41] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. «Visualbert: A simple and performant baseline for vision and language». In: *arXiv preprint arXiv:1908.03557* (2019) (cit. on p. 9).

[42] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. «Vl-bert: Pre-training of generic visual-linguistic representations». In: *arXiv preprint arXiv:1908.08530* (2019) (cit. on p. 9).

[43] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. «Uniter: Universal image-text representation learning». In: *European conference on computer vision*. Springer. 2020, pp. 104–120 (cit. on p. 9).

[44] Xiujun Li et al. «Oscar: Object-semantics aligned pre-training for vision-language tasks». In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX 16*. Springer. 2020, pp. 121–137 (cit. on p. 9).

[45] Zhe Gan, Yen-Chun Chen, Linjie Li, Chen Zhu, Yu Cheng, and Jingjing Liu. «Large-scale adversarial training for vision-and-language representation learning». In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 6616–6628 (cit. on p. 9).

[46] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. «Vinvl: Revisiting visual representations in vision-language models». In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 5579–5588 (cit. on p. 9).

[47] Wei Li, Can Gao, Guocheng Niu, Xinyan Xiao, Hao Liu, Jiachen Liu, Hua Wu, and Haifeng Wang. «Unimo: Towards unified-modal understanding and generation via cross-modal contrastive learning». In: *arXiv preprint arXiv:2012.15409* (2020) (cit. on p. 9).

[48] Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. «Unifying vision-and-language tasks via text generation». In: *International Conference on Machine Learning*. PMLR. 2021, pp. 1931–1942 (cit. on p. 9).

[49] Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. «Mdetr-modulated detection for end-to-end multi-modal understanding». In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 1780–1790 (cit. on p. 9).

[50] Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Faisal Ahmed, Zicheng Liu, Yumao Lu, and Lijuan Wang. «Unitab: Unifying text and box outputs for grounded vision-language modeling». In: *European Conference on Computer Vision*. Springer. 2022, pp. 521–539 (cit. on p. 9).

[51] Peng Wang et al. «Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework». In: *International conference on machine learning*. PMLR. 2022, pp. 23318–23340 (cit. on p. 9).

[52] Zhicheng Huang, Zhaoyang Zeng, Bei Liu, Dongmei Fu, and Jianlong Fu. «Pixel-bert: Aligning image pixels with text by deep multi-modal transformers». In: *arXiv preprint arXiv:2004.00849* (2020) (cit. on p. 9).

[53] Zhicheng Huang, Zhaoyang Zeng, Yupan Huang, Bei Liu, Dongmei Fu, and Jianlong Fu. «Seeing out of the box: End-to-end pre-training for vision-language representation learning». In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 12976–12985 (cit. on p. 9).

[54] 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?» In: *arXiv preprint arXiv:2107.06383* (2021) (cit. on pp. 9, 12).

[55] Wonjae Kim, Bokyung Son, and Ildoo Kim. «Vilt: Vision-and-language transformer without convolution or region supervision». In: *International conference on machine learning*. PMLR. 2021, pp. 5583–5594 (cit. on p. 9).

[56] Hongwei Xue, Yupan Huang, Bei Liu, Houwen Peng, Jianlong Fu, Houqiang Li, and Jiebo Luo. «Probing inter-modality: Visual parsing with self-attention for vision-and-language pre-training». In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 4514–4528 (cit. on p. 9).

[57] Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. «Git: A generative image-to-text transformer for vision and language». In: *arXiv preprint arXiv:2205.14100* (2022) (cit. on p. 9).

[58] Hangbo Bao, Wenhui Wang, Li Dong, Qiang Liu, Owais Khan Mohammed, Kriti Aggarwal, Subhojit Som, and Furu Wei. «Vlmo: Unified vision-language pre-training with mixture-of-modality-experts». In: *arXiv preprint arXiv:2111.02358* (2021) (cit. on p. 9).

[59] Wenhui Wang et al. «Image as a foreign language: Beit pretraining for all vision and vision-language tasks». In: *arXiv preprint arXiv:2208.10442* (2022) (cit. on p. 9).

[60] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. «Align before fuse: Vision and language representation learning with momentum distillation». In: *Advances in neural information processing systems* 34 (2021), pp. 9694–9705 (cit. on pp. 9, 15).

[61] Zi-Yi Dou et al. «Coarse-to-fine vision-language pre-training with fusion in the backbone». In: *Advances in neural information processing systems* 35 (2022), pp. 32942–32956 (cit. on p. 9).

[62] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. «Zero-shot text-to-image generation». In: *International conference on machine learning*. Pmlr. 2021, pp. 8821–8831 (cit. on p. 11).

[63] Aaron Van Den Oord, Oriol Vinyals, et al. «Neural discrete representation learning». In: *Advances in neural information processing systems* 30 (2017) (cit. on p. 11).

[64] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. «Clip-adapter: Better vision-language models with feature adapters». In: *International Journal of Computer Vision* 132.2 (2024), pp. 581–595 (cit. on p. 12).

[65] Konstantin Schall, Kai Uwe Barthel, Nico Hezel, and Klaus Jung. «Optimizing CLIP Models for Image Retrieval with Maintained Joint-Embedding Alignment». In: *International Conference on Similarity Search and Applications*. Springer. 2024, pp. 97–110 (cit. on p. 12).

[66] Ziqin Zhou, Yinjie Lei, Bowen Zhang, Lingqiao Liu, and Yifan Liu. «Zegclip: Towards adapting clip for zero-shot semantic segmentation». In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 11175–11185 (cit. on p. 12).

[67] Johnathan Xie and Shuai Zheng. «Zero-shot object detection through vision-language embedding alignment». In: *2022 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE. 2022, pp. 1–15 (cit. on p. 12).

[68] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. «A clip-hitchhiker's guide to long video retrieval». In: *arXiv preprint arXiv:2205.08508* (2022) (cit. on p. 12).

[69] Renrui Zhang, Ziyao Zeng, Ziyu Guo, and Yafeng Li. «Can language understand depth?» In: *Proceedings of the 30th ACM International Conference on Multimedia*. 2022, pp. 6868–6874 (cit. on p. 12).

[70] Ron Mokady, Amir Hertz, and Amit H Bermano. «Clipcap: Clip prefix for image captioning». In: *arXiv preprint arXiv:2111.09734* (2021) (cit. on p. 12).

[71] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. «Perceiver: General perception with iterative attention». In: *International conference on machine learning*. PMLR. 2021, pp. 4651–4664 (cit. on p. 13).

[72] Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. «Data2vec: A general framework for self-supervised learning in speech, vision and language». In: *International Conference on Machine Learning*. PMLR. 2022, pp. 1298–1312 (cit. on p. 13).

[73] Yiyuan Zhang, Kaixiong Gong, Kaipeng Zhang, Hongsheng Li, Yu Qiao, Wanli Ouyang, and Xiangyu Yue. «Meta-transformer: A unified framework for multimodal learning». In: *arXiv preprint arXiv:2307.10802* (2023) (cit. on p. 13).

[74] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. «Microsoft coco: Common objects in context». In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer. 2014, pp. 740–755 (cit. on pp. 15–19, 26, 29, 31, 32, 36, 37, 39).

[75] Leland McInnes, John Healy, and James Melville. «Umap: Uniform manifold approximation and projection for dimension reduction». In: *arXiv preprint arXiv:1802.03426* (2018) (cit. on pp. 15, 17).

[76] Mingxing Tan and Quoc Le. «Efficientnet: Rethinking model scaling for convolutional neural networks». In: *International conference on machine learning*. PMLR. 2019, pp. 6105–6114 (cit. on p. 15).

[77] Shashank Goel, Hritik Bansal, Sumit Bhatia, Ryan Rossi, Vishwa Vinay, and Aditya Grover. «Cyclip: Cyclic contrastive language-image pretraining». In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 6704–6719 (cit. on p. 16).

[78] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. «Imagebind: One embedding space to bind them all». In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 15180–15190 (cit. on p. 16).

[79] Werner Zellinger, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, and Susanne Saminger-Platz. «Central moment discrepancy (cmd) for domain-invariant representation learning». In: *arXiv preprint arXiv:1702.08811* (2017) (cit. on p. 18).

[80] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. «Lit: Zero-shot transfer with locked-image text tuning». In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 18123–18133 (cit. on pp. 22, 24, 28).

[81] Tongzhou Wang and Phillip Isola. «Understanding contrastive representation learning through alignment and uniformity on the hypersphere». In: *International conference on machine learning*. PMLR. 2020, pp. 9929–9939 (cit. on p. 23).

[82] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. «Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning». In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2018, pp. 2556–2565 (cit. on pp. 26, 29, 31, 32, 36, 37, 39).

[83] Ilya Loshchilov, Frank Hutter, et al. «Fixing weight decay regularization in adam». In: *arXiv preprint arXiv:1711.05101* 5 (2017) (cit. on p. 27).

[84] Ilya Loshchilov and Frank Hutter. «Sgdr: Stochastic gradient descent with warm restarts». In: *arXiv preprint arXiv:1608.03983* (2016) (cit. on p. 27).