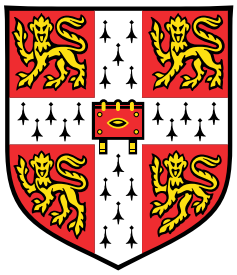
# Understanding and Fixing the Modality Gap in Vision-Language Models



**Vishaal Udandarao**

Department of Engineering
University of Cambridge

This dissertation is submitted for the degree of
*Master of Philosophy in Machine Learning and Machine Intelligence*

St Edmund's College

August 2022

To Sreenivas and Usha Rani Udandarao, for everything.

# Declaration

I, Vishaal Udandarao of St Edmund's College, hereby declare that this thesis and its constituent sections are my own work, unaided except as may be specified in the references and the acknowledgements. The report does not contain material that has already been used to any substantial extent for a comparable purpose. All code used in this report has been written by me in Python, mainly using the Pytorch and Numpy packages, except as specified below:

1. For the few-shot experiments, the dataloader and zero-shot evaluation scripts were adapted from TIP-Adapter, CoOP and CLIP.

2. For the vector arithmetic experiments, the dataset and evaluation scripts were adapted from SIMAT.

3. For all the baseline comparisons, pre-trained models were used from BLIP, TCL, CLIP-Adapter and TIP-Adapter

This thesis contains 14973 words including appendices.

Vishaal Udandarao
August 2022

# Acknowledgements

This thesis is the culmination of the direct and indirect efforts of innumerable people. It is impossible to enumerate them all, but here is an honest attempt.

To my thesis advisors, Samuel Albanie and Ankush Gupta. You have made my time working on the thesis the most rewarding and enlightening. Thank you for being so patient with me, offering me countless ideas, and guiding me every step along the way. Sam: Your resolve to correctly define, investigate and solve a problem the right way has been a great inspiration to me, and taught me the right way to do research. Ankush: Your ability to generate ideas out of seemingly thin air were astounding and motivated me to get better at connecting the dots. I am grateful to have had this opportunity to be advised by both of you, and hope we can work together again.

To all the folks of the MPhil MLMI 2022 cohort. It was great fun getting to know each one of you. My time in Cambridge would not have been the same if it was not for all the fun times we shared.

To all my friends around the world. Keeping in touch across cities and continents is hard, but thank you for always being available as pillars of fun, motivation and support.

To my family, my parents and my brother. Thank you for always believing in me and supporting me through all my decisions. Knowing that you will always be beacons of encouragement and solace makes everything easier.

Lastly, To God, without whom everything would be futile.

# Abstract

Contrastive language-image pre-training has emerged to be a simple yet effective way to train large-scale vision-language models [165, 83, 181, 220] that are capable of learning semantic and structured information jointly from images and texts – this has been fostered by the curation of internet-scale multi-modal datasets. These models are capable of remarkable zero-shot performance on unseen visual tasks corroborating their rich correlated multi-modal knowledge. This has led to their widespread use in several downstream tasks like image classification, semantic segmentation, visual question-answering *etc*. However, a key understanding of why these models work so well is still lacking.

In this thesis, we aim to understand one such large-scale vision-language model, CLIP [165]. We dive deep into unpacking the CLIP model architecture, and present a counter-intuitive phenomenon that occurs in its embedding space called the *Modality Gap*. We showcase the existence of this modality gap in several settings and hypothesise possible reasons for its existence. We then systematically study the emergence of this modality gap by trying to reproduce realistic behaviour through several toy experiments, which we then transfer to real world settings. Having gained an improved understanding of what the modality gap is and why it exists, we set about delineating its implications on downstream tasks. We discuss how the presence of the modality gap prevents effective visualisation of vision-language embedding spaces. We then present a simple method to mitigate this issue thereby leading to an increased interpretability of CLIP's embedding space. We then uncover problems with CLIP's intra-image embedding space, and discuss its implications on few-shot classification tasks. We propose a method called *TIP-X* that fixes these issues and achieves state-of-the art results for few-shot classification on 11 benchmark datasets. Finally, we study a new task, vector arithmetic, under the light of the modality gap in CLIP's embedding space. We conclude our thesis by discussing the implications of the modality gap on downstream task and vector arithmetic performance, and find interesting and conflicting regimes emerging.

# Table of contents

# List of figures

# List of tables

# Chapter 1

# Introduction

We perceive the world around us through multiple sensory inputs [142]. For example, our eyes help us sense our surroundings through the visual lens, and our ears help us localise sounds related to diverse objects. Clearly, our brain simultaneously processes such multi-modal inputs [101, 188, 9] – the complementary information encoded by these different modalities is combined efficiently to enable us to perform varied tasks [183, 21].

Inspired by this multi-modal processing in the human brain, the deep learning community has embraced the idea of learning from multiple modalities to effectively solve several tasks. Recent years have seen an influx of large pre-trained multi-modal foundation models [16] that efficiently train on web-scale multi-modal datasets, and are capable of learning data representations that are transferable to several uni-modal and multi-modal downstream tasks [165, 160, 181, 209, 223, 83, 3, 111, 218, 100, 115, 48, 222]. In this thesis, we limit our scope to studying pre-trained vision-language models *i.e.* models that are trained to learn representations for image and text data only.

OpenAI's CLIP [165] is one prominent vision-language model that has shown impressive zero-shot classification and cross-modal retrieval performance. By leveraging large-scale paired image-text data, CLIP learns a shared multi-modal embedding space with rich semantic properties. CLIP's remarkable performance on unseen visual tasks further demonstrates the utility of this embedding space in not only correlating multi-modal knowledge, but also improving downstream task performance. As a result, the emergence of CLIP has fostered several works that aim to utilise its strong representation learning capabilities on a myriad of downstream tasks including few-shot image classification [232, 56, 239, 240, 235], video-retrieval [11, 10, 127], depth estimation [236], image captioning [223, 140, 15, 32] and visual question answering [178, 92]. However, most of these methods are heuristic and somewhat poorly understood. They simply optimise for downstream task performance without comprehending the effects on the underlying embedding space. Rather than taking this top-down approach, a more principled bottom-up approach of understanding the embedding space can lead to better insights into why the model performs so well, contributing to increased interpretability. There are several such bottom-up analyses of uni-modal embedding spaces in self-supervised representation learning [203, 63, 37], manifold learning [191, 163] and generative modelling [8, 179, 175]. However, despite the

aforementioned plethora of vision-language models, only a handful of works try to understand the properties of their multi-modal embedding spaces.

One such interesting work from Liang et al. [118] uncovers a fascinating geometric phenomenon called the *Modality Gap*. They observe that the image and text embeddings of CLIP (and several of its variants) lie far apart in the embedding space. They explain this phenomenon by analysing the loss used to train CLIP-like models. They further perform some analyses on the implications of this effect on downstream tasks. However, they do not study these relationships in concrete detail and only provide a brief overview of the modality gap.

In this thesis, we aim to tease apart the modality gap phenomenon in greater detail. We first introduce the modality gap by demonstrating its existence in several vision-language models and subsequently hypothesising possible reasons for it. We then conduct toy simulations to test our hypotheses by digging into the properties of the embedding space. Finally, we discuss the implications of the modality gap phenomenon on multi-modal downstream tasks.

## 1.1 Contributions

We now lay out the explicit contributions made by this thesis to the field of multi-modal learning. To the best of our knowledge, these are novel contributions.

- We decompose the factors affecting the contrastive loss into six abstract factors, and provide intuitive explanations for the existence of the modality gap. Through this, we build on previous works that explain the behaviour of the embedding spaces of vision-language models (Chapter 4).
- We propose a novel yet simple method for the visualisation of the embedding spaces of vision-language models, that can capture their underlying semantic structure (Section 5.1).
- We dig into the properties of CLIP's embedding space, and uncover a pathology with respect to its intra-modal vs inter-modal similarity comparisons. By proposing a simple fix, we obtain state-of-the-art results on 11 benchmark datasets for few-shot image classification (Section 5.2).
- We provide an in-depth study of the relationships between modality gap, downstream task performance and vector arithmetic performance (Section 5.3).

## 1.2 Thesis Outline

We now outline the broad structure of the thesis, by enumerating each of the chapters with a brief description for each.

In Chapter 2, we provide a thorough review of multi-modal learning methods and their evolution over the years. As CLIP and its variants are used extensively in this work, we review CLIP's training

and inference paradigms. We also describe two major frameworks, namely *Prompt Learning* and *Adapters*, that aim to improve CLIP's few-shot classification performance. We then move into the first of the three technical chapters of the thesis.

In Chapter 3, we give a broad overview of the modality gap phenomenon. We describe the factors that lead to this phenomenon, and conduct a simple experiment to motivate why studying the modality gap is important.

In Chapter 4, we pick apart several components of the contrastive loss function and provide arguments for the formation of the modality gap. We conduct several toy and real-world experiments to study the behaviour of CLIP's embedding space under various conditions. Throughout this chapter, we discuss claims about the modality gap, and its relationship to CLIP's design choices.

In Chapter 5, we show why the modality gap phenomenon even matters in practice. We propose a simple yet effective method to visualise CLIP's embedding space that allows us to view the image and text embeddings under a new light. We subsequently lay out the implications of the modality gap on downstream task performance, and provide recommendations for how to improve it. With such an analysis, we propose a novel method that achieves state-of-the-art performance on few-shot image classification.

Finally, we conclude the thesis in Chapter 6 by summarising our main contributions and highlighting potential future research directions.

# Chapter 2

# Background

In this chapter, we position our work in the vast landscape of vision-language models. We briefly introduce the two major motivating pillars that fostered the rise of large-scale vision-language learning: (1) multi-modal learning, and (2) self-supervised contrastive learning. We then provide an overview of the different types of vision-language models in the literature, before diving into the backbone around which our thesis revolves, the CLIP model. We describe CLIP's model architecture, its training protocol, and its accompanying design choices in-depth. Finally, we explore several post-CLIP methods that repurpose, fine-tune and build upon CLIP for the task of few-shot image classification.

## 2.1 The Two Pillars

In this section, we give a broad overview of multi-modal learning and self-supervised contrastive learning, motivating them as the main pillars underlying the vast terrain of vision-language models.

### 2.1.1 Multi-modal learning

Human perception of the world is largely multi-modal [14, 79, 225]. Two key characteristics of multi-modal perception have received particular attention from psychologists:

1. *Degeneracy.* The principle of degeneracy (or redundancy) allows humans to function even with the loss of a sensory component [50]. For example, our knowledge of objects is not limited by sight alone, we experience things around us by using our other senses of touch, sound, and even smell. Several experimental studies [20, 161, 104, 17] investigated the effects of degeneracy in infants, concluding that the complementary nature of multiple sensory systems enable the development of the human cognitive system.

2. *Re-entry.* This principle refers to the simultaneous and explicit inter-relation of multiple representations across modalities. This implies that the sensation of one modality can subsequently invoke the sensation of another, suggesting that humans can effortlessly link high-level semantics across different modalities [51, 182, 87].

The efficacy of multi-modal learning in humans has inspired a multitude of machine learning methods that make use of cross-modal redundancy [145, 143, 144, 146, 4] and fusion [121, 90, 180, 212, 84] to solve diverse tasks. The effectiveness of these methods is a testament to the potential of using multiple modalities to improve representation quality and task performance.

### 2.1.2 Self-supervised Contrastive learning

In recent years, the dominant paradigm of machine learning has been supervised learning, where labelled data is provided with input-target pairs [139]. This is however a bottleneck for building intelligent generalist models that can perform multiple tasks adaptively [106]. This bottleneck stems from the laborious effort required to manually label large-scale datasets.

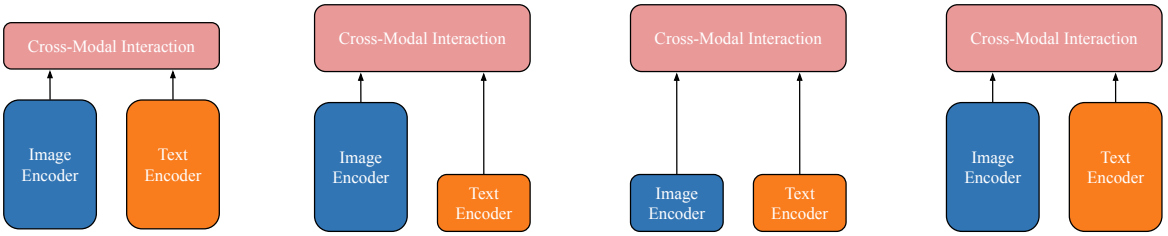
This limitation of supervised learning has paved the way for the field of self-supervised learning. It has enabled AI systems to learn from orders of magnitude more data and endowed them with the ability to understand subtle patterns of the world.

Self-supervised learning methods obtain proxy supervisory signals from the data itself. They often leverage the underlying structure of the data to predict unobserved/hidden parts of the inputs to obtain strong representations. This is common in the natural language processing literature with many models predicting masked, hidden or missing tokens from texts [45, 122, 94, 164, 35, 86, 134, 156] to learn high-quality representations. Most self-supervised learning approaches for computer vision use different kinds of *pretext tasks* for acquiring their proxy supervisory signal. Examples include predicting image rotations [58], colouring images [234], predicting patch spatial positions [46], solving jigsaw puzzles [150, 26] and predicting affine transformations [230, 168, 151, 89].

Underlying many recent works in self-supervised learning is the idea of contrastive learning: we learn representations for an input datapoint by maximising similarity with a noisy or transformed version of the input datapoint itself while minimising similarities with all other input datapoints [152]. These methods rely on the contrastive loss [162], first introduced by Gutmann et al. [68] to estimate unnormalised statistical models. In recent years, the profusion of internet-scale data and access to massive compute infrastructure has given rise to several of these self-supervised models [69, 27, 72, 196, 23, 73, 138, 28, 62, 31, 22], which achieve state-of-the-art results on visual tasks, and even close the performance gap to supervised models. For a deeper review of self-supervised learning, several in-depth surveys can be referred [171, 82, 99, 5, 85].

## 2.2 A Taxonomy of Vision-Language Models

Owing to the substantial but independent successes of self-supervised contrastive learning and multi-modal learning, it is natural to wonder if combining these approaches can lead to learning rich, semantic representations of the world that can be transferred to both uni-modal (image classification, text retrieval *etc.*) and multi-modal (cross-modal retrieval, visual-question answering *etc.*) tasks.




(a) IE = TE >> CI (b) IE > TE > CI (c) CI > IE = TE (d) IE = TE > CI

Fig. 2.1 **Taxonomy of the vision-language pre-training model landscape.** IE is short for image encoder, TE is short for text encoder, and CI is short for cross-modal interactions. The height of each box depicts the relative capacity of the corresponding model. For example, if IE = TE, this represents that both the image and text encoder have a comparable number of parameters. TE >> CI represents that the number of parameters for computing the cross-modal interaction is zero or almost negligent as compared to the text encoder. We adopt this taxonomy from [92].

There have been several initial attempts at answering this question by fusing the strong representation learning capabilities of uni-modal self-supervised learning methods [29, 27, 65, 227, 13, 30, 103, 109, 19, 166, 45] and supervised multi-modal learning methods [205, 60, 6, 67, 197, 77, 91]. Such approaches have used proxy tasks like image-captioning (VirTex [44]), masked language modelling (ICMLM [170]) or contrastive learning (ConVIRT [238]) to obtain rich representations, highlighting the potential of using natural language for learning representations that excel at downstream task transfer.

In the past few years, internet-scale multi-modal dataset curation has rapidly accelerated the growth of such models. This new breed of vision-language *foundation* models [16] appear to exhibit qualitatively different behaviours to their predecessors. Kim et al. [92] proposed a unifying taxonomy of such vision-language models[1], which we revisit in Figure 2.1 to ground our summary of several vision-language models. All these models use independent image and text encoders (blue and orange boxes in Figure 2.1). They then capture the interactions between the image and text embeddings (red box in Figure 2.1) using several different techniques.

Type 2.1a models use encoders of comparable capacity but perform a shallow-level fusion (simple dot-product) of image-text embeddings. CLIP [165] (discussed in detail in Section 2.3) falls under this umbrella of models – It uses two independent image and text encoders that output normalised embeddings which can be compared through simple dot products. Jia et al. [83] extended this by introducing ALIGN, which focused on scaling up the pre-training paradigm of CLIP. They scraped a dataset of 1.8 billion images with alt-text descriptions and employed it to train a CLIP-like architecture, achieving strong downstream performance on a suite of tasks. The ALIGN model also seems to acquire a compositional embedding space capable of doing vector arithmetic, similar to that exhibited by the Word2Vec [134, 135] model. We further explore this capability of vision-language models in detail in

[1]This taxonomy was also reused and slightly extended by Xu et al. [215]

Section 5.3. Some other examples of Type 2.1a models include BASIC [160], PyramidCLIP [57], DeCLIP [117], LiT [229], WenLan [80], K-LITE [177], CapKP [112] and SLIP [141].

Several models come under the umbrella of Type 2.1b where the image encoder is heavier than the text encoder. Examples of these models include VSE [53], SCAN [107], OSCAR [116], CM3 [2], and VisualBERT [114]. They typically use pre-trained image encoders with lightweight text encoders that usually process simple tags/textual tokens. To model cross-modal interactions, they use a cross-attention Transformer [199].

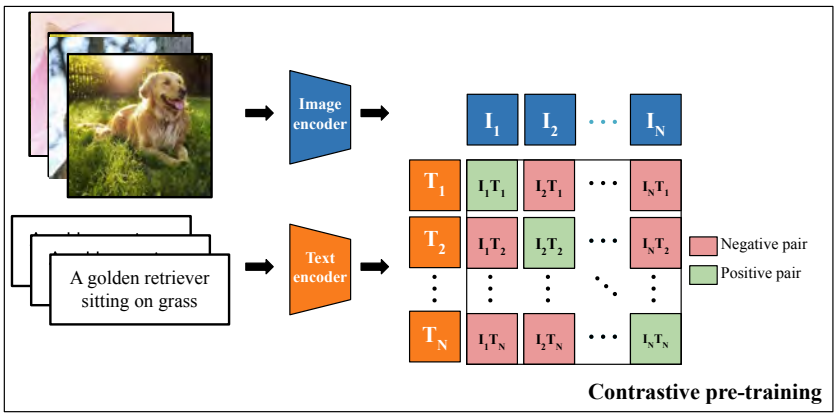
The Type 2.1c models concentrate entirely on modelling the intricate cross-modal interactions. They use lightweight encoder networks for the individual modalities and complex methods to model their interactions. Examples include OFA [206], SimVLM [209], FILIP [220], X-VLM [228], FLAMINGO [3] and ViLT [92].

Finally, models that come under Type 2.1d use expressive image and text encoders of similar capacities along with large Transformers for capturing cross-modal interactions. A few examples include FLAVA [181], ALBEF [113], BLIP [111], CoCa [224], TCL [218], METER [49], LOUPE [110] and MDETR [88].

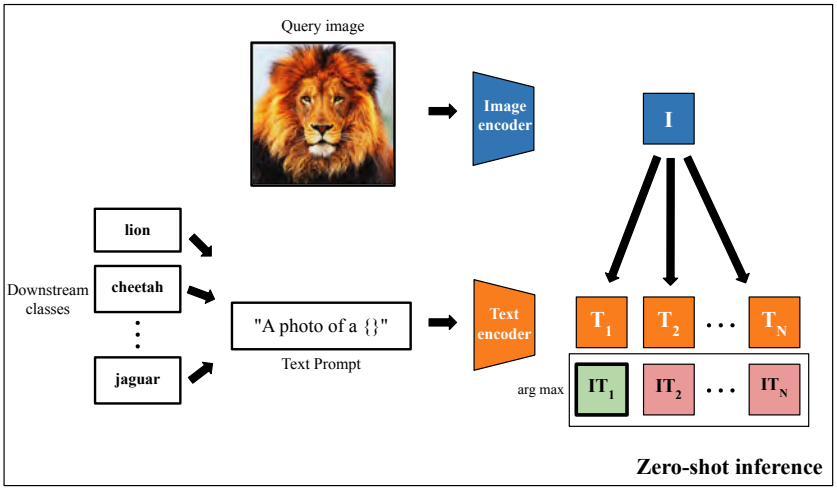
## 2.3 Contrastive Language-Image Pre-training (CLIP)

We now describe the vision-language model that is the focus of this thesis, CLIP. Radford et al. [165] introduced CLIP – A large-scale vision-language model that was trained on a massive corpus of 400M image-text pairs acquired from the internet, and exhibits exemplary downstream visual task transfer. CLIP extends upon the pre-training task of ConVIRT [238] – It's training objective is to maximise the similarities of the embeddings of paired image-text samples while minimising the similarities of unpaired samples.

**Training Objective.** Given a set of $N$ paired image-text samples, CLIP learns a joint image-text embedding space by training image and text encoders jointly. Both the encoders map their corresponding uni-modal inputs into the joint embedding space. Note that the embeddings of both images and texts are $l_2$-normalised. Therefore, the embedding space of CLIP effectively is a subspace on the unit hypersphere. Several works [204, 214, 153] have empirically shown the utility of working on the unit hypersphere justifying this design choice. By constructing a similarity matrix of size $N \times N$ (similarities of each image embedding with every text embedding), CLIP is trained to predict which of the $N \times N$ samples are the true paired-samples. This standard approach to contrastive learning boils down to a symmetric loss that involves two cross-entropy loss terms as shown below:




(a) Contrastive Pre-training




(b) Zero-shot Inference

Fig. 2.2 CLIP's training and zero-shot inference pipelines

$$
\begin{aligned}
L_{T\to I} &= -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(\langle T_i, I_i\rangle/\tau)}{\sum_{j=1}^{N}\exp(\langle T_i, I_j\rangle/\tau)} \\
L_{I\to T} &= -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(\langle I_i, T_i\rangle/\tau)}{\sum_{j=1}^{N}\exp(\langle I_i, T_j\rangle/\tau)} \\
L &= \frac{1}{2}\left[L_{I\to T} + L_{T\to I}\right]
\end{aligned}
\tag{2.1}
$$

where $\langle a, b\rangle = \frac{a^T b}{\|a\|\|b\|}$ represents the cosine similarity of two embeddings $a$ and $b$ and $\tau \in \mathbb{R}^+$ represents the temperature parameter. Since CLIP's embeddings are $l_2$-normalised, the cosine similarity of two embeddings is equal to their dot product. CLIP's training objective is illustrated in Figure 2.2a.

**Building a large dataset.** In spite of CLIP using a standard cross-modal contrastive loss that has been used before in the vision-language setting [238], it achieves tremendous performance on visual recognition tasks. One of the biggest driving factors of this performance gain is the massive and diverse vision-language dataset that was collected. Previous vision-language datasets were either too small (MS-COCO [119] with 88K image-text pairs/VisualGenome [97] with 101K image-text pairs/CC3M [176] with 3M image-text pairs) or annotated with poor quality captions (YFCC100M [195] with 100M image-text pairs). The authors of CLIP mitigated these issues by collecting a large-scale dataset from the web containing 400M diverse image-text pairs. For maximal semantic concept coverage, they manually curated a set of queries for searching image-text pairs on the web[2].

**Model Architectures and Training Protocol.** CLIP uses a modified Transformer [199] network as its text encoder. It uses a lower-cased byte-pair-encoding (BPE) [174] for pre-processing textual tokens. By bracketing all text sequences with start (<SOS>) and end (<EOS>) tokens, CLIP's text embedding is readily retrieved as the representation of the <EOS> token from the final layer of the Transformer. For CLIP's image encoder, it makes use of different variants of either a ResNet [70] or a Vision-Transformer [47]. CLIP uses linear projection heads on top of its text and image encoders for embedding the representations of both images and texts into the same space. These embeddings are then $l_2$-normalised.

For training, CLIP uses a large batch size of 32768 samples. Each variant is trained for 32 epochs with an Adam optimizer [93] and weight decay [124] with mixed precision [133] to accelerate training and improve memory constraints. Their largest models take up to 20 days to converge.

**Zero-shot Transfer.** One of the biggest paradigm shifts introduced by the CLIP model is its ability to perform zero-shot image classification. The term *zero-shot classification* is traditionally used to refer to models generalising to unseen classes [102, 132]. CLIP however transforms this idea into a classification setup where none of the dataset classes are known a-priori *i.e.* it extends the task to unseen datasets. This capability of CLIP is one of the main reasons for its pervasive use across domains.

Since CLIP was directly optimised to measure similarities between a given image and text pair, this ability is reused to perform zero-shot task transfer. For any given downstream classification task, the labels of the dataset can be directly converted into suitable captions. For example, if the classification task is "cats" vs "dogs", the labels can be converted into class-wise captions using a suitable textual prompt such as 'A photo of a <CLASS>' where the <CLASS> token is replaced by the corresponding class label ("cat" or "dog"). Using these class-wise captions, CLIP transforms the classification task into a simple image-caption matching task: For every test image, CLIP computes the similarity scores of the test image with every class caption, and the class with the maximal similarity is predicted. More concretely, CLIP generates a classifier weight matrix $W_{C\times d}$ by concatenating all the generated class caption embeddings. Here, $C$ denotes the number of classes and $d$ is CLIP's

[2]Schuhmann et al. [173] constructed a similar open-source dataset of approximately the same size and distribution called LAION-400M.

embedding dimension. Using $W$, CLIP conducts classification over $t$ test features $f_{t\times d}$ by producing logits using a matrix multiplication:

$$ZSL = fW^T \tag{2.2}$$

This entire zero-shot pipeline is depicted in Figure 2.2b.

## 2.4 Improving CLIP's few-shot image classification

One of CLIP's major strengths was that it demonstrated surprisingly robust few-shot image classification and text-to-image retrieval on a broad range of data distributions. However, to reach its full performance potential, fine-tuning on the target domain still appears to be necessary [165, 186, 211, 81]. Recently, several works [56, 232, 239, 193] have highlighted the potential of two techniques, *Prompt Learning* and *Adapters*, that may enable these models to achieve some of the benefits of fine-tuning without the associated computational costs. Figure 2.3 illustrates a high level overview of both these methods. In this section, we describe these two avenues for improving CLIP's few-shot classification.

### 2.4.1 Few-shot Learning Setup

We first describe the typical few-shot image classification setup [202, 184]. For a dataset containing $C$ classes, a $K$-shot dataset consists of $K$ labelled images per class. Therefore, the training set for this task contains $CK$ labelled samples, which is typically orders of magnitude smaller than a full-sized training dataset. The goal is to maximise classification performance on a test set by only using these $CK$ labelled samples.

Seminal works for solving the few-shot learning problem used attention-weights for linearly combining labels of the few-shot dataset [202] and methods to construct prototypes in a learned metric-space to compute query distances efficiently [184]. These methods improved classification performance with the added benefits of reduced training latency and increased data efficiency.

### 2.4.2 Prompt Learning

Rather than using manually engineered prompts for generating the class captions, prompt learning approaches aim to learn the optimal set of prompts by initialising learnable token vectors into the class caption prompt. These vectors are then trained using a cross-entropy loss directly on the few-shot dataset keeping CLIP's image and text encoders frozen. Once these prompt vectors are trained, classification can be conducted by constructing the weight matrix $W_{C\times d}$ and computing logits akin to Equation 2.2. Examples of such methods include CoOP [239], CoCoOp [240], DualCoOP [192], ProDA [126], ProGrad [241], CPT [221], UPL [78] and PromptTuning [217].

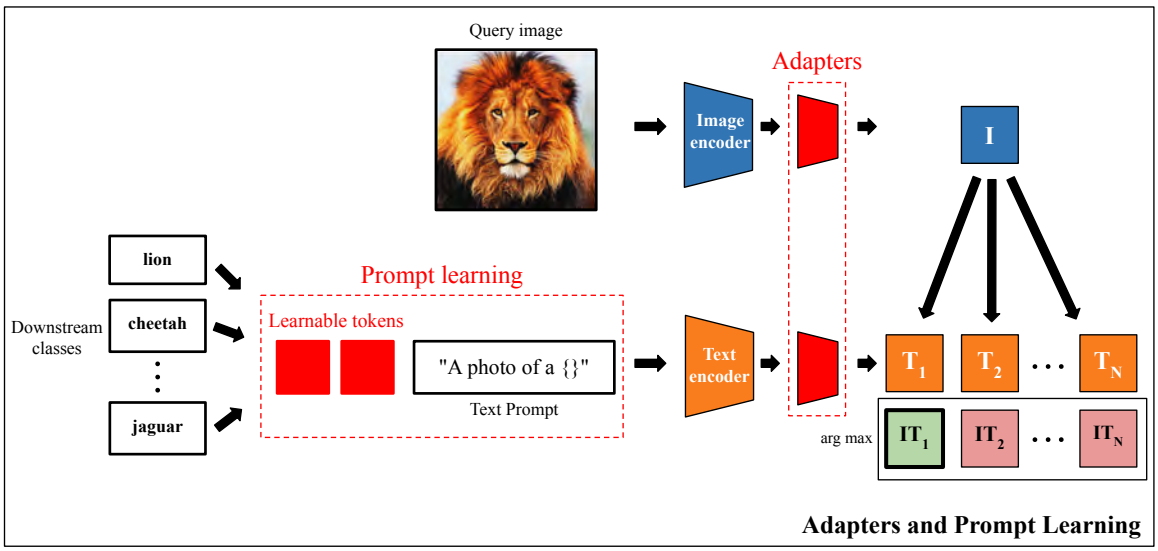



Fig. 2.3 Depiction of prompt learning and adapters for few-shot classification

### 2.4.3 Adapters

This class of models aims to train lightweight modules on top of CLIP's encoders so as to efficiently fine-tune CLIP in a way that enables balancing novel information (from the few-shot dataset) and world-knowledge (from CLIP's pre-training). Houlsby et al. [75] introduced these lightweight adapter modules for the first time in the domain of natural language processing. This led to a rapid proliferation in the use of such *Adapters* for meticulous adaptation of large pre-trained models to diverse downstream tasks across several domains [194, 158, 159, 219, 105, 193, 189, 125, 216, 34, 207, 40, 231].

Following their popularity, Gao et al. [56] proposed CLIP-Adapter that trained such *Adapter* modules over both the image and text encoders. They further noted that naively doing this fine-tuning might still lead to over-fitting. To mitigate this, they adopted residual connections to smoothly blend the fine-tuned knowledge from the few-shot dataset with the world knowledge from the pre-trained CLIP. Concretely, they trained image and text adapter networks $a_I$ and $a_T$ using a cross-entropy loss. The test features and the classifier weight matrix are updated using:

$$\begin{aligned} f_* &= \underbrace{\alpha a_I(f)}_{\text{New knowledge}} + \underbrace{(1-\alpha)f}_{\text{Pre-trained knowledge}} \\ W_* &= \underbrace{\beta a_T(W)}_{\text{New knowledge}} + \underbrace{(1-\beta)W}_{\text{Pre-trained knowledge}} \end{aligned} \tag{2.3}$$

Using these updated features and classifier weights, the logits for prediction are computed as:

$$CL = f_* W_*^T \tag{2.4}$$

where $\alpha$ and $\beta$ are residual ratios that balance the pre-trained and few-shot knowledge.

Zhang et al. [237] take this one step further by improving few-shot classification accuracy without the need for fine-tuning. Instead of training the image adapter, they directly set the weights of the adapter layer to be a set of affinities that are pre-computed between the test features and the few-shot dataset. Concretely, they embed the few-shot images using CLIP's image encoder, and call these image embeddings as cache keys $F_{CK\times d}$. They then convert each of the few-shot class labels to one-hot vectors, and call them cache values $L_{CK\times C}$. They compute similarities of the $t$ test features $f_{t\times d}$ with all the cache keys, which are then used as attention weights for the cache values. They then compose these weighted values with the zero-shot CLIP logits as their final predicted logits:

$$\begin{aligned} \overbrace{A}^{\text{Affinities}} &= \exp(-\beta(1-fF^T)) \\ TL = \underbrace{\alpha AL}_{\text{Few-shot knowledge}} &+ \underbrace{fW^T}_{\text{Pre-trained knowledge}} \end{aligned} \tag{2.5}$$

where $\beta$ controls the sharpness of the affinity distribution and $\alpha$ balances CLIP's pre-trained knowledge with the new few-shot knowledge. They further extend TIP-Adapter into the fine-tuning domain by training the adapter layer (initialised with the cache keys) using standard cross-entropy loss. This achieves state-of-the-art results for few-shot classification on 11 benchmark datasets.

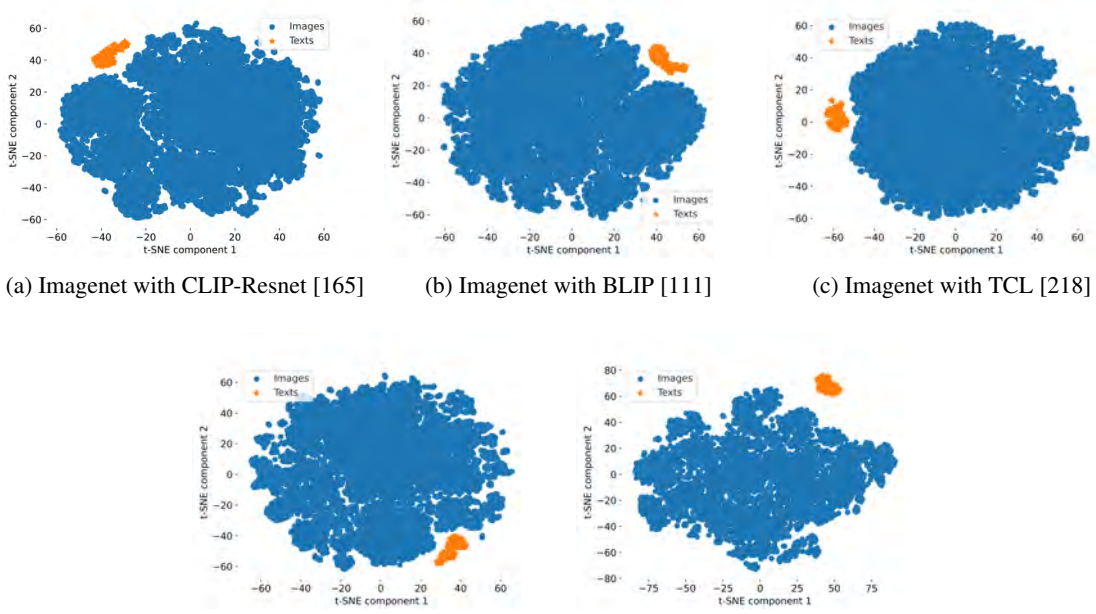
Apart from these two major methods, several other *Adapter*-based methods have been proposed that extend CLIP's capabilities to other task domains including 3D point-cloud understanding [233], dense prediction [167], video understanding and retrieval [148, 24, 11], depth understanding [236], image captioning [140, 15], and object detection [66, 226, 137].

# Chapter 3

# What is the modality gap?

In this chapter, we illustrate a ubiquitous but non-intuitive geometric phenomenon that occurs in the embedding spaces of vision-language models, termed the *Modality Gap*. This phenomenon has been studied in some detail in previous work by Liang et al. [118].

## 3.1 The Modality Gap




(a) Imagenet with CLIP-Resnet [165] (b) Imagenet with BLIP [111] (c) Imagenet with TCL [218]

(d) Imagenet with CLIP-ViT-B/16 [47] (e) Stanford Cars with CLIP-Resnet

Fig. 3.1 t-SNE visualisations of image-text embeddings across datasets (*Imagenet vs Stanford Cars*), contrastive pre-training methods (*BLIP vs CLIP vs TCL*) and model architectures (*Resnet vs ViT-B/16*). The blue points are image embeddings while orange points are text embeddings.

To demonstrate the modality gap phenomenon, we take the labelled test sets of two supervised classification datasets, Imagenet [169] and Stanford Cars [96], and encode their images and corresponding text captions using various CLIP-like models[1]. We then visualise these embeddings using a 2-dimensional t-SNE [198] projection. From Figure 3.1, we see that the image and text embeddings of CLIP (and its variants) are placed far-apart from each other in the shared multi-modal space. This phenomenon generalises across datasets, contrastive pre-training methods and model architectures. This is counter-intuitive to the expected structure of the embedding spaces of these models – since CLIP-like models use the contrastive loss (refer Equation 2.1) to maximise the similarity of paired image-text embeddings, we would expect that the paired image-text embeddings lie close together (thereby maximising their cosine similarity) while being separated from other paired image-text embeddings.

## 3.2 Pairwise Distances of Image-Text embeddings

A natural question arises on inspecting the t-SNE visualisations in Figure 3.1 – 'Is this visualisation an accurate depiction of the true embedding space or simply an artefact of the t-SNE method used for projecting the embeddings into 2 dimensions?'. Since the low dimensional embeddings obtained using t-SNE[2] only depend on the distance distribution of the high dimensional points, examining their pairwise-distances can help answer this question.

In Figure 3.2a, we plot the pairwise distances between high dimensional embeddings of 50000 image features and 1000 text features from Imagenet[3]. We see that the pairwise intra-modal distances are much smaller than the inter-modal distances *i.e.* the smallest image-text distances are likely larger than the largest image-image or text-text distances. This serves as evidence that the t-SNE method of visualisation is not a confounding factor but merely reflects the properties of the underlying multi-modal space. Since t-SNE trades-off distance preservation *vs* dimensionality reduction, the large high dimensional inter-modal distances are amplified in the low dimensional space, and therefore the text and image embeddings lie far apart in the visualisations.

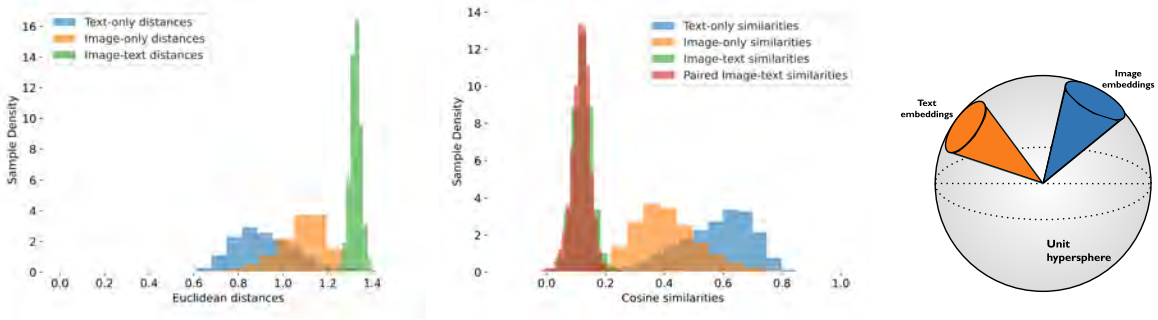
## 3.3 Narrow Cone Effect

Having analysed the distribution of distances of these embedding spaces, a natural next step is to analyse the intra-modal and inter-modal cosine similarities. Since during training we use a contrastive loss, we are directly optimising for a maximisation of the cosine similarities between paired image-text embeddings *i.e.* for example, an embedding of a dog image should ideally have a large cosine similarity ($\gtrapprox 0.5$) with an embedding of a text caption of a dog. Further, since these embeddings

[1] Since these datasets are supervised classification datasets, we use prompt ensembling as in [165] to create text captions corresponding to each test image by using their label information.

[2] A rigorous explanation of t-SNE can be obtained from the original paper [198].

[3] We use the 50000 Imagenet test set to plot the image features and use a prompt ensemble with 7 prompts for each of the 1000 Imagenet classes to plot the text features. For obtaining image and text features, we use the CLIP-Resnet model.




(a) Euclidean Distances (b) Cosine Similarities (c) Narrow Cone Effect

Fig. 3.2 Pairwise inter-modal and intra-modal euclidean distances (a), cosine similarities (b) and depiction of the narrow cone effect (c)

lie on the unit hypersphere, they are all unit vectors. Hence, maximisation of paired image-text cosine-similarities can only be achieved by aligning them as close together as possible *i.e.* reducing the angle between the paired image and text embeddings.

However, Figure 3.2b uncovers an interesting effect – the inter-modal image-text similarities are very sparsely distributed with a mean cosine-similarity of $\approx 0.1$. Upon analysing the cosine similarities of paired image-text samples, we observe a similar effect – despite the model being explicitly trained to maximise these paired image-text cosine similarities (the red region in Figure 3.2b), they are still significantly smaller than the intra-modal cosine similarities. This leads us to conclude that the image and text embeddings lie in two separate regions of the shared multi-modal embedding space, termed as the *Narrow Cone Effect* (refer Figure 3.2c). This effect has been studied extensively by Liang et al. [118]. They suggest that the effect is caused at model initialization by the independent two-tower architectural bias of these CLIP-like models – each modality encoder constrains all of its embeddings, regardless of the input, to a very *narrow cone* in the embedding space *i.e.* at initialisation itself, the average intra-modal cosine similarity for a given modality is very high. Hence, at initialisation, the shared multi-modal embedding space has two separate *narrow cones*, one for each modality, each with a very high average within-modality cosine similarity. They further showcase that the contrastive loss optimisation process is unable to coalesce these two *image and text cones*, and therefore even after the loss optimisation, the model is stuck with an embedding space that has two separate image and text cones (see Figure 3.2c). Under this new light, the modality gap phenomenon can be expressed as the distance between the two cones of the text and image embeddings.

## 3.4 A Simple Control Experiment and its Implications

Having established the narrow cone effect and the modality gap phenomenon, we perform a simple control experiment to modify the multi-modal embedding space of CLIP to make it more aligned. This is a further test to establish that the modality gap actually exists in the embedding spaces and

is not just a pathology of the t-SNE visualisation method. We apply a distance scaling method to determine if aligning the distributions of inter-modal and intra-modal distances yields more intuitive t-SNE visualisations.

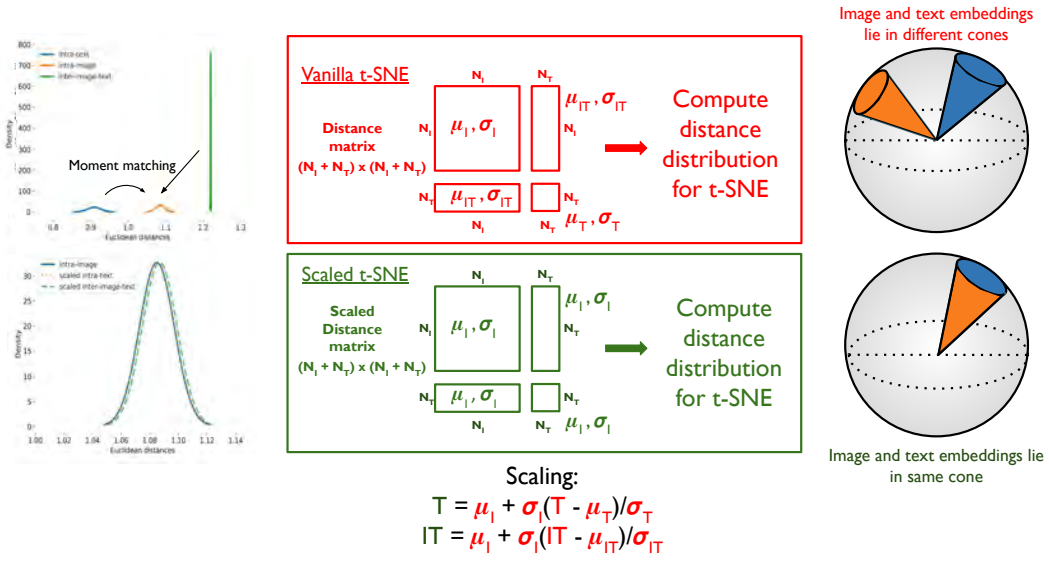



Fig. 3.3 Illustration of simple distance scaling method

We start with the observation that the t-SNE method requires pairwise distances between all possible pairs of points in the shared embedding space. For the plots in Figure 3.2, this yields a pairwise distance matrix of size $51000 \times 51000$ as there are 50000 image embeddings and 1000 text embeddings. From the previous section, we know that each modality-specific sub-matrix in this matrix has different distributions (see red box in Figure 3.3). To ensure that all distance distributions follow the same scale, we perform a crude moment matching by scaling the intra-text distances and the inter image-text distances to be exactly aligned with the distribution of intra-image distances[4]. We then use this modified distance matrix as input to t-SNE. As all the distances are on the same scale, we observe more aligned t-SNE plots (Figure 3.4), and the text and image embeddings are no longer farther apart from each other. Hence, this simple method can be used to visualise the image-text embeddings by removing the effect of the modality gap. We illustrate this simple distance scaling method in Figure 3.3.

This naive method of scaling therefore produces visualisations that are more intuitive and in line with our expectations of the structure of contrastive vision-language embedding spaces. Therefore, this experiment further confirms the existence of the modality gap and reiterates that the t-SNE method is not a confounder. However, this raises several questions about the modality gap and its implications on effective visualisation and downstream task performance:

[4]We match the first two moments by modelling each distance distribution as a Gaussian. The scaled distance matrix will therefore have sub-matrices with equal means and variances (see green box in Figure 3.3).

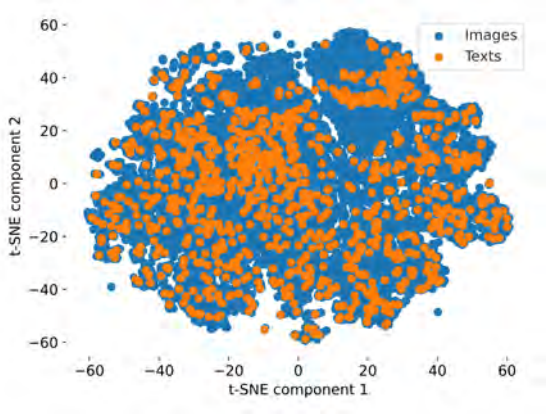



Fig. 3.4 t-SNE visualisation using the simple distance scaling method

1. Why does the modality gap exist at all? What are the factors in the contrastive loss that lead to this phenomenon?
2. Are there effective and principled ways to visualise the image-text embeddings that are both understandable and reflective of the true underlying space?
3. What are the implications of the modality gap on the zero-shot downstream capabilities of these models?

We aim to answer each of these questions in detail in the subsequent chapters.

# Chapter 4

# Why is there a modality gap?

In the previous chapter, we established the existence of a modality gap between image and text embeddings of CLIP-like models. In this chapter, we aim to decompose the contrastive loss to identify the most prominent factors affecting the modality gap. We formulate several toy experiments to illustrate the principles underlying this phenomenon. We then extend these toy analyses to real world image-text datasets. To ensure readability, we end each section with the key results from that section. We begin by reviewing the contrastive loss, and identifying its different moving components.

## 4.1 Revisiting and Abstracting the Contrastive Loss

The contrastive loss used by CLIP and its variants aims to maximise the cosine similarities of paired image-text samples while minimising the cosine-similarities of unpaired image-text samples. Consider $N$ $d$-dimensional paired image-text embeddings. $I = \{I_1, I_2, \ldots, I_N\}$ is the set of image embeddings and $T = \{T_1, T_2, \ldots, T_N\}$ is the set of text embeddings. Note that all embeddings lie on the unit hypersphere *i.e.* $I_i \in S^{d-1}$ and $T_i \in S^{d-1}, i \in \{1, 2, \ldots, N\}$, and are hence unit-norm vectors[1]. The contrastive loss for these sets of embeddings is formulated as:

$$\begin{aligned} L_{T\to I} &= -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(T_i\cdot I_i/\tau)}{\sum_{j=1}^{N}\exp(T_i\cdot I_j/\tau)} \\ L_{I\to T} &= -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(I_i\cdot T_i/\tau)}{\sum_{j=1}^{N}\exp(I_i\cdot T_j/\tau)} \\ L &= \frac{1}{2}\left[L_{I\to T} + L_{T\to I}\right] \end{aligned} \tag{4.1}$$

where $\tau$ denotes the temperature and $A \cdot B$ denotes the dot product of vectors $A$ and $B$.

We now discuss the factors that are crucial for optimising the contrastive loss thereby directly impacting the geometry of the embedding space:

[1] We use the standard notation $S^{d-1}$ to represent the $d$-dimensional unit hypersphere, following [74, 130].

1. ***Batch Size*** $N$**.** The batch size modulates the difficulty of the learning task. For larger batch sizes, the $N$-way classification task of picking the true paired sample out of the $N-1$ negative samples is much more difficult compared to smaller batch sizes.

2. ***Embedding Dimension*** $d$**.** The dimensionality of the embedding space plays a key role in its geometric properties. Since there are more degrees of freedom in higher dimensional spaces as compared to the constraints present in lower dimensional spaces [95], this could be a trivial yet fundamental forming factor for the modality gap.

3. ***Temperature*** $\tau$**.** The temperature can be used to modulate the entropy of the image-text similarity distribution. Small temperatures can amplify minute differences in similarity whereas large temperatures produce flat distributions tending to the uniform distribution.

4. ***Mismatch Ratio*** $M$**.** We define the mismatch ratio as the ratio of number of mismatches divided by total number of samples. We call an image-text pair a mismatch if the cosine similarity of the image embedding with any aribitrary text embedding is larger than with its true paired text embedding. The mismatch ratio acts as a proxy for downstream task performance.

5. ***Alignment*** $A$**.** The alignment between the image-text embeddings controls the distribution of the pairwise inter-modal cosine similarities. The modality gap phenomenon arises due to the alignment of the image-text embeddings being small.

6. ***Uniformity*** $U$**.** The uniformity of the image-text embedding space controls spread of the embeddings on the unit hypersphere. Intuitively, the more spread out the embeddings are, the easier it is to cluster points semantically.

In each subsequent section, we list all our key results in terms of these six factors.

## 4.2 Toy Problem with Points on the Unit Circle

Having identified the main factors that affect the contrastive loss, we set up a simple toy problem that abstracts away several of the moving components in the loss, and simply works with points on the unit circle $S^1$. By working with points in 2 dimensions, we can gain an intuitive understanding of the properties of the contrastive loss.

Assume two fixed image points on the unit circle $S^1$, $I_1 = \begin{bmatrix} i_{1x} & i_{1y} \end{bmatrix}^T$ and $I_2 = \begin{bmatrix} i_{2x} & i_{2y} \end{bmatrix}^T$. Our goal in this toy problem is to analytically derive the two text points $T_1 = \begin{bmatrix} t_{1x} & t_{1y} \end{bmatrix}^T$ and $T_2 = \begin{bmatrix} t_{2x} & t_{2y} \end{bmatrix}^T$ on the unit circle that minimise the contrastive loss for different settings of $I_1$ and $I_2$ (see Figure 4.1). An analytical derivation of the optimal text points $T_1$ and $T_2$ given a fixed set of image points $I_1$ and $I_2$ can be found in Appendix A.

Through this setup, we aim to decouple the effects of two main parameters: (1) $d$, the distance between the two image points $I_1$ and $I_2$[2], and (2) $\tau$, the temperature. Note that $d$ is a simple proxy for *Uniformity*.

We run this toy experiment for different settings of $d$ and $\tau$ to understand their implications, on the contrastive loss, and on the placement of optimal text points $T_1$ and $T_2$[3].

### 4.2.1 The effects of the distance between image points

We use 4 different initial configurations of the image points by varying $d$ as shown in Figure 4.1. For each configuration, we solve for the optimal text points and visualise them in Figure 4.1. In Table 4.1, we report the contrastive loss $L$ obtained for each setting with the optimal text points. We also report the loss obtained if we manually set $T_1$ and $T_2$ to be equal to $I_1$ and $I_2$ *i.e.* if we exactly align the image and text points. We denote this loss as $L_{align}$. Note that we fix $\tau = 1$ for the loss computation in this experiment.

| $d$ | Optimal $T_1$ | Optimal $T_2$ | $L$ | $L_{align}$ |
|---|---|---|---|---|
| 1 | $\left(-\frac{\sqrt{3}}{2}, \frac{1}{2}\right)$ | $\left(\frac{\sqrt{3}}{2}, -\frac{1}{2}\right)$ | 0.313 | 0.474 |
| $\sqrt{2}$ | $\left(-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$ | $\left(\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}\right)$ | 0.217 | 0.313 |
| $\sqrt{3}$ | $\left(-\frac{1}{2}, \frac{\sqrt{3}}{2}\right)$ | $\left(\frac{1}{2}, -\frac{\sqrt{3}}{2}\right)$ | 0.163 | 0.201 |
| 2 | $(0, 1)$ | $(0, -1)$ | 0.126 | 0.126 |

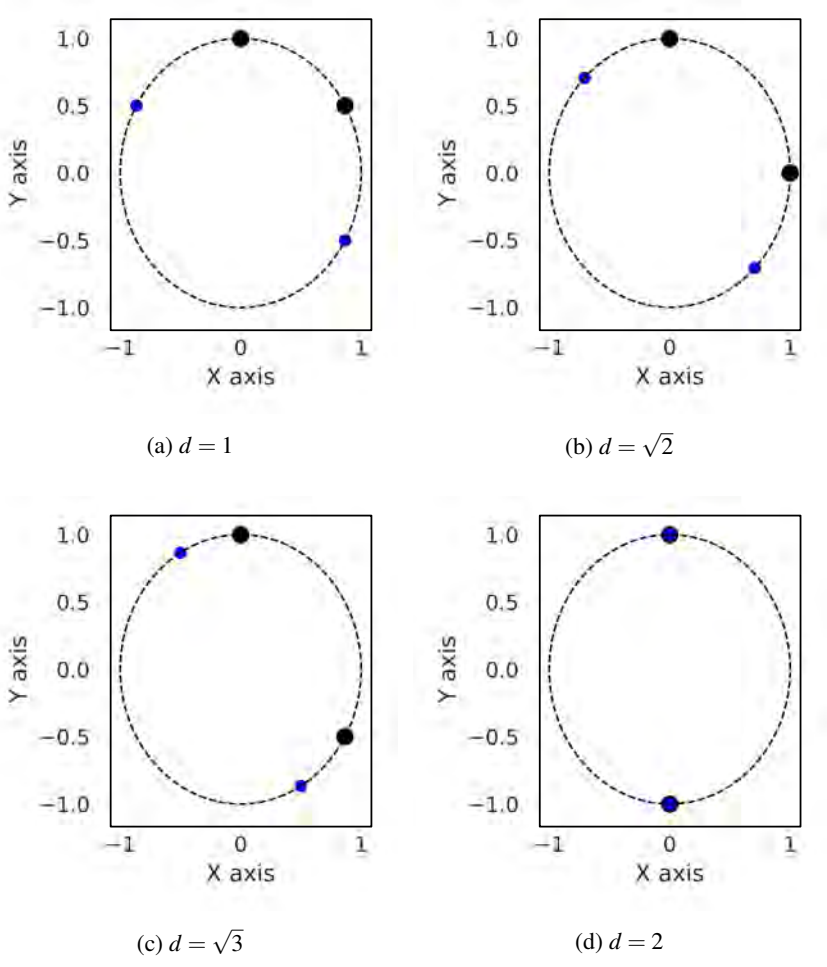
Table 4.1 Optimal text points and the losses obtained for different values of $d$. $L_{align}$ is the loss obtained when the image and text points are exactly aligned with each other.

We observe that as the distance between the two image points increases, we get a smaller loss value. Further, when the two image points are as far apart as possible on the unit circle (Figure 4.1d), the loss is minimised at exact alignment of the image and text points. Therefore, this simple toy experiment provides us with some initial clues about the behaviour of the contrastive loss with respect to the configuration of image and text points on the unit circle. One curious point to note is that $d$ controls the alignment[4] between the optimal text points and the fixed image points. Figure 4.2 makes this observation concrete – the distance between the optimal text points and the image points shrinks as $d$ grows *i.e.* the image and text points get more aligned when the image points themselves are farther away from each other. Since $d$ represents *Uniformity*, this result implies that for this setting, *Uniformity* and *Alignment* go hand in hand – at the optimal loss, the more uniformly spread the two points are on the unit circle, the more aligned the image and optimal text points are.

[2]We compute $d$ as the euclidean distance between the two image points, $d = \sqrt{(i_{1x} - i_{2x})^2 + (i_{1y} - i_{2y})^2}$

[3]We refer to the text points that minimize the loss at any given setting of $d$ and $\tau$ as the optimal text points

[4]We measure alignment between text and image points as: $\frac{\sqrt{(i_{1x}-t_{1x})^2+(i_{1y}-t_{1y})^2}+\sqrt{(i_{2x}-t_{2x})^2+(i_{2y}-t_{2y})^2}}{2}$




(a) $d = 1$ (b) $d = \sqrt{2}$

(c) $d = \sqrt{3}$ (d) $d = 2$

Fig. 4.1 **Image points and optimal text points.** Image points are represented in black and text points are represented in blue.

### 4.2.2 The effects of temperature

For each of the settings studied in the previous section, we analyse the effect of different temperature values on the loss and the configuration of the optimal text points. We linearly sweep through 1000 values of $\tau$ from 0.001 to 1. For each value, we measure the alignment between the fixed image points and the optimal text points. We also analyse the loss values across temperatures. We first compute the loss when we use the optimal text points, and term that as *optimal loss*. Next, we compute the loss incurred when the image points and text points are exactly aligned ($L_{align}$ from the previous section), and call it *aligned loss*. By comparing the aligned and optimal losses, we hope to decipher the key relationships between *Temperature*, *Alignment* and *Uniformity* (initial distance between the image points), for this toy setting (see Figure 4.3).

Evidently, at low temperatures ($\tau \in [0, 0.2]$) we get perfect alignment *i.e.* the image and text points lie exactly on top of each other. However, there is a steep decrease in the alignment as we

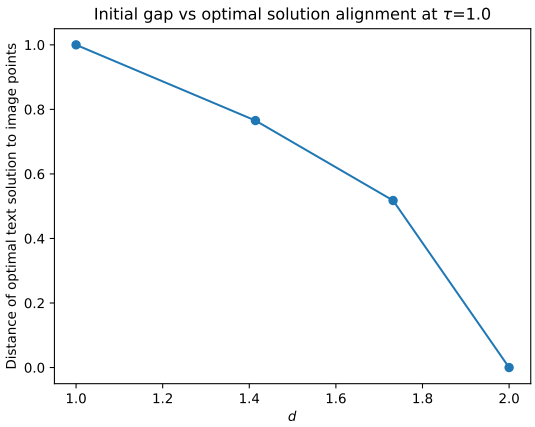
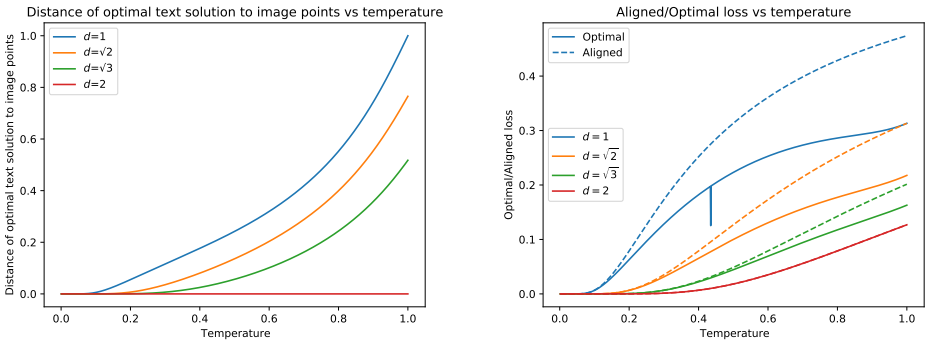



Fig. 4.2 $d$ vs alignment of text and image points




(a) Alignment vs temperature

(b) Loss vs temperature

Fig. 4.3 Effect of temperature on alignment between image-text points and loss values. 'Optimal' refers to the loss values computed by using the optimal text points whereas 'Aligned' refers to the loss values computed by exactly aligning the text and image points.

increase the temperature (Figure 4.3a). This observation is further bolstered by the loss landscape at low temperatures. Figure 4.3b depicts the deviation between the aligned loss, $L_{align}$, and the optimal loss, $L$. We see that at higher temperatures, this deviation grows sharply, and this effect is exacerbated with smaller $d$. $L$ and $L_{align}$ converge at two points: (i) when $\tau$ is very low, and (ii) when $d$ is high. This empirical evaluation therefore suggests that for a given setting of fixed image points, the alignment between image-text pairs is negatively correlated with temperature.

This toy experiment therefore provides us with some intuition about the contrastive loss, and its sensitivity to temperature and alignment of the image-text points. However, since this setting is limited (in terms of dimensionality and number of points), it is not clear if these relationships will hold at higher dimensions with large batch sizes. This experiment was conducted *only* to get an

intuitive sense of the optimal alignment of text points with the image points under the contrastive loss at different settings, and further tests are required to generalise these inferences.

**Key Results.**

1. *Uniformity* is desirable for incurring the lowest loss across different settings.
2. *Uniformity* and *Alignment* are correlated regardless of *Temperature*.
3. *Temperature* is negatively correlated with *Alignment*, and the strength of this correlation is modulated by *Uniformity*.

## 4.3 Reproducing the Modality Gap in Simulation Space

Taking the results from the toy problem as a starting point, we aim to understand how the different moving components interact as the complexity of the problem increases.

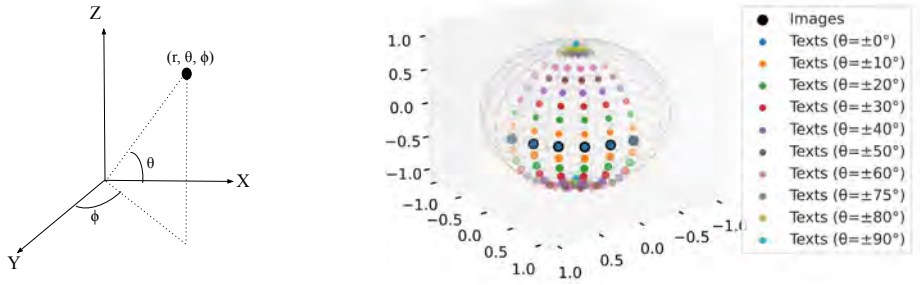
Liang et al. [118] conducted one such empirical study where they investigated the existence of the modality gap using a simple loss probing analysis and 3D toy simulations. We first aim to reproduce and extend their results, and then discuss their implications.

**Setup.** We begin with some notation (see Figure 4.4a). We represent any point $x$ on $S^2$ as (1, $\theta$, $\phi$). $\theta$ is the angle that the projection of $x$ onto the $XZ$-plane makes with the $X$-axis. $\phi$ is the angle that the projection of $x$ onto the $XY$-plane makes with the $Y$-axis.

The experimental setup of Liang et al. [118] consisted of 6 image-text embedding pairs on $S^2$. They sample the image embeddings on the equatorial line of the sphere (*i.e.* $\theta = 0°$), all separated by $\phi = 15°$. They then sample text embeddings with the exact same $\phi = 15°$ as the image embeddings. To simulate alignment between image and text embeddings, they alter the $\theta$ of the text embeddings. The entire setup is succinctly represented in Figure 4.4a. Having set up the image-text embeddings, they proceed to simulate mismatches by simply swapping the first two text embeddings with each other (See Figure 4.4b). Using this, they study the effects of mismatches on the alignment through expected loss plots. Through the lens of our initial abstraction, we can view $\theta$ as controlling *Alignment* and $\phi$ as controlling *Uniformity*. However, the experiments conducted by Liang et al. [118] only vary the *Alignment* while keeping the *Uniformity* fixed. Therefore, we extend their analysis by additionally modifying $\phi$ (thereby modifying *Uniformity*) and studying its effects on the contrastive loss.

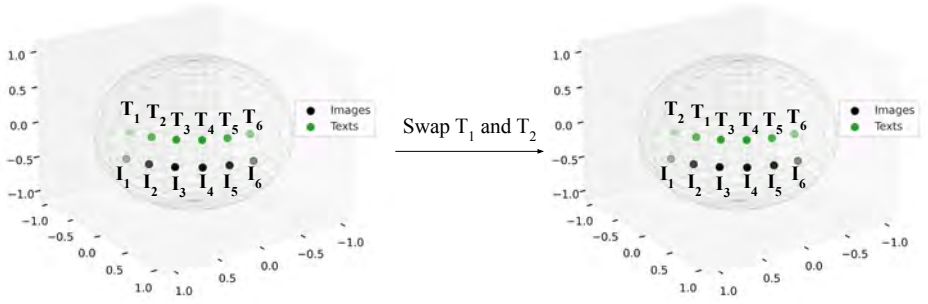
***Mismatches* create a temperature-dependent repulsive structure.** We first reproduce the main result of Liang et al. [118] in Figure 4.5[5]. We observe that when there are no mismatches, the contrastive loss is minimised when $\theta$ is small *i.e.* the optimisation of the contrastive loss prefers alignment of image-text embeddings when there are no mismatches (Figure 4.5a). However, when

[5]We discover a minor inaccuracy in the paper's simulation setup. The results shown in the original paper are exactly reproduced when we use $\phi = 20°$ and not $\phi = 15°$ as noted in the paper. However, this does not undercut their main claim as the exact results still hold at $\phi = 20°$.




Coordinate system for sampling Samples using different θ

(a) Image and text samples for simulation




Setup with no mismatches Setup with mismatches

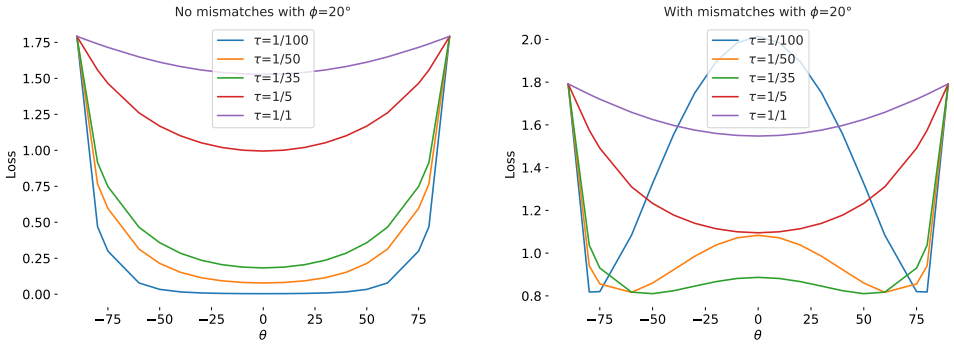
(b) Simulation of mismatches

Fig. 4.4 **Simulation setup for modality gap reproduction.** In our case, $r = 1$ since we are working with points on the unit hypersphere. Given $\theta$ and $\phi$, the exact image point coordinates are $(\sin\phi, \cos\phi, 0)$ and the exact text point coordinates are $(\cos\theta\sin\phi, \cos\theta\cos\phi, \sin\theta)$.

we introduce mismatches, a temperature-dependent repulsive structure emerges in the loss-landscape (Figure 4.5b). At low temperatures ($\tau = \{\frac{1}{100}, \frac{1}{50}, \frac{1}{35}\}$), the loss is minimised when the image-text embeddings are not aligned whereas at high temperatures ($\tau = \{\frac{1}{5}, 1\}$), the loss minimisation still prefers alignment of the image-text embeddings.

**$\phi$ modulates the repulsive structure.** We now extend these results in Figures 4.6 and 4.7 by depicting the loss landscape plots at various $\phi$ values. It is evident that $\phi$ plays a major role in the resulting loss structure. The temperature-dependent repulsive structure observed previously is now modulated by $\phi$: At small $\phi$, the repulsive structure disappears and the loss structures both with and without mismatches appear similar (Figure 4.6a, 4.6b, 4.7a, 4.7b). Contrarily, at large $\phi$s, the repulsive structure with mismatches is exacerbated – at $\phi = 60^\circ$ (Figure 4.7d), even a high temperature of $\tau = \frac{1}{5}$ experiences a small but significant repulsive structure thereby causing the image and text embeddings to lie far apart from each other.

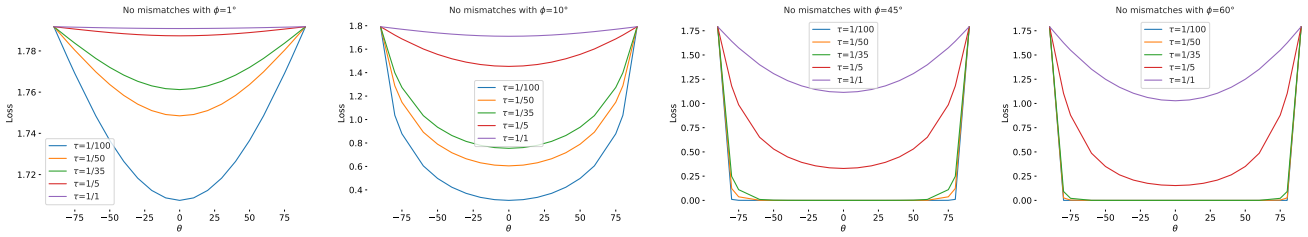
***Uniformity* is only desirable when there are no mismatches.** From Figure 4.6, we observe that for a given temperature when there are no mismatches, the lowest losses are incurred at large




(a) No mismatches (b) With mismatches

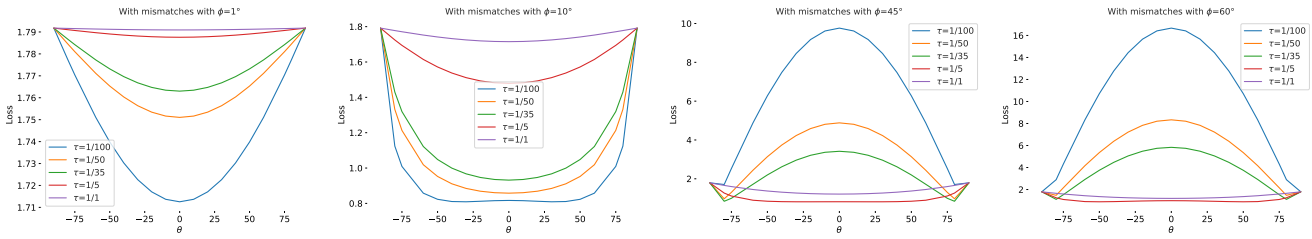
Fig. 4.5 Main result of Liang et al. [118] at $\phi = 20°$

$\phi$s. Further, this lowest less is incurred over a wider range of $\theta$s when $\phi$ is large. This indicates that when there are no mismatches, uniformity is a desirable property for the embedding space. However, Figure 4.7 suggests that the contrastive loss rapidly increases as $\phi$ increases. This is especially prominent at low temperatures. A simple justification of this phenomenon is that the strength of the mismatches increases as we increase $\phi$. Hence, the loss penalises these mismatches more severely, especially at low temperatures.



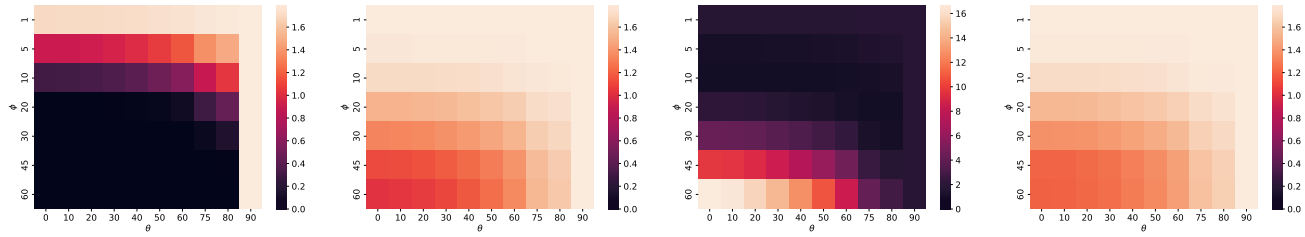

(a) $\phi = 1°$ (b) $\phi = 10°$ (c) $\phi = 45°$ (d) $\phi = 60°$

Fig. 4.6 Loss landscape without mismatches at different $\phi$s




(a) $\phi = 1°$ (b) $\phi = 10°$ (c) $\phi = 45°$ (d) $\phi = 60°$

Fig. 4.7 Loss landscape with mismatches at different $\phi$s



(a) $\tau = 0.01$, no mismatches (b) $\tau = 1$, no mismatches (c) $\tau = 0.01$, with mismatches (d) $\tau = 1$, with mismatches

Fig. 4.8 Loss heatmaps depicting $\phi$ v/s $\theta$ with and without mismatches at $\tau = 0.01$ and $\tau = 1$. Darker cells denote smaller loss values and lighter cells denote larger loss values. Therefore, darker is better in these plots. Results with more temperatures can be found in Appendix B.

**High *Temperatures* always prefer *Alignment*.** The heatmaps in Figure 4.8 concretely showcase that at high temperatures (Figure 4.8b, 4.8d), the contrastive loss is minimised at perfect alignment. Contrarily, at low temperatures with mismatches (Figure 4.8c), the loss structure promotes misalignment *i.e.* the image and text embeddings to be far apart. Therefore, low temperatures coupled with the existence of mismatches foster the modality gap.

**Limitations of this simulation setup.** Despite this simulation setup being useful for studying the effects of *Alignment*, *Uniformity* and *Temperature* on the contrastive loss, it is limited in its scope and not representative of real-world settings (due to a small fixed batch size of 6, small mismatch ratio and uniform intra-modal distances with fixed $\phi$). This calls into question the transferability of these results to real word settings which are likely more stochastic and unstructured. Hence, in the next section, we introduce a more representative simulation setup that uses random sampling of points on the hypersphere to study these effects.

**Key Results.**

1. The *Uniformity* results from the toy problem (Section 4.2) only hold when there are no *Mismatches*.
2. High *Temperature* promotes *Alignment* – this result is opposite to the one obtained from the toy problem.
3. *Mismatches* and *Temperature* play a large role in promoting the *Modality Gap*.

## 4.4 A More Representative Simulation Framework

To isolate the effects of each individual factor outlined in Section 4.1, we formulate a simple sampling framework that allows us to alter each factor independently and thus simulate diverse settings of real-world image-text embedding spaces. We model image and text embeddings on the unit hypersphere

using a Power Spherical distribution [41] which is a stable approximation to the well-known von Mises–Fisher distribution [210, 129][6].

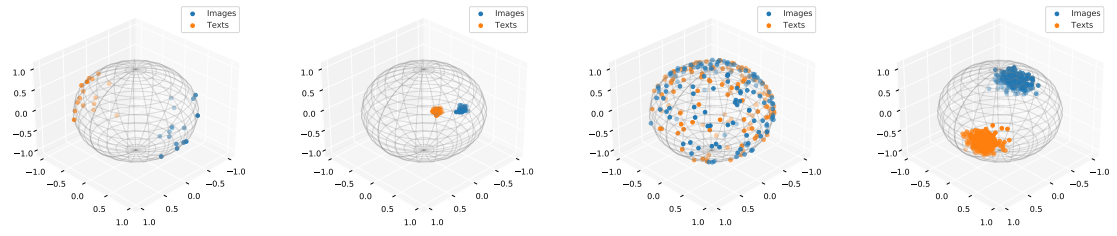
Our sampling algorithm takes as inputs embedding dimension $d$, batch size $N$, alignment angle $\theta$, and concentration $\kappa$. The concentration of the Power Spherical distribution denotes the variance of the angular distribution on the sphere – the larger the concentration, the closer together the samples on the sphere are. Following our initial abstraction from Section 4.1, $\theta$ represents *Alignment* and $\kappa$ represents *Uniformity*.

We first sample two random vectors from a standard multivariate Gaussian to denote the means of the image and text embeddings. We then align the text embedding to the image embedding at an angle $\theta$ by using a simple projection operation. We then sample $N$ image and $N$ text embeddings using the computed means and fixed concentration $\kappa$. Our sampling algorithm is depicted in Algorithm 1. We showcase a few samples on $S^2$ generated by our algorithm in Figure 4.9.

**Algorithm 1:** Generating image and text embeddings

**Input:** Batch size $N$, Dimension $d$, Alignment angle $\theta$, Concentration $\kappa$
**Output:** Image embeddings $I \in S^{N\times d-1}$, Text embeddings $T \in S^{N\times d-1}$

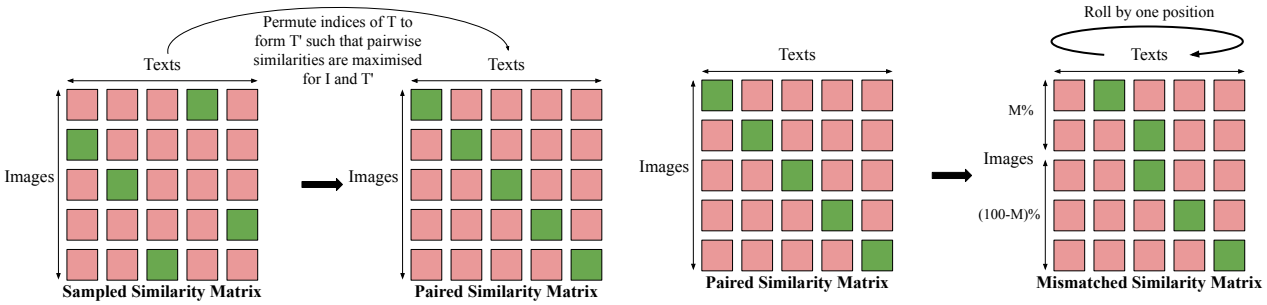
sample mean image embedding: $i \sim \mathcal{N}(0, I_d)$ `// I_d is d-dim identity matrix`
normalise $i$ onto hypersphere: $i \leftarrow \frac{i}{\|i\|_2}$
sample random embedding: $r \sim \mathcal{N}(0, I_d)$
compute projection: $p = r - (r \cdot i)i$ `// ensure` $p \perp i$
normalise $p$ onto hypersphere: $p \leftarrow \frac{p}{\|p\|_2}$
compute text embedding: $t = i\cos\theta + p\sin\theta$ `// ensure` $\langle i, t\rangle = \theta$
sample: $I \overset{N}{\sim} \text{PowerSpherical}(i, \kappa)$
sample: $T \overset{N}{\sim} \text{PowerSpherical}(t, \kappa)$
**return** $I$, $T$




(a) $N = 16, \kappa = 10, \theta = 125°$ (b) $N = 32, \kappa = 1000, \theta = 25°$ (c) $N = 128, \kappa = 1, \theta = 0°$ (d) $N = 256, \kappa = 100, \theta = 180°$

Fig. 4.9 **Generated image and text samples on $S^2$.** Higher $\kappa$ leads to samples being more concentrated around the mean whereas lower $\kappa$ leads to more uniformity on the sphere.

[6]Both the Power Spherical and von Mises-Fisher (vMF) distributions define probability distributions on the unit hypersphere. The vMF distribution is widely regarded as the analogue of the Gaussian distribution on the unit hypersphere. Refer to [190] for a detailed review of sampling from these distributions.




(a) Pairing operation

(b) Mismatch simulation operation

Fig. 4.10 The pairing and mismatch simulation operations for the expected loss computation

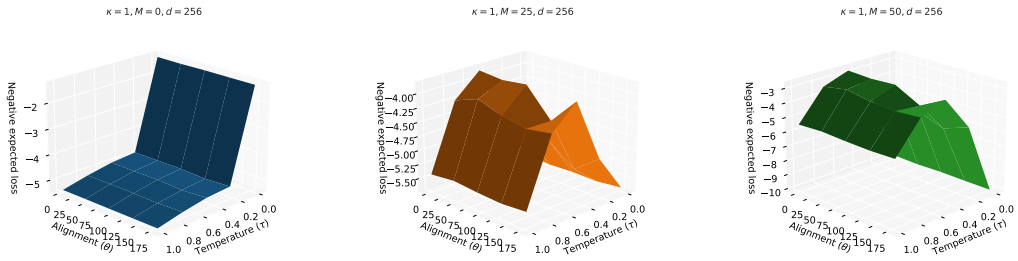
Equipped with image and text embeddings that mimic real world samples, we now describe our method for computing the expected contrastive loss incurred by the sampled embeddings under different temperatures and mismatch ratios. Given sampled image embeddings $I$ and text embeddings $T$, we simulate paired samples by permuting the indices of $T$ to match $I$ such that each paired image-text embedding are maximally similar to each other. This means that the cosine similarity matrix containing all cross-similarities between $I$ and $T$ will have maximal values on the diagonal (Figure 4.10a). Then, to simulate a mismatch ratio of $M$, we simply roll all columns by one position for the top $M\%$ rows in the similarity matrix while preserving the bottom $(100-M)\%$ rows as is (Figure 4.10b). We then use the new mismatched similarity matrix to compute the expected contrastive loss. We estimate the expected loss for a particular setting by simulating 100 runs with different random samples. Algorithm 2 shows our expected loss computation for a given setting of temperature and mismatch ratio. For studying the effects of each factor, we run several simulations by varying the factors of interest. We keep a large fixed batch size of $N = 256$ to represent realistic simulations. See Appendix B for more details.

**Algorithm 2:** Expected loss computation

**Input:** Batch size $N$, Dimension $d$, Temperature $\tau$, Mismatch Ratio $M$, Alignment angle $\theta$, Concentration $\kappa$

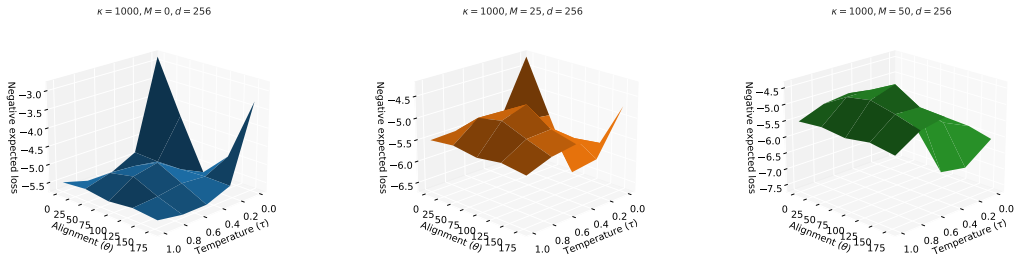
**Output:** Expected loss $L$ for given input settings

$L$=0
**for** $k = 1$ **to** 100 **do**
    generate $I$, $T$ using Algorithm 1 with inputs $N$, $d$, $\theta$, $\kappa$
    perform pairing and mismatch simulation operations with mismatch ratio $M$ as shown in Figure 4.10
    use mismatched similarity matrix to compute contrastive loss $l$ with temperature $\tau$ from Equation 4.1
    $L$+=$l$
**return** $\frac{L}{100}$




(a) $M = 0$ (b) $M = 25$ (c) $M = 50$

Fig. 4.11 Loss landscape at high *Uniformity* ($\kappa = 1$)




(a) $M = 0$ (b) $M = 25$ (c) $M = 50$

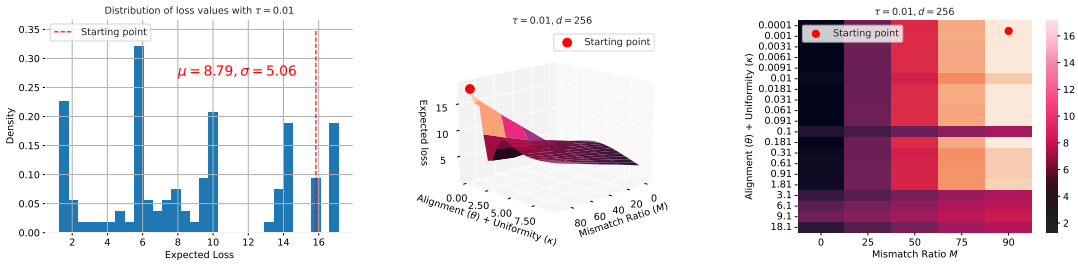
Fig. 4.12 Loss landscape at low *Uniformity* ($\kappa = 1000$)

**Low *Uniformity* facilitates the Modality Gap.** To pinpoint the exact factors causing the modality gap, we specifically search for settings in our simulation where the losses incurred by smaller $\theta$s are bigger than those for larger $\theta$s. For ease of analysis, we plot negative expected loss for all experimental settings, therefore higher is better for all our plots. We note from Figures 4.11 and 4.12 that low *Uniformity* (*i.e.* large $\kappa$) seems to play the most prominent role in determining the mis-alignment of the image-text embeddings (*i.e.* large $\theta$). This observation is agnostic to embedding dimension $d$ (Figures B.6, B.7, B.8 and B.9) and mismatch ratio $M$ (Figures 4.12 and B.2). This finding backs our previous key results on *Uniformity* and *Alignment* from Sections 4.2 and 4.3.

### 4.4.1 Simulating a training run at different temperatures.

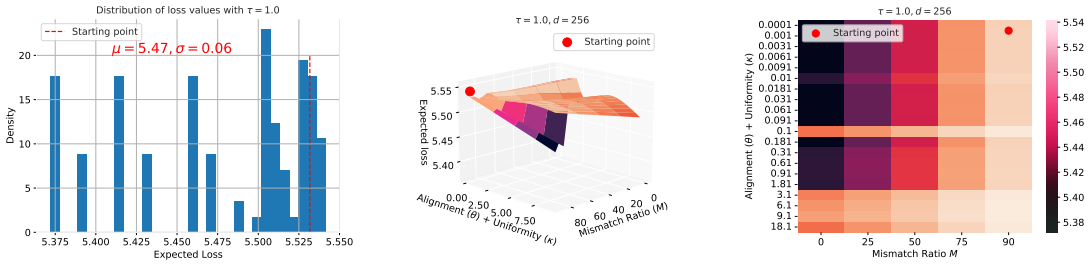
We now aim to simulate a training run across different temperatures by mirroring the settings of an initial CLIP model. To replicate as close a real world-setup as possible, we use batch size $N = 256$ and dimensionality $d = 256$ in all our training simulations.

Liang et al. [118] concretely showed that the modality gap exists at model initialisation due to the *Narrow Cone Effect* (Section 3.3). This implies that at model initialisation, there is low *Uniformity* and low *Alignment*. Further, we make the assumption that since model initialisation is random, our model starts off with a large mismatch ratio $M$. Incorporating these model initialisation heuristics, we start off all our training simulations at $\kappa = 100$, $\theta = 60^\circ$ and $M = 90$. Since in a real-world training run




(a) Loss distribution (b) Loss landscape (Azimuth view) (c) Loss landscape (Top view)

Fig. 4.13 Expected loss dynamics at $\tau = 0.01$




(a) Loss distribution (b) Loss landscape (Azimuth view) (c) Loss landscape (Top view)

Fig. 4.14 Expected loss dynamics at $\tau = 1.0$

we fix the temperature prior to training, we simulate different training runs for different temperatures. For each simulation, we plot the distribution of loss values that can be achieved starting from our heuristic model initialisation point. We also plot the the expected loss landscape of the model[7].

**Global minimum occurs at high *Uniformity* and no *Mismatches*.** From our training simulations, we sort the expected losses in descending order, and report the settings of the 10 lowest losses for every temperature. We report these values in Tables B.2, B.3, B.4, B.5 and B.6 in the Appendix due to lack of space. We observe that regardless of temperature, the expected loss is lowest for settings where the *Uniformity* is high and the *Mismatch Ratio* is 0.

**High temperatures close the modality gap.** We observe that at low temperatures, the expected loss distributions have a high variance (Figures 4.13a and B.10a). However, this variance shrinks rapidly as we increase the temperature (Figures B.11a, B.12a and 4.14a). Further, the mean expected loss at lower temperatures is also much larger than at higher temperatures. This indicates that (1) at lower temperatures, the gradients of the loss are much stronger than at higher temperatures, and (2) at higher temperatures, the magnitude of the difference between the initial and final losses of the model is much smaller than at lower temperatures. Further, from Figure 4.14 we see that the loss landscape

[7] In all our loss landscape plots, we plot the landscape as a function of *Mismatch Ratio* ($M$) and a function of *Alignment* and *Uniformity* which we simply denote as *Alignment* ($\theta$) + *Uniformity* ($\kappa$). We define this as: *Alignment* ($\theta$) + *Uniformity* ($\kappa$) = $\frac{\kappa+\kappa\times\theta}{10^4}$

at a high temperature ($\tau = 1.0$) is very smooth and hence it is easier for the optimisation to arrive at the global minima. However, Figure 4.13 depicts a much more rugged loss landscape prone to local minima. Therefore, it is more likely for the optimisation procedure to get stuck in "*bad valleys*" [147]. This analysis shows that higher temperatures facilitate the model to reach the global minima easily whereas low temperatures could lead to the model getting stuck at sub-optimal minima. This indicates that at low temperatures, the model might not fully reach the global minima (complete alignment), and therefore might not close the modality gap. Contrarily, at high temperatures, due to the easier optimisation route, closing the gap becomes easier. This finding throws further light upon why high temperatures has been empirically shown to reduce the modality gap in previous works [118, 185].

**Key Results.**

1. Low *Uniformity* facilitates the *Modality Gap*
2. High *Temperatures* facilitate an easier loss optimisation, thereby promoting *Alignment* – this result adds weight to the result obtained in Section 4.3.

## 4.5 Transferring to the Real World

Having understood the behaviour of the contrastive loss through toy simulations, we now aim to verify that these results hold for real world training runs as well. Specifically, we want to examine if increased temperatures lead to increased alignment, and subsequently reduced modality gap. Further, we investigate if the relationships between *Uniformity* and the *Modality Gap* from Section 4.4 transfer to the real world.

**Fine-tuning CLIP: A Simple Proxy.** For conducting this simulation-to-real-world transfer study, we would ideally like to train CLIP from scratch across different temperatures. However, due to the prohibitive amount of compute required to train CLIP from scratch, this is infeasible for us. We instead propose to fine-tune the pre-trained CLIP model as a reasonable proxy. Despite fine-tuning potentially biasing the embeddings due to different training data distributions, we justify this design choice as this bias affects all our training runs, and hence we can still isolate the effects of *Temperature* and *Uniformity* on the embedding space.

Previous works [38, 56] have suggested that fine-tuning the entire CLIP model (both image and text encoders) can lead to gradient instability, over-fitting and high latency. Our initial fine-tuning experiments reflected these issues[8]. We therefore adopt the strategy used by Couairon et al. [38] to fine-tune CLIP at different temperatures. Specifically, we fine-tune linear adapter layers over CLIP-ViT-B/32's frozen image and text encoders using a contrastive loss with different temperatures on the MS-COCO dataset [119]. We perform the different fine-tuning runs at

[8]These training instability issues are also well known in the community: https://github.com/openai/CLIP/issues/150, https://github.com/openai/CLIP/issues/161

$\tau = \{0.005, 0.01, 0.05, 0.1, 0.15, 0.25, 0.5, 1.0\}$. We use an output embedding dimensionality of 512. We train our models for a maximum of 50 epochs with a batch-size of 512. We use an Adam optimiser [93] with a learning rate decay scheduler starting from 0.001. We also clip gradients at norm 1 to ensure training stability. We run all our experiments on 4 NVIDIA A-100 GPUs.

**Formalising the *Modality Gap*.** Until now, we have used the terms *Alignment* and *Modality Gap* interchangeably. However, to analyse the relationships of the *Modality Gap* with *Uniformity* and *Temperature*, we require a formal definition. We borrow the definition of the modality gap from Liang et al. [118]. Assume we have $N$ paired image embeddings ($\{I_1, I_2, \ldots, I_N\}$) and text embeddings ($\{T_1, T_2, \ldots, T_N\}$) on the unit hypersphere. Then, we define the modality gap as:

$$\begin{aligned} \mu_I &= \frac{\sum_{i=1}^{N} I_i}{N} \\ \mu_T &= \frac{\sum_{i=1}^{N} T_i}{N} \\ \textit{Modality Gap} &= \|\mu_I - \mu_T\|_2 \end{aligned} \tag{4.2}$$

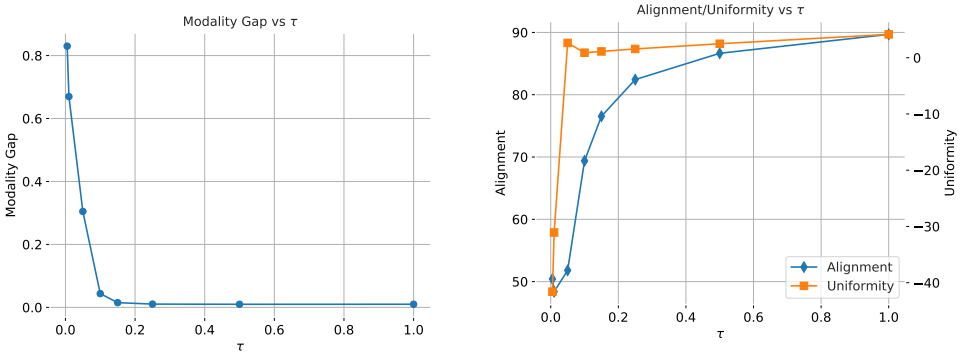
**Formalising *Alignment* and *Uniformity*.** In the toy experiments, we were readily able to alter *Alignment* and *Uniformity* as they were controllable parameters. However, for real world settings, we do not have access to the underlying statistics of the embedding spaces, rather just the image-text embeddings themselves. We therefore require a method to compute metrics reflecting the true underlying *Alignment* and *Uniformity*. Wang et al. [208] defined these quantities for the embedding spaces of uni-modal self-supervised learning methods [27, 29, 69]. Goel et al. [59] extended these definitions to the image-text setting. We reuse these definitions for our analysis:

$$\begin{aligned} \textit{Alignment} &= \frac{\sum_{i=1}^{N} I_i^T T_i}{N} \\ \textit{Uniformity} &= \log\Big(\frac{\sum_{i=1}^{N}\sum_{j=1, j\neq i}^{N} \exp(-I_i^T T_j)}{N}\Big) \end{aligned} \tag{4.3}$$

**High *Temperatures* reduce the *Modality Gap*.** Having formalised our quantitative metrics, we analyse the effect of high temperatures on the modality gap. For this, we encode the test set of MS-COCO [119] (5000 image-text pairs) using our different temperature fine-tuned models. Figure 4.15a shows that as we increase the temperature, the modality gap decreases. Therefore, this result corroborates the findings of our toy experiments.

***Uniformity* and *Alignment* increase with *Temperature*.** In Figure 4.15b, we show the relationship of *Alignment* and *Uniformity* with temperature[9]. As the temperature increases, both the *Alignment* and *Uniformity* increase. As we have uncovered from Sections 4.3 and 4.4, high *Uniformity* is desirable for obtaining the optimal loss. Therefore, the results from Figure 4.15b further bolsters our claim that

[9]This correlation between *Uniformity* and *Temperature* has not been studied in the vision-language literature before. Wang et al. [203] however observed contradictory behaviour *i.e.* a negative correlation between *Uniformity* and *Temperature* when performing a similar analysis for uni-modal image-only self-supervised learning (SSL) methods. This further highlights that despite the contrastive loss being essentially the same, the vast difference of whether negative samples come from data augmentations (SSL) or a different modality (CLIP) is pivotal.




(a) Modality Gap vs Temperature (b) Alignment/Uniformity vs Temperature

Fig. 4.15 Effects of Fine-tuning with different Temperatures

high temperatures facilitate easier optimisation routes, thereby allowing the model to reach the global minima.

**Key Results.**

1. High *Temperatures* facilitate easy loss optimisation, thereby promoting *Alignment*. Hence, at high *Temperatures*, we reduce the *Modality Gap*.

2. *Uniformity* and *Alignment* are desirable for the embedding space. High *Temperatures* make it amenable for the embedding space to achieve these properties.

## 4.6 Summary

In this chapter, we took a deep dive into understanding the modality gap phenomenon. We first reviewed the contrastive loss and disentangled its components in Section 4.1. We then analysed a simple 2-dimensional toy scenario in Section 4.2 to gain an intuitive sense for the behaviour of the loss. We formulated toy experiments for real world simulation of the expected contrastive loss under different settings in Sections 4.3 and 4.4 to capture how different factors affected the loss behaviour. Finally, we confirmed our results from the simulations by transferring to real-world settings in Section 4.5. We summarise results from all our individual sections below for ease of comparison.

| Points on the unit circle | Points on the 3D sphere | Realistic toy sampling setup | Real-world settings |
|---|---|---|---|
| 1. $U \uparrow \implies$ loss $\downarrow$<br>2. $U \overset{\propto}{\sim} A$<br>3. $\tau \overset{\propto}{\sim} \frac{1}{A}$, and the strength of this correlation depends on $U$. | 1. $U \uparrow \implies$ loss $\downarrow$, when $M$=0.<br>2. $\tau \overset{\propto}{\sim} A$<br>3. $M$ and $\tau$ play a large role in promoting *Modality Gap*. | 1. $U \downarrow$ facilitates *Modality Gap*<br>2. $\tau \uparrow \implies$ smooth optimisation, hence $A \uparrow$ | 1. $\tau \uparrow \implies$ reduced *Modality Gap*.<br>2. $U \uparrow$ and $A \uparrow$ are desirable.<br>3. $\tau \uparrow$ facilitates $U \uparrow$ and $A \uparrow$. |

**Table on Notation.**

| Symbol | Meaning | Symbol | Meaning | Symbol | Meaning |
|---|---|---|---|---|---|
| $U$ | *Uniformity* | $A$ | *Alignment* | $\tau$ | *Temperature* |
| $M$ | *Mismatch Ratio* | loss | Expected contrastive loss | $X \uparrow$ | large values of $X$ |
| $X \downarrow$ | small values of $X$ | $X \overset{\propto}{\sim} Y$ | $X$ is correlated to $Y$ | $X \implies Y$ | $X$ implies $Y$ |

**Key Takeaways.** The most clear statement that exists about the modality gap is provided by Liang et al. [118]. They posit that the presence of mismatches at low temperatures causes the preservation of the modality gap. However, our evidence suggests that these results do not hold under all conditions, and our extensive experimental results depict that more work is needed to pinpoint the exact set of factors that underlie the modality gap. We can however provide two key takeaways with confidence:

1. Increasing the temperature is a sure-shot method to reduce the modality gap.
2. For perfect loss optimisation, high alignment and high uniformity are desirable.

# Chapter 5

# Mitigating the modality gap

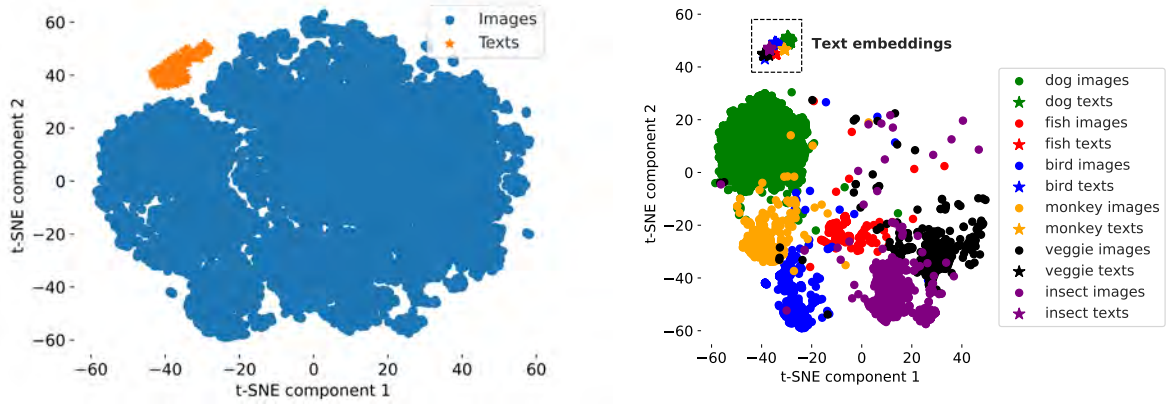
In the previous chapter, we have thrown light upon the mechanisms underpinning the behaviour of CLIP-like models under various conditions. Having understood to some extent what conditions lead to the formation of the modality gap, we now turn to the question of why its existence matters. To answer this question, it is useful to remind ourselves that the original motivation of CLIP and its variants was to solve diverse multi-modal downstream tasks without requiring extensive fine-tuning. The curious characteristics of the emergent embedding space of these models are only a byproduct of the larger overarching goal of achieving strong downstream task transfer in zero-shot and few-shot settings.

In this chapter, we discuss why the modality gap precludes effective visualisation of CLIP-like models' embedding spaces. We then propose a simple method to effectively visualise these embedding spaces by acknowledging the existence of the modality gap. Further, we delineate the implications of the modality gap phenomenon on different downstream tasks under several settings, and propose methods to improve the downstream task performance by navigating around the modality gap. Finally, we motivate an understudied downstream task, vector arithmetic in the embedding space, and benchmark CLIP's performance on this task under various settings by relating it to the modality gap.

## 5.1 A new way to visualise CLIP's embedding space

We motivate this section by taking the perspective of a data-scientist / machine learning practitioner. A meticulous machine learning practitioner is constantly trying to understand and visualise the methods that he/she is working with. This has led to the blossoming of several interpretability and visualisation techniques [200, 1, 120]. t-SNE [198] has emerged to be one such strong technique.

In Section 3.1, we showcased several t-SNE plots of the embedding spaces of CLIP-like models (see Figure 5.1 for a refresher). We observed that the plots showed the image and text embeddings lying in two separated regions of the embedding space due to the modality gap. This is clearly a sub-optimal visualisation since it does not capture the intricate inter-modal distances between the




(a) Entire embedding space

(b) Subspace with specific Imagenet classes

Fig. 5.1 t-SNE visualisation of the entire image-text embedding space on Imagenet (left) and subspace containing class-specific image-text embeddings (right)

image and text pairs. Hence, a machine learning practitioner working with these CLIP-like models can be left unsatisfied with such a visualisation.

Before proposing a solution to this problem, we first consider the applications in which visualising these embedding spaces are most helpful. Typically, we visualise these embedding spaces to understand the underlying local structure (*i.e.* clustering of similar semantic concepts) in the case of downstream zero-shot classification datasets. Consider the case of Imagenet [42]. It has 1000 different classes, and covers a wide span of semantic concepts ranging from birds and animals, to everyday inanimate objects. These classes are interrelated since they are derived from the Wordnet hierarchy [136]. Visualising the CLIP embedding space on Imagenet would hence help us understand how well CLIP is able to capture the relationships between these different semantic classes. Usually, in these zero-shot problems, we are given $N$ classes (having some text labels). In the case of Imagenet, a few examples of class labels are 'tench', 'goldfish' and 'sturgeon'. For each of these $N$ classes, we have access to $M$ image samples. Therefore, in the zero-shot setup, we typically have $N$ text prompts (one for each class) and $N \times M$ images. We encode them using CLIP (or its variants) to obtain $N$ text embeddings and $N \times M$ image embeddings.

Having established the problem setup, we present our solution to the visualisation problem. We start with the pairwise distance matrix $D$ that is typically used to perform t-SNE visualisation. The regular method to construct $D$ is to compute the pairwise distances between each image-image pair, each text-text pair, and each image-text pair. This leads to a square matrix of size $(N+NM) \times (N+NM)$ *i.e.* $D \in \mathbb{R}_{+}^{(N+NM)\times(N+NM)}$ (illustrated in Figure 5.2). As discussed above, passing in $D$ to the t-SNE algorithm leads to sub-optimal visualisations (Figure 5.1).

We design a simple method to manipulate $D$ to contain only inter-modal distances. Since $D$ consists of all pairwise distances, it can be decomposed into 4 sub-matrices $D_{II}$, $D_{TT}$, $D_{IT}$ and $D_{TI}$ (see Figure 5.2). $D_{II}$ denotes the sub-matrix containing only pairwise image-image distances, $D_{TT}$

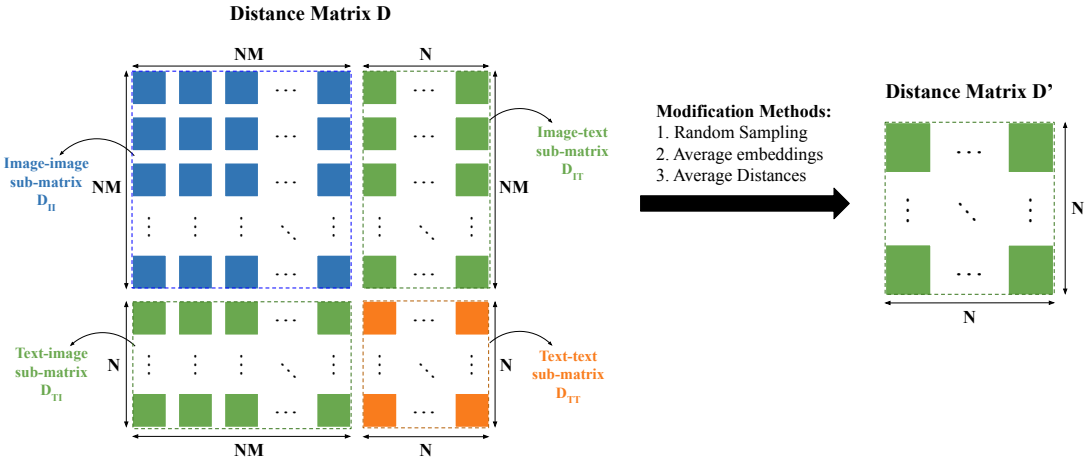



Fig. 5.2 Illustration of original and modified distance matrices

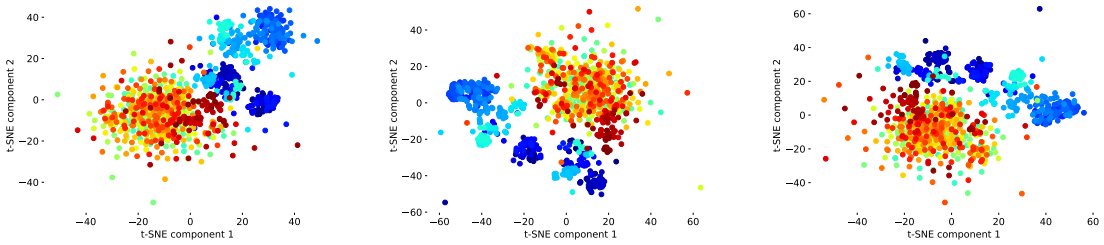
is the sub-matrix containing only pairwise text-text distances, and $D_{IT}$ and $D_{TI}$ are transposes of each other and contain only pairwise image-text distances. We extract only the $D_{IT}$ sub-matrix (or alternatively its transpose, $D_{TI}$) and consider it to be our inter-modal distance matrix. We then manipulate this matrix in 3 different ways to form our modified distance matrix $D'$ which can be directly used as input to t-SNE (or other related algorithms):

1. **Randomly sampling $N$ image embeddings.** We randomly sample one image embedding corresponding to each of the $N$ classes. This way, we sample $N$ out of the total $N \times M$ image embeddings. We then accumulate all the distances from $D_{IT}$ corresponding to the $N$ sampled images, leading to an $N \times N$ matrix. We treat this matrix as $D'$.

2. **Averaging image embeddings.** For each class, we average all $M$ image embeddings belonging to that class, and re-normalise onto the unit hypersphere. We then compute the pairwise distances between the $N$ average image embeddings and $N$ class text embeddings. We then treat the resulting $N \times N$ matrix as $D'$.

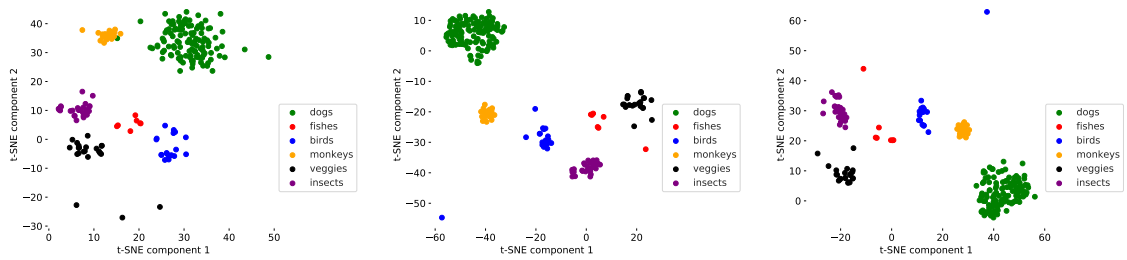
3. **Averaging distances across image embeddings per class.** For each class, we take an average of the pairwise distances of each image embedding belonging to that class with all of the class text embeddings. This way, for each class we compute the average pairwise distance to each of the $N$ class text embeddings. Enumerating over all classes again leads to an $N \times N$ matrix which we treat as $D'$.

In essence, this might seem like an unrefined method since the resulting matrix $D'$ does not follow the norms of a regular distance matrix, namely (i) It is not symmetric, *i.e.* $D'[i,j] \neq D'[j,i], 1 \leq i \leq N, 1 \leq j \leq N$, and (ii) It has non-zero elements on its diagonal, *i.e.* $D'[i,i] \neq 0, 1 \leq i \leq N$. However, this distance matrix exactly encapsulates what the CLIP models were trained for – the paired image-text distances are small *i.e.* the diagonal elements of $D'$ are the smallest elements in their corresponding




(a) Random Sampling (b) Average Embeddings (c) Average Distances

Fig. 5.3 **t-SNE visualisation of entire embedding space using $D'$.** We colour the different points according to the order they appear in the dataset class label IDs. This automatically translates to a strong local semantic clustering behaviour since the label IDs are ordered in such a way that similar classes have adjacent IDs. For example, there are 120 different dog breeds having adjacent label IDs in the range 151-270.




(a) Random Sampling (b) Average Embeddings (c) Average Distances

Fig. 5.4 t-SNE visualisation of specific concepts using $D'$

rows and columns. Thus, when we use $D'$ as input to t-SNE, we retrieve a visualisation that exactly captures CLIP's "*pseudo-cross modal space*" *i.e.* the semantic embedding landscape of where CLIP thinks each class lies relative to other classes. Hence, each point in the above t-SNE plots represents an explicit embedding of a class.

From Figure 5.3, we see that the embedding space of CLIP now accurately captures the semantic structure of the different classes in Imagenet. To further concretise this, in Figure 5.4 we show the local semantic structure of different concepts. We borrow the definition of a concept from the Wordnet hierarchy [136] *i.e.* each concept consists of all the classes belonging to that concept synset from Wordnet. We observe that the points corresponding to the same concept are clustered close together, highlighting the ability of CLIP to capture semantic concept-level similarities.

One point to note here is that the averaging-based methods for obtaining $D'$ result in cleaner visualisations of the semantic concept space as compared to the random sampling method (which can be prone to outliers due to its stochasticity).

We believe the proposed visualisation technique is both simple and useful. It clearly depicts how the inter-modal distances of CLIP allow it to capture relationships between different semantic

concepts, thus enabling it to generalise well to zero-shot downstream tasks. Hence, this method leads to a richer visualisation of CLIP's properties. This simple technique can be trivially extended to any vision-language model since it only involves operating on pairwise distances.

## 5.2 Improving few-shot classification using inter-modal distances

In this next section, we discuss the implications of the modality gap for the downstream task of few-shot image classification. As we have previously seen from Chapter 3, the intra-modal and inter-modal distances and cosine-similarities of the CLIP embeddings have largely different distributions. The inter-modal cosine similarities are sparsely distributed with a small variance and a small mean, whereas the intra-modal cosine similarities are much more spread out with larger means and variances (refer Figure 3.2). Further, we know that CLIP (and its variants) was trained to maximise the inter-modal cosine similarities of paired samples through the contrastive loss (refer Equation 4.1). However, the contrastive loss has no explicit terms controlling the intra-modal cosine similarities. This implies that the intra-modal cosine similarities are not explicitly optimised for, and cannot be considered reliable estimates for the true intra-modal similarities. These intra-modal similarities simply follow by virtue of the transitivity property: For example, consider that CLIP's training dataset has two paired instances, both of a dog. Since both texts in these training samples likely contain the word "dog" and other similar words that usually co-occur with the word "dog", CLIP's text encoder is constrained to encode these two texts closer together in embedding space than with respect to the other "non-dog" texts[1]. Further, since the contrastive loss forces the paired image and text embeddings to be close together in the space, by virtue of transitivity, the two image embeddings and the two text embeddings are implicitly brought close together, as illustrated in Figure 5.5.

We make one further observation: In practice the intra-modal image similarities are weaker than the intra-modal text similarities. This observation is validated by Figure 3.2 depicting that the mean intra-modal text similarity is much larger than the mean intra-modal image similarity. This follows from a simple observation: It is highly unlikely that two images have the exact same set of pixels whereas it is quite likely for two captions to have the same set of text tokens. Therefore, in practice, the smoothness regularisation in the text space is stronger than in the image space.

We thus conclude that the image-only sub-space of CLIP's embedding space is not well calibrated. This means that the image-only embedding space is not reliable for computing intra-modal image-image similarity. We further clarify this using a simple example illustrated in Figure 5.6. Consider two image embeddings that are required to be a distance $r$ away from a particular text embedding. The two image embeddings can satisfy this condition by being very close to each other or very far apart from each other. Figure 5.6 shows that this constraint can be satisfied by any two arbitrary points on a hypersphere of radius $r$ – directly corroborating our conjecture that the image intra-modal

[1]This smoothness phenomenon follows from the regularisation effects of the architectural prior of text encoders. This phenomenon has also been hypothesised by concurrent work [59, 25].

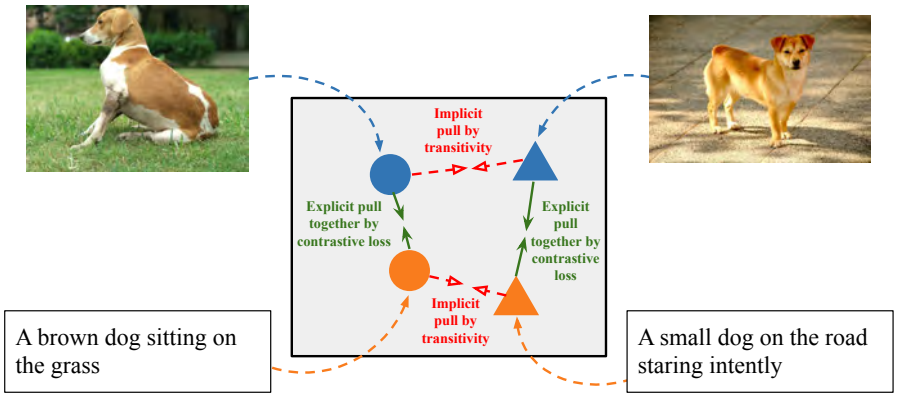



Fig. 5.5 Depiction of the different forces between image-image, text-text and image-text embeddings. The circles denote embeddings for the dog sample on the left and the triangles denote embeddings for the dog sample on the right. The blue embeddings denote image embeddings while the orange embeddings denote text embeddings. The green arrows show strong explicit forces in the optimisation whereas the red arrows depict a weak implicit force.

embedding space is not well-calibrated. To further bolster this claim, we perform a simple analysis on the intra-modal and inter-modal class rankings obtained by CLIP on Imagenet.

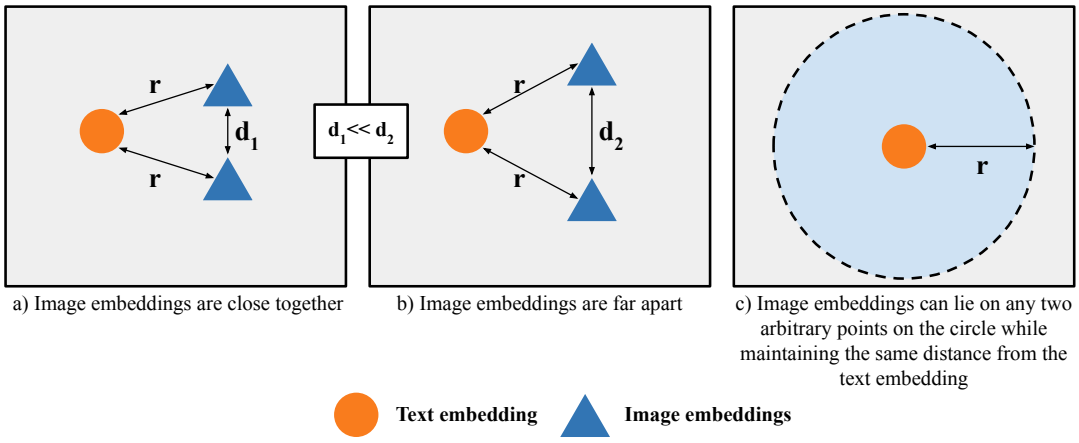



Fig. 5.6 Figure showing the image intra-modal embedding space calibration problem.

### 5.2.1 Motivating Analysis: Classwise rankings on Imagenet

Our goal in this experiment is to study the rankings of different classes using inter-modal similarities and intra-modal similarities. For example, given 4 classes, namely, “Dalmation”, “Greyhound”, “Indian elephant” and “Tabby cat”, we want to inspect the similarity ranking of each class to every other class, using image-image ranking, text-text ranking, and image-text ranking. We consider the

image-text ranking as our gold-standard ranking method since CLIP is trained to optimise image-text similarities.

For this experiment, we encode the Imagenet test set using CLIP. For the images, we directly use CLIP's image encoder whereas for the texts, we use prompt ensembling on the class names, followed by encoding using CLIP's text encoder. We get 50000 image embeddings and 1000 text embeddings, since there are 1000 classes and each class has 50 corresponding images in the test set. As we want to do ranking of classes, we ideally require one prototype image vector per class – for this, we simply take the normalised mean of all image embeddings within a specific class, and consider it to be our class image prototype. This step is not required for the text embeddings since we already have one representative text embedding per class. We now perform ranking in the image-image space, image-text space and text-text space using cosine-similarity as our metric. As an example, the top-5 closest classes to the "Dalmation" class and the "Vespa" class, across the three ranking methods are shown in Table 5.1.

| **Class** | **Image–Image Ranking** | **Text–Text Ranking** | **Image–Text Ranking** |
|---|---|---|---|
| *Dalmation* | Great Dane | Leopard | English Setter |
| | English Setter | Great Dane | Great Dane |
| | Basset Hound | Jaguar | Appenzeller Sennenhund |
| | Bluetick Coonhound | Cheetah | Brittany dog |
| | Wimaraner | Tiger | German Shorthaired Pointer |
| *Vespa* | Moped | Moped | Moped |
| | Tricycle | Tricycle | Tricycle |
| | Tandem bicycle | Tractor | Go-kart |
| | Car wheel | Golf cart | Recreational vechicle |
| | Golf cart | Recreational vehicle | Crash helmet |

Table 5.1 Ranking of top 5 closest classes in image-image space, text-text space and image-text space, of two Imagenet classes: *Dalmation* and *Vespa*. We exclude the true classes from their own ranking lists.

It is evident that the ranking orders across the intra-modal and inter-modal methods is quite different, further hinting at the fact that the intra-modal embedding spaces (image-only and text-only) are not well calibrated. Having conducted a fine-grained class-level ranking analysis, we test the coarse-grained ranking similarities across the three methods by leveraging the Wordnet hierarchy [136]. We use the path similarity metric as defined by Pedersen et al. [155] to compute the similarity between any two classes – it is computed as the inverse of the shortest path distance between the Wordnet synset nodes corresponding to the two classes. We retrieve the correct Wordnet synset nodes for each class from the Imagenet metadata. While performing the ranking, we take each retrieved class, and compute the path similarity of it with the current class we are performing the ranking for. We then bin

these similarity values into discrete ranking bins, and assign a bin value for each retrieved class[2]. This way, for each class, we get a ranking of Wordnet path similarity bins between classes, rather than the actual retrieved classes themselves. There can be multiple classes that have the same path similarity bin – for example, "leopard", "jaguar", "cheetah" and "tiger" all have a path similarity bin of 5 when using "Dalmation" as a query class.

Therefore, this method can help us get a more coarse-grained ranking order of classes. This in turn allows us to compare across the three ranking methods at a coarser scale, rather than at a fine-grained per-class level. We illustrate the ranking orders of "Dalmation" and "Vespa" using this coarse-grained method in Table 5.2.

| Class | Image–Image Ranking | Text–Text Ranking | Image–Text Ranking |
|---|---|---|---|
| *Dalmation* | 2 | 5 | 4 |
| | 4 | 2 | 2 |
| | 3 | 5 | 3 |
| | 3 | 5 | 4 |
| | 3 | 5 | 4 |
| *Vespa* | 5 | 5 | 5 |
| | 1 | 1 | 1 |
| | 2 | 2 | 3 |
| | 8 | 3 | 2 |
| | 3 | 2 | 8 |

Table 5.2 Wordnet path similarity bins of top 5 closest classes in image-image space, text-text space and image-text space, of two Imagenet classes: *Dalmation* and *Vespa*. A smaller path similarity bin indicates that the query class and retrieved class are closer (in terms of shortest path distance) in the Wordnet tree. Hence, the smaller the bin, the more semantically similar the two classes are according to the Wordnet taxonomy. We exclude the true classes from their own ranking lists.

We now compare the ranking orders across all classes, using the fine-grained class-wise ranking as well as the coarse-grained Wordnet path similarity ranking. For the fine-grained ranking, we consider the entire ranked list of 999 classes for each query class[3], whereas for the coarse-grained ranking, we consider only the path similarities of the top-5 closest classes. For enabling this comparison, we consider the image-text rankings to be the gold-standard ranking, and compare the image-image and text-text rankings individually with the image-text rankings. We then compute the Kendall's rank correlation coefficient ($\tau$) between the rankings. In Figure 5.7, we show the distribution of correlation coefficient values over all 1000 classes.

[2]For performing the binning of the path similarity values, we first sort all possible unique similarity values obtained across all pairwise combinations of classes, in descending order. To find the bin value for a particular retrieved class, we simply index into the sorted unique value array – this implies that the more similar two classes are (high path similarity), the smaller their bin value would be.

[3]We remove each class from its own ranking list since we always get a cosine similarity of 1 when considering a class with itself. Therefore, we are left with 999 classes in the ranking list for each class.

We see that the the Kendall's $\tau$ values for both fine-grained and coarse-grained rankings using both image-image and text-text rankings have a very high variance – especially in the case of coarse-grained rankings, the Kendall's $\tau$ ranges from -1 to 1. This indicates that the intra-modal rankings are not consistent with the gold-standard inter-modal rankings, which directly corroborates our hypothesis that the intra-modal embedding spaces are not well-calibrated. Therefore, the results of this simple experiment further cement the claim that the image-image embedding landscape cannot be directly used for measuring image-image similarities reliably.

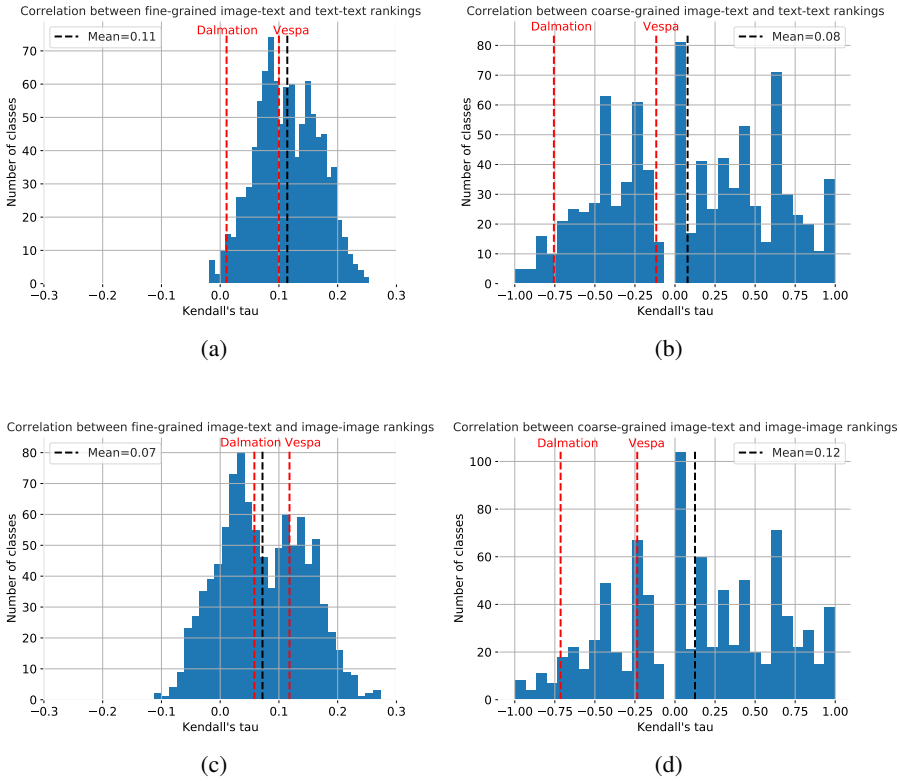



Fig. 5.7 Distribution of Kendall's rank correlations between the gold-standard image-text rankings and the intra-modal rankings. We annotate the correlations of the *Dalmation* and *Vespa* classes with red dotted lines to indicate the degree to which the qualitative results shown previously are representative. For ease of comparison, the x-axis limits for the fine-grained plots are (-0.3, 0.3) whereas for the coarse-grained plots are (-1.0, 1.0)

### 5.2.2 Using the image-text embedding similarities for image-image comparison

Having established that using the intra-image cosine similarities as a similarity metric across image samples is not reliable, we now discuss why this finding is important for the task of few-shot image classification.

Recall the TIP-Adapter [232] method (discussed in Section 2.4.3, see Figure 5.8 for a refresher) for improving few-shot classification. This method uses an image-image similarity comparison in

CLIP's image-space to compute the similarities between query images and images from the cache[4]. However, our aforementioned analysis uncovers a systematic issue with this approach – since the image-only embedding space of CLIP is not well calibrated, this method of computing similarities between the query and cache images might be sub-optimal. We aim to fix this issue by relying on inter-modal image-text similarities rather than the unreliable intra-modal image-image similarities. We call our new method *TIP-X*[5].

**Method Setup.** We follow the exact architectural setup of TIP-Adapter. For a $K$-shot classification task, we are given access to a supervised dataset consisting of $C$ classes, with $K$ training images per class. Therefore, the training dataset consists of $CK$ labelled images. We are also given access to the pre-trained CLIP model *i.e.* both CLIP's image and text encoders. We first convert each of the $C$ classes into their corresponding embedding vectors by using prompt ensembling [232, 165]. This gives rise to the standard text classifier weights used by the CLIP zero-shot classifier. We denote the text classifier as $W \in \mathbb{R}^{C \times d}$. We next convert the given few-shot dataset to a set of cache image features and cache labels akin to TIP-Adapter [232]. We encode each of the images in the few-shot dataset using CLIP's image encoder, and construct a $CK \times d$ matrix, which we call the cache image features $F$. $F$ can be thought of as the set of keys with which to compute affinities for each of the test query images. Then, we compute one-hot encodings of the class labels for each of the images in the few-shot dataset. We call these the cache labels $L \in \mathbb{R}^{CK \times C}$ . Intuitively, TIP-Adapter can be thought of as computing attention weights for each of the few-shot dataset images by weighing each image's class label with how similar that image is to the given query test image. Our classification task is to correctly classify $t$ test images. The test image features encoded through CLIP's image encoder are denoted $f \in \mathbb{R}^{t \times d}$. Recall the computation of the classification logits for Zero-shot CLIP ($ZSL$) and TIP-Adapter ($TL$):

$$\boxed{ZSL = fW^T} \tag{5.1}$$

$$\boxed{TL = fW^T + \alpha\phi(fF^T)L} \tag{5.2}$$

where $\phi(x) = \exp(-\beta(1-x))$. Here, $\alpha$ and $\beta$ are hyperparameters. $\alpha$ controls the balance between the zero-shot component and the few-shot component of the logits computation whereas $\beta$ modulates the sharpness of the exponential activation of the cache affinities.

**Constructing Signature Distributions.** We now present our method for fixing TIP-Adapter's intra-modal image-image comparison pathology. We first compute the inter-modal similarities between the cache image embeddings and the text classifier weights:

$$S = \texttt{softmax}(FW^T) \tag{5.3}$$

[4]Here cache refers to the few-shot dataset. We adopt the same terminology as the original paper [237].

[5]Our method name is inspired by the Tipp-Ex Rapid Correction Fluid – we hope to *rapidly correct* the effects of using unreliable intra-modal distances.

$S \in \mathbb{R}^{CK \times C}$ contains probability distributions of how similar each cache feature is with respect to all the class text vectors. Since we use these distributions as features for classification, we call $S$ as the set of cache signatures (terminology adopted from [12]). We also construct a set of query signatures $s \in \mathbb{R}^{t \times C}$ similar to the cache signature construction:

$$s = \texttt{softmax}(fW^T) \tag{5.4}$$

Intuitively, these query and cache signatures are analogous to the query and cache image features $f$ and $F$. However, the difference stems from the fact that each signature feature is now a probability distribution over all class text vectors, and not a simple high-dimensional feature vector. This simple construction of signature distributions helps us compute similarities between the cache image features and queries by leveraging their inter-modal image-text similarities.

**Computing Affinities.** Equipped with the cache and query signatures, we need to compute similarities between them to formulate affinity weights for each of the cache labels. Since the signatures represent probability distributions, a straightforward way to compute their similarities is by measuring the KL-divergence between them. We therefore construct the affinity matrix $M \in \mathbb{R}^{t \times CK}$ between our queries and cache image features as follows:

$$\begin{aligned} M_{i,j} &= \texttt{KL}(s_i || S_j) \\ 1 \leq i \leq t, &1 \leq j \leq CK \end{aligned} \tag{5.5}$$

where $s_i$ represents the $i^{th}$ query signature and $S_j$ represents the $j^{th}$ cache signature. Since we are working with discrete probability distributions, we compute the KL-divergence as $\texttt{KL}(P||Q) = \sum_i P_i \log \frac{P_i}{Q_i}$.

There are several other techniques for computing the similarity between the signature distributions including optimal transport [201, 157], Jensen-Shannon divergence [131], Bhattacharya coefficient [43] *etc.* – we leave this exploration for future work.

The construction of the affinity matrix $M$ can be seen as an analogous step to the affinity computation through image-image cosine similarity between the cache keys and queries in the TIP-Adapter framework. However, our affinity construction prevents the problem of relying on the erratic intra-modal image-image similarities.

Finally, before using our affinity matrix $M$ as weights for the cache values, we pass them through an activation function $\psi$ to ensure appropriate scaling. Further, since our affinity matrix $M$ consists of KL-divergence values, the most similar samples will get small weights since their KL-divergence will be low (close to 0). To mitigate this, we simply negate the values in the affinity matrix $M$. The predicted logits of the query images using our TIP-X method are then computed as:

$$\boxed{TXL = fW^T + \alpha\phi(fF^T)L + \gamma\psi(-M)L} \tag{5.6}$$

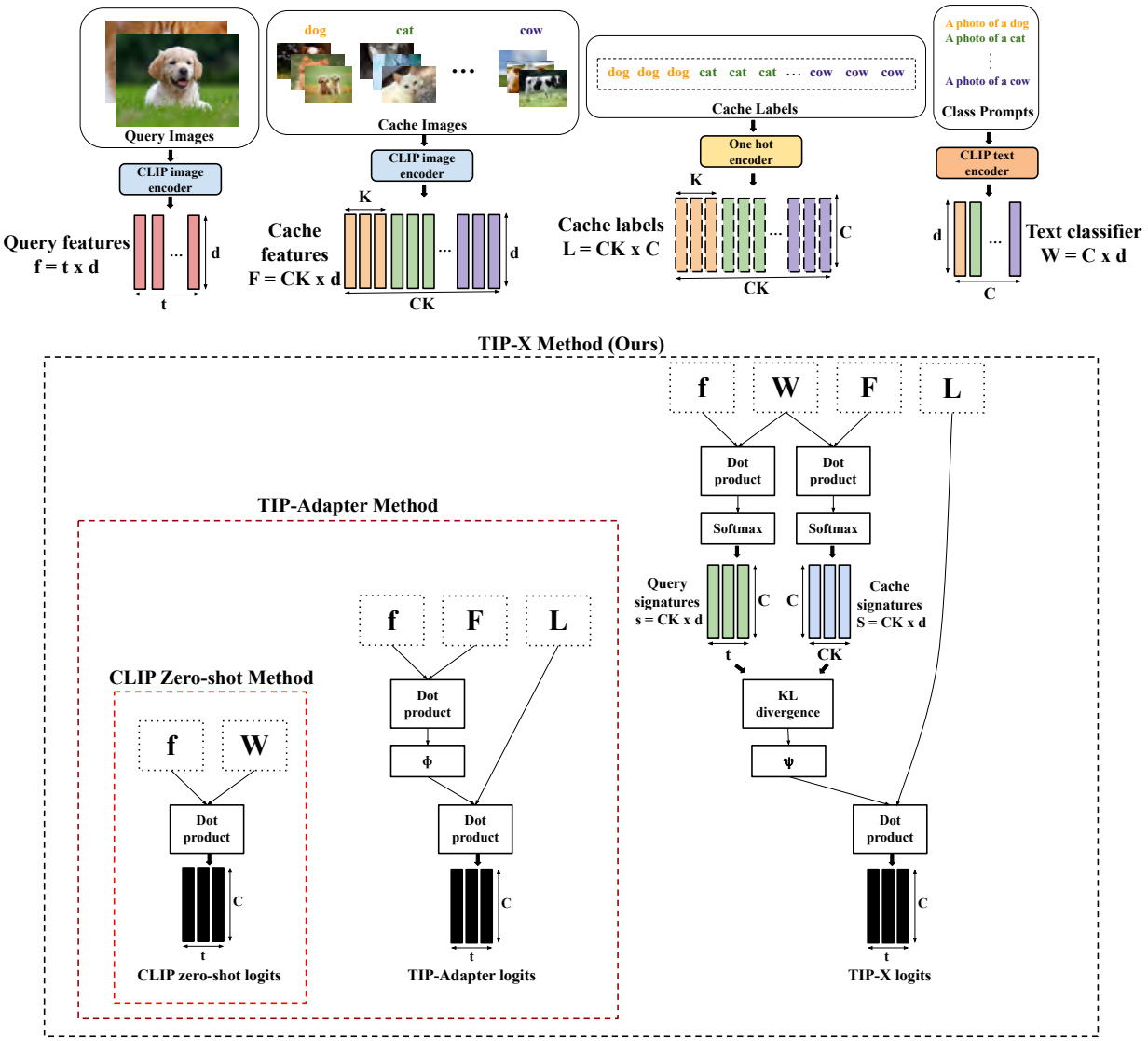



Fig. 5.8 Depiction of the TIP-X method. We show the computation of CLIP's zero-shot logits, TIP-Adapter's cache logits, and our TIP-X logits. We use the cache model (cache image features and cache values) to compute affinities of each query image with every cache image. We do this image-image comparison by using the image-text similarities. For this, we compute the image-text similarity signatures for both query images and cache images, and then calculate the KL-divergence between these signatures. These KL-divergences are then used as weights for our cache values.

where the $\psi$ function heuristically scales all the values of $M$ to be of the same scale as the TIP-Adapter affinities. The entire TIP-X method is depicted in Figure 5.8.

**Fine-tuning TIP-X.** The original version of TIP-Adapter performs extremely well given a few number of shots. However, as the number of shots increase, there is still a modest gap in performance with prior few-shot classification methods [56, 239]. To close this gap, the authors propose a variant of TIP-Adapter called as TIP-Adapter-F that uses fine-tuning on the few-shot dataset. They parameterise

a one-layer MLP $g$ with the weights initialised to the cache keys $F$. They then train $g$ in the exact same setup as TIP-Adapter, with a cross-entropy loss. The authors justify this method as boosting the estimation of affinities between the cache keys and the queries. We also propose a variant of our method, called TIP-X-F, which analogously trains MLPs $g$ and $h$ to compute richer affinities between the cache and query signature distributions. For training $g$, we follow the exact same procedure as TIP-Adapter-F. For training $h$, we parameterise it by initialising its weights as the cache signatures $S$. We consider the output of $h$ to be the pre-affinities between the cache and query signatures. We use two activation functions prior to the final logits computation: (i) a Hard-Swish activation function [76] defined as $\texttt{h-swish}(x) = x\frac{ReLU6(x+3)}{6}$, and (ii) a weighted exponential $w(x) = \exp(\beta x)$. Our final logits computation for the fine-tuned TIP-X-F is:

$$\boxed{TXFL = fW^T + \alpha\phi(g(f))L + \gamma w(\texttt{h-swish}(-h(s)))L} \tag{5.7}$$

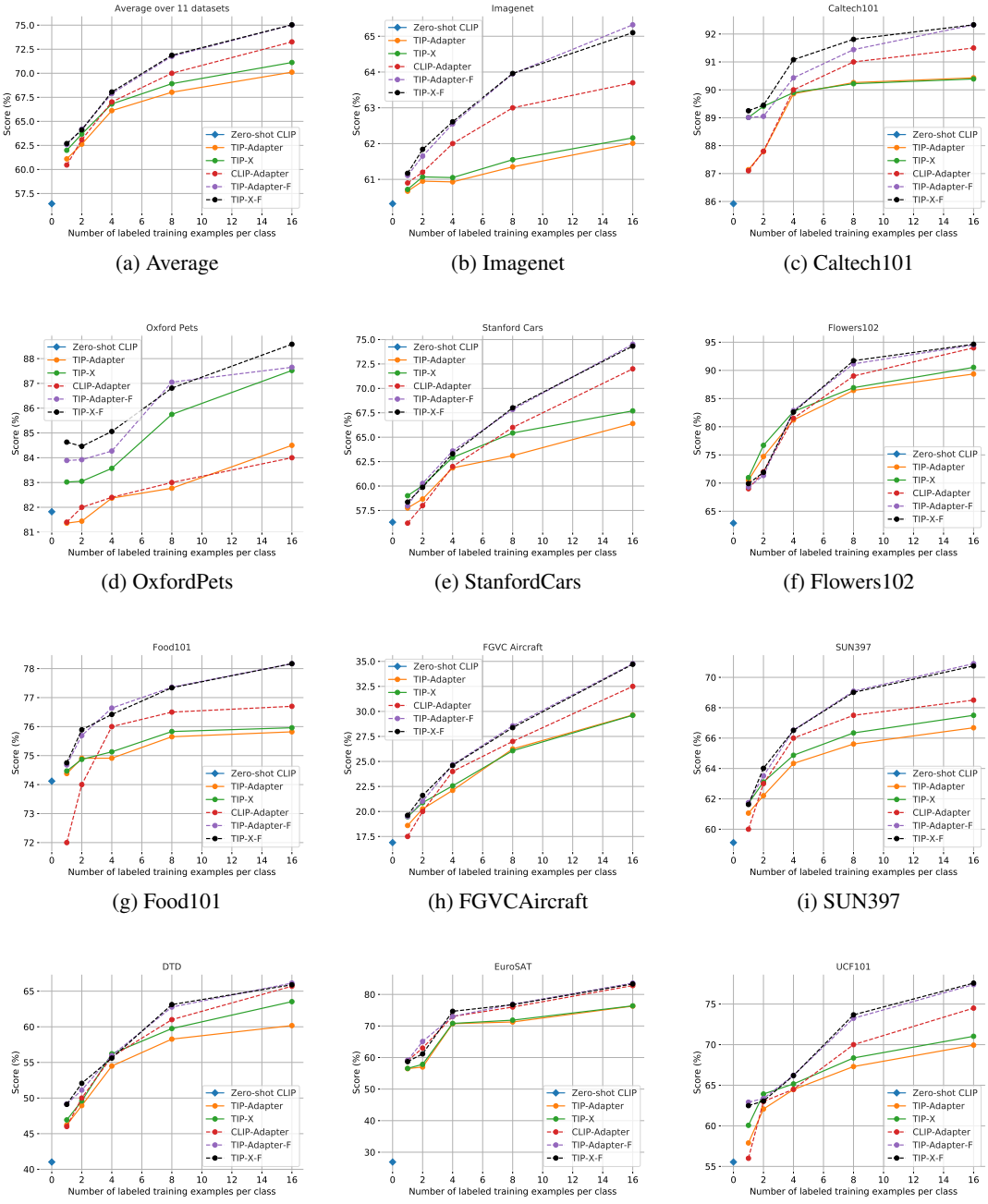
**Experiments and Results.** Following recent works [239, 232, 56, 235], we conduct few-shot classification on 11 benchmark datasets: Imagenet [42], StanfordCars [96], UCF101 [187], Caltech101 [54], Flowers102 [149], OxfordPets [154], Food101 [18], SUN397 [213], DTD [33], EuroSAT [71], and FGVCAircraft [128]. We compare the results of TIP-X and its fine-tuned variant TIP-X-F with zero-shot CLIP [165], CLIP-Adapter [56], TIP-Adapter and TIP-Adapter-F [232]. Since it has been well established that TIP-Adapter [232] outperforms CoOP [239] across all settings, we exclude a comparison with CoOP from our results.

We train each model with five different shot settings of the $K$-shot classification task: 1, 2, 4, 8 and 16. We test classification accuracy on the held-out test sets, following the exact splits used by previous works. We use the pre-trained CLIP ResNet-50 [70] image encoder for all our experiments. For the text encoder, we use CLIP's pre-trained transformer [199]. We use prompt ensembling for converting the class labels into text prompts for all datasets[6]. For our TIP-X method, we fix the temperature of the softmax distribution for computing the signature distributions to be 0.5.

For the fine-tuned variant, we train TIP-X-F on the $K$-shot dataset for 40 epochs with a batch size of 256. We use an Adam optimiser [93] with weight decay [124] using a learning rate of 0.001 and a cosine learning rate scheduler [123]. We tune all our hyperparameters ($\alpha$, $\beta$ and $\gamma$) on the validation set and then transfer them to our test set, similar to the approach taken by TIP-Adapter. We conduct all our experiments on a single NVIDIA A-100 GPU.

We present our main results in Figure 5.9. It is immediately evident that our untrained TIP-X method outperforms the Zero-Shot CLIP and TIP-Adapter baselines by a large margin. We improve the few-shot classification performance over TIP-Adapter across all datasets by an absolute gain of 0.91%. Further, we observe that our untrained TIP-X method even approaches the performance of CLIP-Adapter and TIP-Adapter-F (both of which have been explicitly trained for the few-shot classification task) under certain settings. This further demonstrates the efficacy of our TIP-X method.

[6]We use the same set of 7 prompt templates for prompt ensembling as TIP-Adapter. The 7 prompt templates are: "itap of a <*class*>.", "a origami <*class*>.", "a bad photo of the <*class*>.", "a photo of the large <*class*>.", "a <*class*> in a video game.", "art of the <*class*>.", and "a photo of the small <*class*>."




(a) Average (b) Imagenet (c) Caltech101

(d) OxfordPets (e) StanfordCars (f) Flowers102

(g) Food101 (h) FGVCAircraft (i) SUN397

(j) DTD (k) EuroSAT (l) UCF101

Fig. 5.9 Main few-shot classification results of all methods across 11 datasets. The solid lines represent methods that don't require training whereas the dashed lines represent methods that use fine-tuning.

Therefore, the plots from Figure 5.9 illustrate the benefits of using the inter-modal similarities to compute image-image similarities for matching the query images to cache images.

Despite performing better than the two baselines, TIP-X still lags behind CLIP-Adapter and TIP-Adapter-F across most datasets. This is due to the two methods in question being explicitly fine-tuned for the few-shot task. Our fine-tuned version, TIP-X-F is able to close this gap to these two baselines. We outperform CLIP-Adapter by 1.58% on average and TIP-Adapter-F by 0.07%. We note that the performance gain of TIP-X-F over TIP-Adapter-F is not as substantial as for the corresponding untrained variants – we hypothesise that this may be due to the task-specific fine-tuning bridging the gap between the intra-modal and inter-modal similarity comparison methods, and thereby potentially masking the failure of the intra-modal image-image comparison.

In sum, our TIP-X method preserves the efficiency of TIP-Adapter's training-free protocol while boosting few-shot performance. Our fine-tuned variant, TIP-X-F further improves this few-shot performance, and achieves state-of-the-art results in the few-shot classification task across 11 datasets. Therefore, we have shown a simple yet effective way to leverage the reliable inter-modal image-text similarities to perform an image-image comparison, leading to superior few-shot classification performance when compared to previous state-of-the-art approaches.

## 5.3 Reducing the Modality Gap by Fine-tuning

In Chapter 4, we established that fine-tuning CLIP at higher temperatures is an effective solution for reducing the modality gap in CLIP's embedding space. This strategy has also been hypothesised in prior work [118, 185]. However, none of these works concretely analyse the implications of reducing the modality gap on downstream tasks. In this section, we aim to bridge this gap in the literature by conducting an in-depth empirical analysis of the relationships between modality gap and downstream task performance. Further, we introduce a new task that has not been studied in the context of CLIP's modality gap phenomenon, vector arithmetic in the embedding space. We then formally relate vector arithmetic task performance, downstream task performance and the modality gap.

### 5.3.1 Vector Arithmetic in CLIP's embedding space

In recent years, there have been an abundance of works that study the geometry of word embedding models [134, 135, 55, 108, 52]. This has led to the observation that the embedding spaces of such models enable vector arithmetic, giving rise to tasks that test the ability of these models to perform analogies in their embedding spaces. The canonical example of this phenomenon was provided by Mikolov et al. [135] when they showed that the word embedding of "Queen" was very close in the embedding space to the embedding produced from the arithmetic operation "King" - "Man" + "Woman".

Despite the existence of such analogy methods for word embedding models, only a handful of works have studied such effects in the vision-language landscape. Jia et al. [83] demonstrated that simple arithmetic operations like addition and subtraction could be performed across image and text embeddings, leading to interpretable retrieval results. Couairon et al. [39] further studied

this phenomenon and formalised it by creating a dataset (called SIMAT [38]) to test it. The dataset comprises of approximately 6000 images, each annotated with a caption and a subject-relationship-object triplet (see Figure 5.10a for an illustration). To test vector arithmetic performance, SIMAT contains approximately 18000 transformation queries. Each transformation query contains an image along with its caption and annotation triplet, and a target transformation that modifies either the subject, relationship or object in the triplet. Therefore, for each query, we have access to the original triplet and the target triplet that should be exactly aligned with the transformed image after the vector arithmetic operation. Figure 5.10b depicts a few transformation queries from the SIMAT dataset.

For evaluating models on their ability to perform such cross-modal vector arithmetic operations, the paper proposes a simple metric – for each transformation query, we compute image-text matching scores using OSCAR [116][7] for the target caption and the retrieved image upon performing the required arithmetic operation. If the image-text matching score is greater than 0.5, the retrieved image is considered a successful match. More concretely, assume we have a query image $x_q$ with an annotation triplet $< s_q, r_q, o_q >$[8] and a caption $c_q$, and a transformation query that modifies $s_q \rightarrow s_t$. The target annotation triplet is thus $< s_t, r_q, o_q >$ and target caption is $c_t$. If we want to evaluate a model $M$ with image encoder $M_I$ and text encoder $M_T$, we retrieve the nearest neighbour image embedding $x_t$ from the SIMAT dataset corresponding to the following retrieved image embedding:

$$x = M_I(x_q) + \lambda\,(M_T(s_t) - M_T(s_q))$$
$$x_t = \arg\max_{x_i}(x^T x_i)$$

where $\lambda$ is a scaling parameter whose value can be tuned and $i$ iterates over all the images in the dataset. We then match $x_t$ with the target caption $c_t$ to get a matching score, which is thresholded at 0.5 for filtering matches. The final SIMAT score $S$ is computed as the weighted sum of all matches over the entire dataset:

$$S = \sum_{x_{qi}, c_{ti}, \mu_i} \mu_i \mathbb{I}_{P_O(x_{ti}, c_{ti}) \geq 0.5}$$

where $i$ indexes over all transformation queries in the dataset, $\mu_i$ is a weighting coefficient for each query, $\mathbb{I}$ is an indicator variable, and $P_O(x, y)$ denotes the OSCAR image-text matching score of image $x$ and text $y$. For more details about the SIMAT dataset and evaluation strategy of the vector arithmetic performance, refer to the original paper [39].

**A simple way to reduce the modality gap.** To begin our exploratory analysis on how the modality gap affects vector arithmetic performance, we specify our hypothesis about their relationship:

[7] OSCAR is a large-scale pre-trained vision-language model that can be used for computing similarity scores for a given image-text pair. The usage of OSCAR is a design choice made by the authors in [39]; it can be replaced by any suitable vision-language model capable of computing image-text similarity scores

[8] The annotation triplet consists of subject $s_q$, relationship $r_q$ and object $o_q$. For examples of such triplets, refer Figure 5.10

**Caption**: A bear laying on the grass
**Annotation triplet**: <bear, laying on, grass>

**Caption**: A man running in a field
**Annotation triplet**: <man, running in, field>

(a) Annotated image samples

**Query Caption**: A woman sitting on the beach
**Query Annotation triplet**: <woman, sitting on, beach>

**Target Transformation**: woman → man

**Target Caption**: A man sitting on the beach
**Target Annotation triplet**: <man, sitting on, beach>

**Query Caption**: A zebra drinking water
**Query Annotation triplet**: <zebra, drinking, water>

**Target Transformation**: zebra → giraffe

**Target Caption**: A giraffe drinking water
**Target Annotation triplet**: <giraffe, drinking, water>

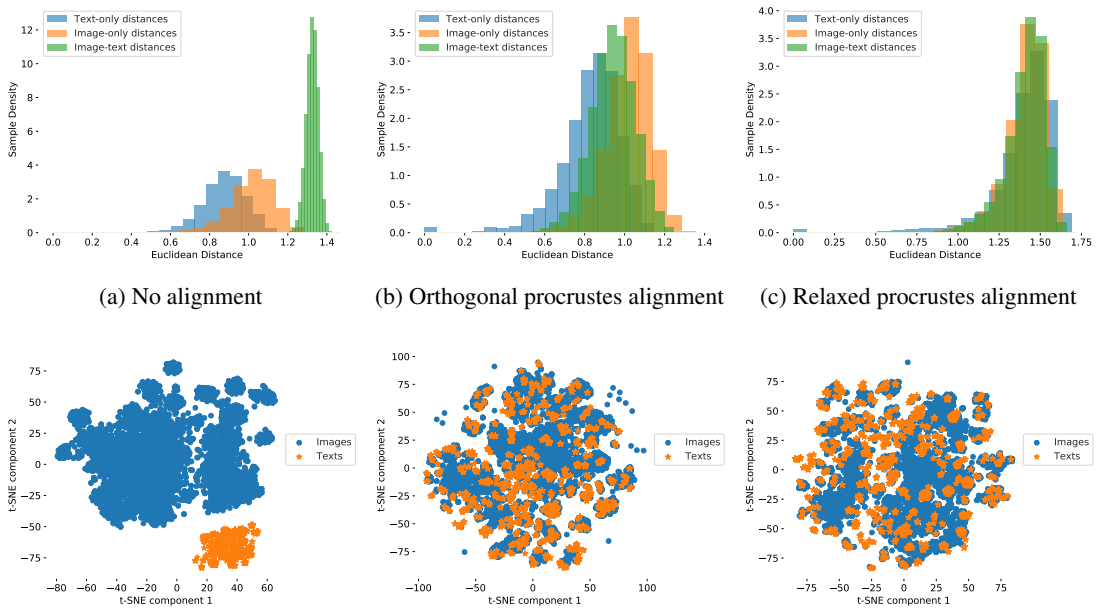
(b) Transformation query examples

Fig. 5.10 Examples of annotated images and transformation queries from the SIMAT dataset

*Since the intra-modal distances and inter-modal distances are not calibrated in the pre-trained CLIP embedding space, we hypothesise that reducing the modality gap*[9] *will help bring the intra-modal and inter-modal distances to the same scale, thereby calibrating them. As vector arithmetic performance in the multi-modal embedding space relies on calibrated distances between the modalities, we further hypothesise that reducing the modality gap should improve the vector arithmetic performance.*

We now conduct a preliminary hypothesis test by formulating a simple way to reduce the modality gap. To quantitatively measure the modality gap, we use our definition from Section 4.5 for all our subsequent experiments. We assume $N$ paired text and image embeddings. The text embeddings are denoted by $T = \{T_1, T_2, \ldots, T_N\}$ and image embeddings are denoted by $I = \{I_1, I_2, \ldots, I_N\}$. Our aim is to reduce the modality gap by aligning $T$ and $I$ close together in embedding space. We employ the Procrustes alignment method [98] for aligning $T$ and $I$. Since the Procrustes alignment solution is a simple technique to align two points clouds in high-dimensional embedding space, we justify its use as a simple proxy for performing the alignment between $T$ and $I$. This has also been used in prior work for aligning embeddings in the NLP domain [36, 7, 64].

[9] Our hypothesis is agnostic to the method used for reducing the modality gap. It only states that a reduction in modality gap is required and does not require a specific method of doing so.




(a) No alignment (b) Orthogonal procrustes alignment (c) Relaxed procrustes alignment

(d) No alignment (e) Orthogonal procrustes alignment (f) Relaxed procrustes alignment

Fig. 5.11 Distance distributions (top row) and t-SNE visualisations (bottom row) before and after alignment of CLIP's image-text embeddings

We use two techniques to perform the alignment: Orthogonal Procrustes [172] and the Relaxed Procrustes method [61][10]. For each method, we consider $I$ to be the fixed embedding set, and transform $T$ to be aligned with $I$. From Table 5.3, we see that both these methods reduce the modality gap by a large margin over the original CLIP. Figure 5.11 depicts the inter-modal and intra-modal distance distributions and t-SNE visualisations after performing the alignment – this validates our hypothesis that by reducing the modality gap, we effectively calibrate the intra-modal and inter-modal distances.

Having demonstrated that the simple alignment method reduces the modality gap, we aim to study the relationship of reducing the gap and vector arithmetic performance. In Table 5.3, we report the SIMAT-scores obtained using the alignment methods compared with the original CLIP model.

[10]Orthogonal Procrustes aligns the two embedding clouds only using a rotation matrix since it is a norm-preserving transformation. However, Relaxed Procrustes can align the two clouds using any linear transformation (rotation, scaling/dilation and translations)

| Alignment Method | Modality Gap | $S(\lambda = 1)$ | $S(\lambda = 2)$ | $S(\lambda = 3)$ |
|---|---|---|---|---|
| None (Original CLIP) (from [39]) | – | 15.90 | 31.20 | 35.40 |
| None (Original CLIP) (*R*) | 0.932 | 14.65 | 28.22 | 34.70 |
| Orthogonal Procrustes | **0.127** | **26.69** | **39.00** | **35.76** |
| Relaxed Procrustes | **0.053** | **26.84** | **39.46** | **35.73** |

Table 5.3 Modality Gap (using definition from Section 4.5) and SIMAT scores using different alignment methods. *R* denotes our reimplementation. Higher is better for the SIMAT scores.

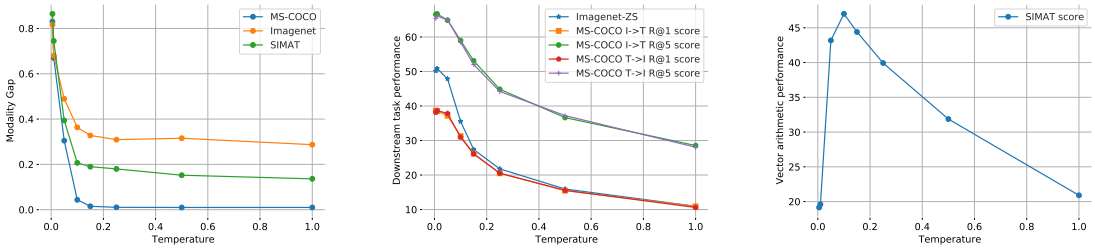
We observe that both alignment methods significantly improve the vector arithmetic performance across different scaling factors. This further supports our hypothesis that by reducing the modality gap, we effectively calibrate distances across modalities in turn boosting vector arithmetic performance.

### 5.3.2 Relating the modality gap with vector arithmetic and downstream tasks

Following our exploratory analysis from the previous section, we now aim to comprehensively study the relationships between the modality gap, vector arithmetic performance and downstream task performance in the context of CLIP's embedding space. We first delineate the two works that have attempted to do such an analysis and identify their limitations. Liang et al. [118] exemplify that modifying the modality gap has some implications on downstream tasks like zero-shot classification and algorithmic fairness. However, they neither quantify the directionality nor the strength of this modification. Couairon et al. [39] conduct experiments by fine-tuning CLIP at various temperatures on the MS-COCO dataset [119], and establish relationships between the fine-tuning temperature $\tau$ and the vector arithmetic performance. However, they do not suggest any concrete reasons as to why $\tau$ modulates the vector arithmetic performance. We extend the results of both these works by viewing the fine-tuning experiments through the lens of the modality gap phenomenon, thereby quantitatively establishing relationships between the three.

**Experimental Setup.** To achieve our goal of establishing relationships, we require a reliable method to modulate the modality gap in a predictable way. We have shown in Section 4.5 that increasing the temperature is an effective way of reducing the modality gap. We therefore use the same fine-tuning setup as Section 4.5 for our experiments.

**Assessing the modality gap.** In Section 4.5, we showcased that high temperatures lead to reducing the modality gap. In Figure 5.12a, we extend this result by measuring the modality gap obtained after fine-tuning at different temperatures across the embedding spaces of three datasets: an in-distribution MS-COCO test set, and two out-of-distribution SIMAT and Imagenet test sets. We term the MS-COCO as in-distribution since the fine-tuning has been performed on the MS-COCO training set. Similar to the results of Section 4.5, we observe that the modality gap decreases steeply from $\tau = 0.005$ to $\tau = 0.1$ across all three datasets. From $\tau = 0.1$ onwards, the decrease in the modality gap is miniscule. Further, we observe that the absolute values of the modality gap are larger for out-of-distribution datasets (Imagenet and SIMAT) than for the in-distribution MS-COCO dataset.




(a) Modality gap (b) Downstream tasks (c) Vector arithmetic

Fig. 5.12 The implications of fine-tuning CLIP at various temperatures on the modality gap, downstream task performance and vector arithmetic performance. Modality gap and downstream task performance monotonically decrease with temperature, while vector arithmetic performance is optimal at moderate temperatures.

**Assessing downstream task performance.** For evaluating the effects of modality gap change on downstream task performance, we study two different tasks – Imagenet zero-shot classification and MS-COCO cross-modal retrieval. We investigate these two tasks since they allow us to compare the effects of in-distribution vs out-of-distribution data to the fine-tuning task. For Imagenet zero-shot classification, we compute the zero-shot accuracy of CLIP using the fine-tuned image and text embeddings. For MS-COCO, we compute the image-text and text-image recall@1 and recall@5 metrics. From Figure 5.12b, it is evident that the data distribution of the downstream task does not affect the relationship of the downstream task performance vs fine-tuning temperature. Further, we observe that the performance of all tasks is optimal at low temperatures, and degrades smoothly as we increase the temperature.

**Assessing vector arithmetic performance.** We use the SIMAT test dataset to evaluate vector arithmetic performance. For all experiments, we use a scaling factor $\lambda = 1$. In Figure 5.12c, we visualise the effects of the fine-tuning temperature on the vector arithmetic performance. We observe that at very low ($\tau = 0.005$) or very high ($\tau = 1.0$) temperatures, our SIMAT score drops drastically. Further, there is only a small band of temperatures ($\tau = 0.05 - 0.2$) at which we improve vector arithmetic performance.

**Bringing it all together.** To study the inter-relation of modality gap, vector arithmetic and downstream performance, we visualise their trends across different fine-tuning temperatures in Figure 5.13. First we compare the scores obtained by the original CLIP with the fine-tuned versions. Recall that the original pre-trained CLIP model was trained with a learnable temperature that converges to a stable value of 0.01. Thus, to isolate the effect of fine-tuning, we compare the scores attained at the same fine-tuning temperature $\tau = 0.01$. We observe that the Imagenet performance takes a drastic hit whereas the MS-COCO retrieval performance is improved. This is due to the fine-tuning being done on MS-COCO and therefore the retrieval task is an in-domain task for the fine-tuned model whereas the Imagenet classification is out-of-domain. We expect to see the inverse effects if

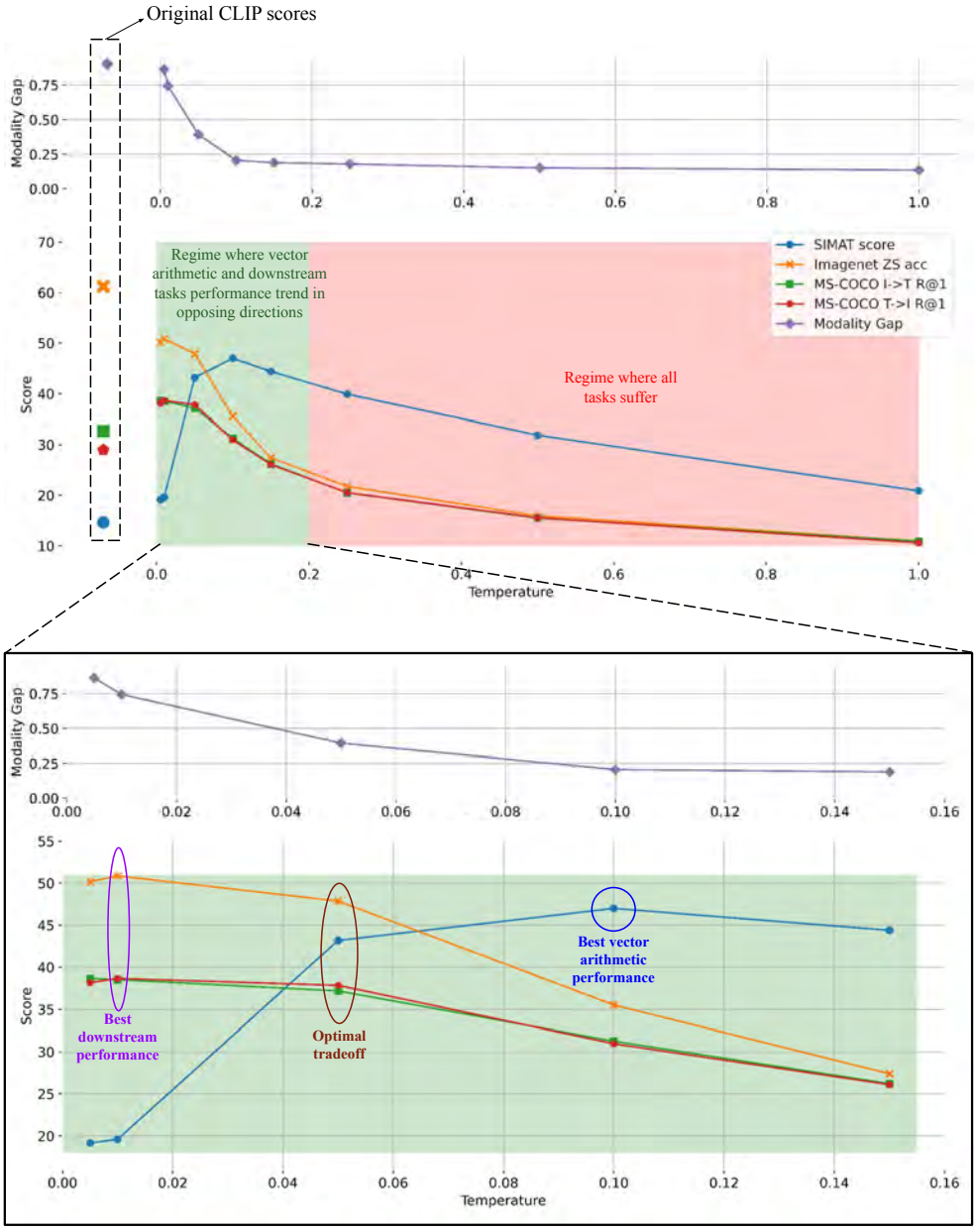



Fig. 5.13 Visualisation of the relationships between the modality gap, downstream task performance and vector arithmetic performance

we performed the fine-tuning on Imagenet instead. Further, we observe that regardless of temperature, fine-tuning always reduces the modality gap when compared to the original CLIP.

Next, we investigate the relationships between the three quantities of interest. From Figure 5.13, we observe two interesting regimes emerging:

1. $\tau = 0 - 0.2$: In this regime, we observe contrasting behaviour between the vector arithmetic performance and the downstream task performance. As illustrated in Figure 5.13, there is an optimal cross-over point at which we can preserve reasonable vector arithmetic and downstream performance. However, on either side of this cross-over point, one quantity improves while the other degrades significantly. Since the modality gap is monotonically decreasing in this regime, it hints towards the fact that the modality gap is not entirely predictive of downstream task or vector arithmetic performance.

2. $\tau = 0.2 - 1$: In this regime, both vector arithmetic and downstream task performance degrade despite the modality gap reducing consistently. This regime indicates that reducing the modality gap is not always beneficial for downstream tasks and vector arithmetic performance.

These results call into question our original hypothesis that reducing the modality gap should always improve vector arithmetic performance. However, we resolve this ambiguity through the lens of *Alignment* and *Uniformity* (Refer Section 4.5). Intuitively, a high alignment implies that the paired image-text embeddings are close together, and hence implies a small modality gap. Further, a high uniformity signifies that the image-text embeddings are all spread out very far apart from each other, uniformly covering the hypersphere. Recall from Figure 4.15b, at very small temperatures, the alignment and uniformity are extremely low. Therefore, since the image and text embeddings are far apart, their distances are not calibrated[11]. Hence, this leads to the observed poor vector arithmetic performance. Contrarily, at very high temperatures, the image and text embeddings have high alignment and high uniformity scores. Due to this, all the image-text embeddings will be spread out on the unit hypersphere, far from each other. This leads to the intra-image and intra-text distance distributions tending towards a uniform distribution. Due to this, we throw out important discriminative information between embeddings that is required for the vector arithmetic task. As a result of this degeneracy, vector arithmetic performance drops. Hence, by viewing the fine-tuning experiment through the lens of alignment and uniformity, we justify that vector arithmetic performance does not always improve with smaller modality gaps.

## 5.4 Summary

In this chapter, we have reflected on why the existence of the modality gap in the embedding spaces of vision-language models matters. In Section 5.1, we showed that it can hamper effective visualisation of these embedding spaces, proposed a simple technique to improve visualisation, and thereby gleaned a better understanding of these embedding spaces. In Section 5.2, we inspected the pathology of the intra-modal similarities of CLIP's embedding space and studied how this affects few-shot classification methods. By proposing a simple fix that leverages inter-modal similarities to perform intra-modal comparisons, we achieved state-of-the-art results for few-shot classification, without fine-tuning.

[11] By calibrated, here we mean that the intra-image distances and intra-text distances do not have the same scale (see Figure 5.11a).

Finally, in Section 5.3 we examined the implications of the modality gap on downstream tasks and vector arithmetic performance.

# Chapter 6

# Conclusions and Future Work

In this thesis, we presented a non-intuitive phenomenon that occurs in the embedding spaces of vision-language models, called the *Modality Gap*. We conducted a deep dive into the modality gap, and hypothesised several reasons for its existence. To this end, we broke down the contrastive loss (used to train CLIP-like models) into its constituent components. We then formulated several toy simulations to understand the effects of each component on the loss, and subsequently the modality gap. Having understood the factors causing the modality gap, we moved on to investigating the implications of the modality gap. We showed that the modality gap hindered effective visualisation of the embedding spaces of CLIP-like models, and proposed a simple fix. Using this, we showed visualisations that furnish greater interpretability. Further, we discussed the implications of the modality gap on different downstream tasks. Our analysis suggested that using intra-modal similarities is sub-optimal as compared to inter-modal similarities for image-image comparison. We fix this pathology in previous few-shot classification methods, by proposing a KL-divergence based method for leveraging the reliable inter-modal similarities for performing an image-image matching. This method achieves state-of-the-art results across a wide suite of datasets for the few-shot image classification task.

Despite the interesting results and analyses showcased by our work, there are still lots of unsolved puzzles to study. We enlist a few of them here:

- **A concrete theoretical explanation for the *Modality Gap*.** In Chapter 4, we presented a myriad of experimental analyses and results to uncover the formation and existence of the modality gap. In spite of these results, we still do not have a strong theoretical grasp on the existence of the modality gap. A potential research direction would be to extend the theoretical analysis conducted in Section 4.2 to the real-world setting, which can help glean better insights into the phenomenon.

- **Analysing the true potential of CLIP's uni-modal spaces.** In Section 5.2, we conducted experiments to showcase that the intra-modal similarities of CLIP are unreliable. However, the extent of this is unclear. Can these embedding spaces be used under certain limited conditions or

are they unreliable at a global level? Answering this question could help derive more effective vision-language pre-training methods.

- **Stronger understanding of vector arithmetic performance.** In Section 5.3, we discussed the effects of the modality gap on vector arithmetic performance, but did so through a simple empirical analysis. However, several questions remain as to what the optimal levels of distance calibration and modality gap should be for attaining the best vector arithmetic performance.

# References


[1] Adadi, A. and Berrada, M. (2018). Peeking inside the black-box: a survey on explainable artificial intelligence (xai). *IEEE access*, 6:52138–52160.

[2] Aghajanyan, A., Huang, B., Ross, C., Karpukhin, V., Xu, H., Goyal, N., Okhonko, D., Joshi, M., Ghosh, G., Lewis, M., et al. (2022). Cm3: A causal masked multimodal model of the internet. *arXiv preprint arXiv:2201.07520*.

[3] Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al. (2022). Flamingo: a visual language model for few-shot learning. *arXiv preprint arXiv:2204.14198*.

[4] Albanie, S., Nagrani, A., Vedaldi, A., and Zisserman, A. (2018). Emotion recognition in speech using cross-modal transfer in the wild. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 292–301.

[5] Albelwi, S. (2022). Survey on self-supervised learning: Auxiliary pretext tasks and contrastive learning methods in imaging. *Entropy*, 24(4):551.

[6] Andrew, G., Arora, R., Bilmes, J., and Livescu, K. (2013). Deep canonical correlation analysis. In *International conference on machine learning*, pages 1247–1255. PMLR.

[7] Artetxe, M., Labaka, G., and Agirre, E. (2018). Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

[8] Arvanitidis, G., Hauberg, S., and Schölkopf, B. (2020). Geometrically enriched latent spaces. *arXiv preprint arXiv:2008.00565*.

[9] Bahrick, L. E. and Lickliter, R. (2000). Intersensory redundancy guides attentional selectivity and perceptual learning in infancy. *Developmental psychology*, 36(2):190.

[10] Bai, J., Liu, C., Ni, F., Wang, H., Hu, M., Guo, X., and Cheng, L. (2022). Lat: Latent translation with cycle-consistency for video-text retrieval. *arXiv preprint arXiv:2207.04858*.

[11] Bain, M., Nagrani, A., Varol, G., and Zisserman, A. (2022). A clip-hitchhiker's guide to long video retrieval. *arXiv preprint arXiv:2205.08508*.

[12] Bao, Y., Wu, M., Chang, S., and Barzilay, R. (2019). Few-shot text classification with distributional signatures. *arXiv preprint arXiv:1908.06039*.

[13] Bardes, A., Ponce, J., and LeCun, Y. (2021). Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*.

[14] Barlow, H. B. (1989). Unsupervised learning. *Neural computation*, 1(3):295–311.

[15] Barraco, M., Cornia, M., Cascianelli, S., Baraldi, L., and Cucchiara, R. (2022). The unreasonable effectiveness of clip features for image captioning: An experimental analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4662–4670.

[16] Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. (2021). On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

[17] Borjon, J. I., Schroer, S. E., Bambach, S., Slone, L. K., Abney, D. H., Crandall, D. J., and Smith, L. B. (2018). A view of their own: Capturing the egocentric view of infants and toddlers with head-mounted cameras. *JoVE (Journal of Visualized Experiments)*, 1(140):e58445.

[18] Bossard, L., Guillaumin, M., and Gool, L. V. (2014). Food-101–mining discriminative components with random forests. In *European conference on computer vision*, pages 446–461. Springer.

[19] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

[20] Bushnell, E. W. (1994). A dual-processing approach to cross-modal matching: Implications for development. *The development of intersensory perception: Comparative perspectives*, pages 19–38.

[21] Calvert, G. A. (2001). Crossmodal processing in the human brain: insights from functional neuroimaging studies. *Cerebral cortex*, 11(12):1110–1123.

[22] Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. (2020). Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924.

[23] Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. (2021). Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660.

[24] Castro, S. and Heilbron, F. C. (2022). Fitclip: Refining large-scale pretrained image-text models for zero-shot video understanding tasks. *arXiv preprint arXiv:2203.13371*.

[25] Chen, D., Wu, Z., Liu, F., Yang, Z., Huang, Y., Bao, Y., and Zhou, E. (2022). Prototypical contrastive language image pretraining. *arXiv preprint arXiv:2206.10996*.

[26] Chen, P., Liu, S., and Jia, J. (2021a). Jigsaw clustering for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11526–11535.

[27] Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020a). A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

[28] Chen, T., Kornblith, S., Swersky, K., Norouzi, M., and Hinton, G. E. (2020b). Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33:22243–22255.

[29] Chen, X., Fan, H., Girshick, R., and He, K. (2020c). Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*.

[30] Chen, X. and He, K. (2021). Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758.

[31] Chen, X., Xie, S., and He, K. (2021b). An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9640–9649.

[32] Cho, J., Yoon, S., Kale, A., Dernoncourt, F., Bui, T., and Bansal, M. (2022). Fine-grained image captioning with clip reward. *arXiv preprint arXiv:2205.13115*.

[33] Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., and Vedaldi, A. (2014). Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613.

[34] Colon-Hernandez, P., Havasi, C., Alonso, J., Huggins, M., and Breazeal, C. (2021). Combining pre-trained language models and structured knowledge. *arXiv preprint arXiv:2101.12294*.

[35] Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V. (2019). Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

[36] Conneau, A., Lample, G., Ranzato, M., Denoyer, L., and Jégou, H. (2017). Word translation without parallel data. *arXiv preprint arXiv:1710.04087*.

[37] Cosentino, R., Sengupta, A., Avestimehr, S., Soltanolkotabi, M., Ortega, A., Willke, T., and Tepper, M. (2022). Toward a geometrical understanding of self-supervised contrastive learning. *arXiv preprint arXiv:2205.06926*.

[38] Couairon, G., Cord, M., Douze, M., and Schwenk, H. (2021). Embedding arithmetic for text-driven image transformation. *arXiv preprint arXiv:2112.03162*.

[39] Couairon, G., Douze, M., Cord, M., and Schwenk, H. (2022). Embedding arithmetic of multimodal queries for image retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4950–4958.

[40] Cui, Z., Li, K., Gu, L., Su, S., Gao, P., Jiang, Z., Qiao, Y., and Harada, T. (2022). You only need 90k parameters to adapt light: A light weight transformer for image enhancement and exposure correction. *arXiv preprint arXiv:2205.14871v3*.

[41] De Cao, N. and Aziz, W. (2020). The power spherical distribution. *arXiv preprint arXiv:2006.04437*.

[42] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

[43] Derpanis, K. G. (2008). The bhattacharyya measure. *Mendeley Computer*, 1(4):1990–1992.

[44] Desai, K. and Johnson, J. (2021). Virtex: Learning visual representations from textual annotations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11162–11173.

[45] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[46] Doersch, C., Gupta, A., and Efros, A. A. (2015). Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430.

[47] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

[48] Dou, Z.-Y., Kamath, A., Gan, Z., Zhang, P., Wang, J., Li, L., Liu, Z., Liu, C., LeCun, Y., Peng, N., et al. (2022a). Coarse-to-fine vision-language pre-training with fusion in the backbone. *arXiv preprint arXiv:2206.07643*.

[49] Dou, Z.-Y., Xu, Y., Gan, Z., Wang, J., Wang, S., Wang, L., Zhu, C., Zhang, P., Yuan, L., Peng, N., et al. (2022b). 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*, pages 18166–18176.

[50] Edelman, G. M. (1987). *Neural Darwinism: The theory of neuronal group selection.* Basic books.

[51] Ellis, R. and Tucker, M. (2000). Micro-affordance: The potentiation of components of action by seen objects. *British journal of psychology*, 91(4):451–471.

[52] Ethayarajh, K., Duvenaud, D., and Hirst, G. (2018). Towards understanding linear word analogies. *arXiv preprint arXiv:1810.04882*.

[53] Faghri, F., Fleet, D. J., Kiros, J. R., and Fidler, S. (2017). Vse++: Improving visual-semantic embeddings with hard negatives. *arXiv preprint arXiv:1707.05612*.

[54] Fei-Fei, L., Fergus, R., and Perona, P. (2004). Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, pages 178–178. IEEE.

[55] Fournier, L., Dupoux, E., and Dunbar, E. (2020). Analogies minus analogy test: measuring regularities in word embeddings. *arXiv preprint arXiv:2010.03446*.

[56] Gao, P., Geng, S., Zhang, R., Ma, T., Fang, R., Zhang, Y., Li, H., and Qiao, Y. (2021). Clip-adapter: Better vision-language models with feature adapters. *arXiv preprint arXiv:2110.04544*.

[57] Gao, Y., Liu, J., Xu, Z., Zhang, J., Li, K., and Shen, C. (2022). Pyramidclip: Hierarchical feature alignment for vision-language model pretraining. *arXiv preprint arXiv:2204.14095*.

[58] Gidaris, S., Singh, P., and Komodakis, N. (2018). Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*.

[59] Goel, S., Bansal, H., Bhatia, S., Rossi, R., Vinay, V., and Grover, A. (2022). Cyclip: Cyclic contrastive language-image pretraining. *arXiv preprint arXiv:2205.14459*.

[60] Gong, Y., Ke, Q., Isard, M., and Lazebnik, S. (2014). A multi-view embedding space for modeling internet images, tags, and their semantics. *International journal of computer vision*, 106(2):210–233.

[61] Gower, J. C. (1975). Generalized procrustes analysis. *Psychometrika*, 40(1):33–51.

[62] Goyal, P., Duval, Q., Seessel, I., Caron, M., Singh, M., Misra, I., Sagun, L., Joulin, A., and Bojanowski, P. (2022). Vision models are more robust and fair when pretrained on uncurated images without supervision. *arXiv preprint arXiv:2202.08360*.

[63] Graf, F., Hofer, C., Niethammer, M., and Kwitt, R. (2021). Dissecting supervised constrastive learning. In *International Conference on Machine Learning*, pages 3821–3830. PMLR.

[64] Grave, E., Joulin, A., and Berthet, Q. (2019). Unsupervised alignment of embeddings with wasserstein procrustes. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1880–1890. PMLR.

[65] Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al. (2020). Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284.

[66] Gu, X., Lin, T.-Y., Kuo, W., and Cui, Y. (2021). Open-vocabulary object detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921*.

[67] Guo, W., Wang, J., and Wang, S. (2019). Deep multimodal representation learning: A survey. *IEEE Access*, 7:63373–63394.

[68] Gutmann, M. and Hyvärinen, A. (2010). Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings.

[69] He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738.

[70] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

[71] Helber, P., Bischke, B., Dengel, A., and Borth, D. (2019). Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226.

[72] Henaff, O. (2020). Data-efficient image recognition with contrastive predictive coding. In *International conference on machine learning*, pages 4182–4192. PMLR.

[73] Hjelm, R. D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., and Bengio, Y. (2018). Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*.

[74] Hocking, J. G. and Young, G. S. (1988). *Topology*. Courier Corporation.

[75] Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. (2019). Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

[76] Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., et al. (2019). Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1314–1324.

[77] Hu, P., Zhen, L., Peng, D., and Liu, P. (2019). Scalable deep multimodal learning for cross-modal retrieval. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, pages 635–644.

[78] Huang, T., Chu, J., and Wei, F. (2022). Unsupervised prompt learning for vision-language models. *arXiv preprint arXiv:2204.03649*.

[79] Hunnius, S. (2022). Early cognitive development: five lessons from infant learning. In *Oxford Research Encyclopedia of Psychology*.

[80] Huo, Y., Zhang, M., Liu, G., Lu, H., Gao, Y., Yang, G., Wen, J., Zhang, H., Xu, B., Zheng, W., et al. (2021). Wenlan: Bridging vision and language by large-scale multi-modal pre-training. *arXiv preprint arXiv:2103.06561*.

[81] Ilharco, G., Wortsman, M., Yitzhak Gadre, S., Song, S., Hajishirzi, H., Kornblith, S., Farhadi, A., and Schmidt, L. (2022). Patching open-vocabulary models by interpolating weights. *arXiv preprint arXiv:2208.05592*.

[82] Jaiswal, A., Babu, A. R., Zadeh, M. Z., Banerjee, D., and Makedon, F. (2020). A survey on contrastive self-supervised learning. *Technologies*, 9(1):2.

[83] Jia, C., Yang, Y., Xia, Y., Chen, Y.-T., Parekh, Z., Pham, H., Le, Q., Sung, Y.-H., Li, Z., and Duerig, T. (2021). Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR.

[84] Jin, Q., Chen, J., Chen, S., Xiong, Y., and Hauptmann, A. (2016). Describing videos using multi-modal fusion. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 1087–1091.

[85] Jing, L. and Tian, Y. (2020). Self-supervised visual feature learning with deep neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(11):4037–4058.

[86] Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. (2016). Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

[87] Kamachi, M., Hill, H., Lander, K., and Vatikiotis-Bateson, E. (2003). Putting the face to the voice': Matching identity across modality. *Current Biology*, 13(19):1709–1714.

[88] Kamath, A., Singh, M., LeCun, Y., Synnaeve, G., Misra, I., and Carion, N. (2021). Mdetr-modulated detection for end-to-end multi-modal understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1780–1790.

[89] Kanazawa, A., Jacobs, D. W., and Chandraker, M. (2016). Warpnet: Weakly supervised matching for single-view reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3253–3261.

[90] Kazakos, E., Nagrani, A., Zisserman, A., and Damen, D. (2019). Epic-fusion: Audio-visual temporal binding for egocentric action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5492–5501.

[91] Kiela, D., Bhooshan, S., Firooz, H., Perez, E., and Testuggine, D. (2019). Supervised multimodal bitransformers for classifying images and text. *arXiv preprint arXiv:1909.02950*.

[92] Kim, W., Son, B., and Kim, I. (2021). Vilt: Vision-and-language transformer without convolution or region supervision. In *International Conference on Machine Learning*, pages 5583–5594. PMLR.

[93] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[94] Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Skip-thought vectors. *Advances in neural information processing systems*, 28.

[95] Köppen, M. (2000). The curse of dimensionality. In *5th online world conference on soft computing in industrial applications (WSC5)*, volume 1, pages 4–8.

[96] Krause, J., Stark, M., Deng, J., and Fei-Fei, L. (2013). 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561.

[97] Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.-J., Shamma, D. A., et al. (2017). Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73.

[98] Krzanowski, W. (2000). *Principles of multivariate analysis*, volume 23. OUP Oxford.

[99] Kumar, P., Rawat, P., and Chauhan, S. (2022). Contrastive self-supervised learning: review, progress, challenges and future research directions. *International Journal of Multimedia Information Retrieval*, pages 1–28.

[100] Kwon, G., Cai, Z., Ravichandran, A., Bas, E., Bhotika, R., and Soatto, S. (2022). Masked vision and language modeling for multi-modal representation learning. *arXiv preprint arXiv:2208.02131*.

[101] Lachs, L. (2017). Multi-modal perception. *Noba textbook series: Psychology. Champaign: DEF Publishers*.

[102] Lampert, C. H., Nickisch, H., and Harmeling, S. (2009). Learning to detect unseen object classes by between-class attribute transfer. In *2009 IEEE conference on computer vision and pattern recognition*, pages 951–958. IEEE.

[103] Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. (2019). Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

[104] Landau, B., Gleitman, L. R., and Landau, B. (2009). *Language and experience: Evidence from the blind child*, volume 8. Harvard University Press.

[105] Le, H., Pino, J., Wang, C., Gu, J., Schwab, D., and Besacier, L. (2021). Lightweight adapter tuning for multilingual speech translation. *arXiv preprint arXiv:2106.01463*.

[106] LeCun, Y. and Misra, I. (2021). Self-supervised learning: The dark matter of intelligence. *URL https://ai. facebook. com/blog/self-supervised-learning-the-dark-matter-of-intelligence*.

[107] Lee, K.-H., Chen, X., Hua, G., Hu, H., and He, X. (2018). Stacked cross attention for image-text matching. In *Proceedings of the European conference on computer vision (ECCV)*, pages 201–216.

[108] Levy, O. and Goldberg, Y. (2014). Linguistic regularities in sparse and explicit word representations. In *Proceedings of the eighteenth conference on computational natural language learning*, pages 171–180.

[109] Li, C., Yang, J., Zhang, P., Gao, M., Xiao, B., Dai, X., Yuan, L., and Gao, J. (2021a). Efficient self-supervised vision transformers for representation learning. *arXiv preprint arXiv:2106.09785*.

[110] Li, J., He, X., Wei, L., Qian, L., Zhu, L., Xie, L., Zhuang, Y., Tian, Q., and Tang, S. (2022a). Fine-grained semantically aligned vision-language pre-training. *arXiv preprint arXiv:2208.02515*.

[111] Li, J., Li, D., Xiong, C., and Hoi, S. (2022b). Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *arXiv preprint arXiv:2201.12086*.

[112] Li, J., Mo, W., Qiang, W., Su, B., and Zheng, C. (2022c). Supporting vision-language model inference with causality-pruning knowledge prompt. *arXiv preprint arXiv:2205.11100*.

[113] Li, J., Selvaraju, R., Gotmare, A., Joty, S., Xiong, C., and Hoi, S. C. H. (2021b). Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34:9694–9705.

[114] Li, L. H., Yatskar, M., Yin, D., Hsieh, C.-J., and Chang, K.-W. (2019). Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*.

[115] Li, L. H., Zhang, P., Zhang, H., Yang, J., Li, C., Zhong, Y., Wang, L., Yuan, L., Zhang, L., Hwang, J.-N., et al. (2022d). Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10965–10975.

[116] Li, X., Yin, X., Li, C., Zhang, P., Hu, X., Zhang, L., Wang, L., Hu, H., Dong, L., Wei, F., et al. (2020). Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*, pages 121–137. Springer.

[117] Li, Y., Liang, F., Zhao, L., Cui, Y., Ouyang, W., Shao, J., Yu, F., and Yan, J. (2021c). Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. *arXiv preprint arXiv:2110.05208*.

[118] Liang, W., Zhang, Y., Kwon, Y., Yeung, S., and Zou, J. (2022). Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning. *arXiv preprint arXiv:2203.02053*.

[119] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

[120] Linardatos, P., Papastefanopoulos, V., and Kotsiantis, S. (2020). Explainable ai: A review of machine learning interpretability methods. *Entropy*, 23(1):18.

[121] Liu, Y., Albanie, S., Nagrani, A., and Zisserman, A. (2019a). Use what you have: Video retrieval using representations from collaborative experts. *arXiv preprint arXiv:1907.13487*.

[122] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019b). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

[123] Loshchilov, I. and Hutter, F. (2016). Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*.

[124] Loshchilov, I. and Hutter, F. (2017). Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

[125] Lu, Q., Dou, D., and Nguyen, T. H. (2021). Parameter-efficient domain knowledge integration from multiple sources for biomedical pre-trained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3855–3865.

[126] Lu, Y., Liu, J., Zhang, Y., Liu, Y., and Tian, X. (2022). Prompt distribution learning. *arXiv preprint arXiv:2205.03340*.

[127] Ma, Y., Xu, G., Sun, X., Yan, M., Zhang, J., and Ji, R. (2022). X-clip: End-to-end multi-grained contrastive learning for video-text retrieval. *arXiv preprint arXiv:2207.07285*.

[128] Maji, S., Rahtu, E., Kannala, J., Blaschko, M., and Vedaldi, A. (2013). Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*.

[129] Mardia, K. V., Jupp, P. E., and Mardia, K. (2000). *Directional statistics*, volume 2. Wiley Online Library.

[130] Maunder, C. R. F. (1996). *Algebraic topology*. Courier Corporation.

[131] Menéndez, M., Pardo, J., Pardo, L., and Pardo, M. (1997). The jensen-shannon divergence. *Journal of the Franklin Institute*, 334(2):307–318.

[132] Mensink, T., Gavves, E., and Snoek, C. G. (2014). Costa: Co-occurrence statistics for zero-shot classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2441–2448.

[133] Micikevicius, P., Narang, S., Alben, J., Diamos, G., Elsen, E., Garcia, D., Ginsburg, B., Houston, M., Kuchaiev, O., Venkatesh, G., et al. (2017). Mixed precision training. *arXiv preprint arXiv:1710.03740*.

[134] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[135] Mikolov, T., Yih, W.-t., and Zweig, G. (2013b). Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pages 746–751.

[136] Miller, G. A. (1998). *WordNet: An electronic lexical database*. MIT press.

[137] Minderer, M., Gritsenko, A., Stone, A., Neumann, M., Weissenborn, D., Dosovitskiy, A., Mahendran, A., Arnab, A., Dehghani, M., Shen, Z., et al. (2022). Simple open-vocabulary object detection with vision transformers. *arXiv preprint arXiv:2205.06230*.

[138] Misra, I. and Maaten, L. v. d. (2020). Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717.

[139] Mitchell, T. M. and Mitchell, T. M. (1997). *Machine learning*, volume 1. McGraw-hill New York.

[140] Mokady, R., Hertz, A., and Bermano, A. H. (2021). Clipcap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734*.

[141] Mu, N., Kirillov, A., Wagner, D., and Xie, S. (2021). Slip: Self-supervision meets language-image pre-training. *arXiv preprint arXiv:2112.12750*.

[142] Nagrani, A. (2020). *Video understanding using multimodal deep learning*. PhD thesis, University of Oxford.

[143] Nagrani, A., Albanie, S., and Zisserman, A. (2018a). Learnable pins: Cross-modal embeddings for person identity. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 71–88.

[144] Nagrani, A., Albanie, S., and Zisserman, A. (2018b). Seeing voices and hearing faces: Cross-modal biometric matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8427–8436.

[145] Nagrani, A., Chung, J. S., Albanie, S., and Zisserman, A. (2020a). Disentangled speech embeddings using cross-modal self-supervision. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6829–6833. IEEE.

[146] Nagrani, A., Sun, C., Ross, D., Sukthankar, R., Schmid, C., and Zisserman, A. (2020b). Speech2action: Cross-modal supervision for action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10317–10326.

[147] Nguyen, Q., Mukkamala, M. C., and Hein, M. (2018). On the loss landscape of a class of deep neural networks with no bad local valleys. *arXiv preprint arXiv:1809.10749*.

[148] Ni, B., Peng, H., Chen, M., Zhang, S., Meng, G., Fu, J., Xiang, S., and Ling, H. (2022). Expanding language-image pretrained models for general video recognition. *arXiv preprint arXiv:2208.02816*.

[149] Nilsback, M.-E. and Zisserman, A. (2008). Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE.

[150] Noroozi, M. and Favaro, P. (2016). Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer.

[151] Novotny, D., Albanie, S., Larlus, D., and Vedaldi, A. (2018). Self-supervised learning of geometrically stable features through probabilistic introspection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3637–3645.

[152] Oord, A. v. d., Li, Y., and Vinyals, O. (2018). Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

[153] Parkhi, O. M., Vedaldi, A., and Zisserman, A. (2015). Deep face recognition.

[154] Parkhi, O. M., Vedaldi, A., Zisserman, A., and Jawahar, C. (2012). Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE.

[155] Pedersen, T., Patwardhan, S., Michelizzi, J., et al. (2004). Wordnet:: Similarity-measuring the relatedness of concepts. In *AAAI*, volume 4, pages 25–29.

[156] Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

[157] Peyré, G., Cuturi, M., et al. (2019). Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607.

[158] Pfeiffer, J., Rücklé, A., Poth, C., Kamath, A., Vulić, I., Ruder, S., Cho, K., and Gurevych, I. (2020a). Adapterhub: A framework for adapting transformers. *arXiv preprint arXiv:2007.07779*.

[159] Pfeiffer, J., Vulić, I., Gurevych, I., and Ruder, S. (2020b). Mad-x: An adapter-based framework for multi-task cross-lingual transfer. *arXiv preprint arXiv:2005.00052*.

[160] Pham, H., Dai, Z., Ghiasi, G., Liu, H., Yu, A. W., Luong, M.-T., Tan, M., and Le, Q. V. (2021). Combined scaling for zero-shot transfer learning. *arXiv preprint arXiv:2111.10050*.

[161] Piaget, J. and Cook, M. T. (1952). The origins of intelligence in children.

[162] Poole, B., Ozair, S., Van Den Oord, A., Alemi, A., and Tucker, G. (2019). On variational bounds of mutual information. In *International Conference on Machine Learning*, pages 5171–5180. PMLR.

[163] Qiao, H., Zhang, P., Wang, D., and Zhang, B. (2012). An explicit nonlinear mapping for manifold learning. *IEEE transactions on cybernetics*, 43(1):51–63.

[164] Radford, A., Jozefowicz, R., and Sutskever, I. (2017). Learning to generate reviews and discovering sentiment. *arXiv preprint arXiv:1704.01444*.

[165] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021). Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR.

[166] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

[167] Rao, Y., Zhao, W., Chen, G., Tang, Y., Zhu, Z., Huang, G., Zhou, J., and Lu, J. (2021). Denseclip: Language-guided dense prediction with context-aware prompting. *arXiv preprint arXiv:2112.01518*.

[168] Rocco, I., Arandjelovic, R., and Sivic, J. (2017). Convolutional neural network architecture for geometric matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6148–6157.

[169] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252.

[170] Sariyildiz, M. B., Perez, J., and Larlus, D. (2020). Learning visual representations with caption annotations. In *European Conference on Computer Vision*, pages 153–170. Springer.

[171] Schiappa, M. C., Rawat, Y. S., and Shah, M. (2022). Self-supervised learning for videos: A survey. *arXiv preprint arXiv:2207.00419*.

[172] Schönemann, P. H. (1966). A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10.

[173] Schuhmann, C., Vencu, R., Beaumont, R., Kaczmarczyk, R., Mullis, C., Katta, A., Coombes, T., Jitsev, J., and Komatsuzaki, A. (2021). Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*.

[174] Sennrich, R., Haddow, B., and Birch, A. (2015). Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

[175] Shao, H., Kumar, A., and Thomas Fletcher, P. (2018). The riemannian geometry of deep generative models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 315–323.

[176] Sharma, P., Ding, N., Goodman, S., and Soricut, R. (2018). 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)*, pages 2556–2565.

[177] Shen, S., Li, C., Hu, X., Xie, Y., Yang, J., Zhang, P., Rohrbach, A., Gan, Z., Wang, L., Yuan, L., et al. (2022). K-lite: Learning transferable visual models with external knowledge. *arXiv preprint arXiv:2204.09222*.

[178] Shen, S., Li, L. H., Tan, H., Bansal, M., Rohrbach, A., Chang, K.-W., Yao, Z., and Keutzer, K. (2021). How much can clip benefit vision-and-language tasks? *arXiv preprint arXiv:2107.06383*.

[179] Shukla, A., Uppal, S., Bhagat, S., Anand, S., and Turaga, P. (2018). Geometry of deep generative models for disentangled representations. In *Proceedings of the 11th Indian Conference on Computer Vision, Graphics and Image Processing*, pages 1–8.

[180] Shvetsova, N., Chen, B., Rouditchenko, A., Thomas, S., Kingsbury, B., Feris, R. S., Harwath, D., Glass, J., and Kuehne, H. (2022). Everything at once-multi-modal fusion transformer for video retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20020–20029.

[181] Singh, A., Hu, R., Goswami, V., Couairon, G., Galuba, W., Rohrbach, M., and Kiela, D. (2021). Flava: A foundational language and vision alignment model. *arXiv preprint arXiv:2112.04482*.

[182] Smith, H. M., Dunn, A. K., Baguley, T., and Stacey, P. C. (2016). Matching novel face and voice identity using static and dynamic facial images. *Attention, Perception, & Psychophysics*, 78(3):868–879.

[183] Smith, L. and Gasser, M. (2005). The development of embodied cognition: Six lessons from babies. *Artificial life*, 11(1-2):13–29.

[184] Snell, J., Swersky, K., and Zemel, R. (2017). Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30.

[185] So, J., Oh, C., Shin, M., and Song, K. (2022). Multi-modal mixup for robust fine-tuning. *arXiv preprint arXiv:2203.03897*.

[186] Song, H., Dong, L., Zhang, W.-N., Liu, T., and Wei, F. (2022). Clip models are few-shot learners: Empirical studies on vqa and visual entailment. *arXiv preprint arXiv:2203.07190*.

[187] Soomro, K., Zamir, A. R., and Shah, M. (2012). Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*.

[188] Stein, B. E. and Meredith, M. A. (1993). *The merging of the senses.* The MIT press.

[189] Stickland, A. C., Li, X., and Ghazvininejad, M. (2020). Recipes for adapting pre-trained monolingual and multilingual models to machine translation. *arXiv preprint arXiv:2004.14911*.

[190] Straub, J. (2017). Bayesian inference with the von-mises-fisher distribution in 3d.

[191] Sun, K. and Marchand-Maillet, S. (2014). An information geometry of statistical manifold learning. In *International Conference on Machine Learning*, pages 1–9. PMLR.

[192] Sun, X., Hu, P., and Saenko, K. (2022). Dualcoop: Fast adaptation to multi-label recognition with limited annotations. *arXiv preprint arXiv:2206.09541*.

[193] Sung, Y.-L., Cho, J., and Bansal, M. (2022). Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5227–5237.

[194] Thomas, B., Kessler, S., and Karout, S. (2022). Efficient adapter transfer of self-supervised speech models for automatic speech recognition. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7102–7106. IEEE.

[195] Thomee, B., Shamma, D. A., Friedland, G., Elizalde, B., Ni, K., Poland, D., Borth, D., and Li, L.-J. (2016). Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73.

[196] Tian, Y., Krishnan, D., and Isola, P. (2020). Contrastive multiview coding. In *European conference on computer vision*, pages 776–794. Springer.

[197] Udandarao, V., Maiti, A., Srivatsav, D., Vyalla, S. R., Yin, Y., and Shah, R. R. (2020). Cobra: Contrastive bi-modal representation algorithm. *arXiv preprint arXiv:2005.03687*.

[198] Van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

[199] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

[200] Vellido, A. (2020). The importance of interpretability and visualization in machine learning for applications in medicine and health care. *Neural computing and applications*, 32(24):18069–18083.

[201] Villani, C. (2009). *Optimal transport: old and new*, volume 338. Springer.

[202] Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al. (2016). Matching networks for one shot learning. *Advances in neural information processing systems*, 29.

[203] Wang, F. and Liu, H. (2021). Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2495–2504.

[204] Wang, F., Xiang, X., Cheng, J., and Yuille, A. L. (2017). Normface: L2 hypersphere embedding for face verification. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1041–1049.

[205] Wang, L., Li, Y., and Lazebnik, S. (2016). Learning deep structure-preserving image-text embeddings. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5005–5013.

[206] Wang, P., Yang, A., Men, R., Lin, J., Bai, S., Li, Z., Ma, J., Zhou, C., Zhou, J., and Yang, H. (2022). Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. *arXiv preprint arXiv:2202.03052*.

[207] Wang, R., Tang, D., Duan, N., Wei, Z., Huang, X., Cao, G., Jiang, D., Zhou, M., et al. (2020). K-adapter: Infusing knowledge into pre-trained models with adapters. *arXiv preprint arXiv:2002.01808*.

[208] Wang, T. and Isola, P. (2020). Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pages 9929–9939. PMLR.

[209] Wang, Z., Yu, J., Yu, A. W., Dai, Z., Tsvetkov, Y., and Cao, Y. (2021). Simvlm: Simple visual language model pretraining with weak supervision. *arXiv preprint arXiv:2108.10904*.

[210] Watson, G. S. (1982). Distributions on the circle and sphere. *Journal of Applied Probability*, 19(A):265–280.

[211] Wortsman, M., Ilharco, G., Kim, J. W., Li, M., Kornblith, S., Roelofs, R., Lopes, R. G., Hajishirzi, H., Farhadi, A., Namkoong, H., et al. (2022). Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7959–7971.

[212] Xiang, L., Chen, Y., Chang, W., Zhan, Y., Lin, W., Wang, Q., and Shen, D. (2018). Deep-learning-based multi-modal fusion for fast mr reconstruction. *IEEE Transactions on Biomedical Engineering*, 66(7):2105–2114.

[213] Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., and Torralba, A. (2010). Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3485–3492. IEEE.

[214] Xu, J. and Durrett, G. (2018). Spherical latent spaces for stable variational autoencoders. *arXiv preprint arXiv:1808.10805*.

[215] Xu, X., Wu, C., Rosenman, S., Lal, V., and Duan, N. (2022). Bridge-tower: Building bridges between encoders in vision-language representation learning. *arXiv preprint arXiv:2206.08657*.

[216] Xu, Z., Guo, D., Tang, D., Su, Q., Shou, L., Gong, M., Zhong, W., Quan, X., Duan, N., and Jiang, D. (2020). Syntax-enhanced pre-trained model. *arXiv preprint arXiv:2012.14116*.

[217] Yang, H., Lin, J., Yang, A., Wang, P., Zhou, C., and Yang, H. (2022a). Prompt tuning for generative multimodal pretrained models. *arXiv preprint arXiv:2208.02532*.

[218] Yang, J., Duan, J., Tran, S., Xu, Y., Chanda, S., Chen, L., Zeng, B., Chilimbi, T., and Huang, J. (2022b). Vision-language pre-training with triple contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15671–15680.

[219] Yang, J., Xiao, G., Shen, Y., Jiang, W., Hu, X., Zhang, Y., and Peng, J. (2021). A survey of knowledge enhanced pre-trained models. *arXiv preprint arXiv:2110.00269*.

[220] Yao, L., Huang, R., Hou, L., Lu, G., Niu, M., Xu, H., Liang, X., Li, Z., Jiang, X., and Xu, C. (2021a). Filip: Fine-grained interactive language-image pre-training. *arXiv preprint arXiv:2111.07783*.

[221] Yao, Y., Zhang, A., Zhang, Z., Liu, Z., Chua, T.-S., and Sun, M. (2021b). Cpt: Colorful prompt tuning for pre-trained vision-language models. *arXiv preprint arXiv:2109.11797*.

[222] You, H., Zhou, L., Xiao, B., Codella, N., Cheng, Y., Xu, R., Chang, S.-F., and Yuan, L. (2022). Learning visual representation from modality-shared contrastive language-image pre-training. *arXiv preprint arXiv:2207.12661*.

[223] Yu, J., Wang, Z., Vasudevan, V., Yeung, L., Seyedhosseini, M., and Wu, Y. (2022a). Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*.

[224] Yu, J., Wang, Z., Vasudevan, V., Yeung, L., Seyedhosseini, M., and Wu, Y. (2022b). Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*.

[225] Zaadnoordijk, L., Besold, T. R., and Cusack, R. (2022). Lessons from infant learning for unsupervised machine learning. *Nature Machine Intelligence*, pages 1–11.

[226] Zang, Y., Li, W., Zhou, K., Huang, C., and Loy, C. C. (2022). Open-vocabulary detr with conditional matching. *arXiv preprint arXiv:2203.11876*.

[227] Zbontar, J., Jing, L., Misra, I., LeCun, Y., and Deny, S. (2021). Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR.

[228] Zeng, Y., Zhang, X., and Li, H. (2021). Multi-grained vision language pre-training: Aligning texts with visual concepts. *arXiv preprint arXiv:2111.08276*.

[229] Zhai, X., Wang, X., Mustafa, B., Steiner, A., Keysers, D., Kolesnikov, A., and Beyer, L. (2022). Lit: Zero-shot transfer with locked-image text tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18123–18133.

[230] Zhang, L., Qi, G.-J., Wang, L., and Luo, J. (2019). Aet vs. aed: Unsupervised representation learning by auto-encoding transformations rather than data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2547–2555.

[231] Zhang, M. and Ré, C. (2022). Contrastive adapters for foundation model group robustness. In *ICML 2022: Workshop on Spurious Correlations, Invariance and Stability*.

[232] Zhang, R., Fang, R., Gao, P., Zhang, W., Li, K., Dai, J., Qiao, Y., and Li, H. (2021a). Tip-adapter: Training-free clip-adapter for better vision-language modeling. *arXiv preprint arXiv:2111.03930*.

[233] Zhang, R., Guo, Z., Zhang, W., Li, K., Miao, X., Cui, B., Qiao, Y., Gao, P., and Li, H. (2022a). Pointclip: Point cloud understanding by clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8552–8562.

[234] Zhang, R., Isola, P., and Efros, A. A. (2016). Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer.

[235] Zhang, R., Qiu, L., Zhang, W., and Zeng, Z. (2021b). Vt-clip: Enhancing vision-language models with visual-guided texts. *arXiv preprint arXiv:2112.02399*.

[236] Zhang, R., Zeng, Z., and Guo, Z. (2022b). Can language understand depth? *arXiv preprint arXiv:2207.01077*.

[237] Zhang, R., Zhang, W., Fang, R., Gao, P., Li, K., Dai, J., Qiao, Y., and Li, H. (2022c). Tip-adapter: Training-free adaption of clip for few-shot classification. *arXiv preprint arXiv:2207.09519*.

[238] Zhang, Y., Jiang, H., Miura, Y., Manning, C. D., and Langlotz, C. P. (2020). Contrastive learning of medical visual representations from paired images and text. *arXiv preprint arXiv:2010.00747*.

[239] Zhou, K., Yang, J., Loy, C. C., and Liu, Z. (2021). Learning to prompt for vision-language models. *arXiv preprint arXiv:2109.01134*.

[240] Zhou, K., Yang, J., Loy, C. C., and Liu, Z. (2022). Conditional prompt learning for vision-language models. *arXiv preprint arXiv:2203.05557*.

[241] Zhu, B., Niu, Y., Han, Y., Wu, Y., and Zhang, H. (2022). Prompt-aligned gradient for prompt tuning. *arXiv preprint arXiv:2205.14865*.

# Appendix A

# Derivation for Simple Toy Experiment

We are given two fixed image points on the unit circle $S^1$, $I_1 = \begin{bmatrix} i_{1x} & i_{1y} \end{bmatrix}^T$ and $I_2 = \begin{bmatrix} i_{2x} & i_{2y} \end{bmatrix}^T$ (see Figure A.1). Our goal is to analytically derive the two text points $T_1 = \begin{bmatrix} t_{1x} & t_{1y} \end{bmatrix}^T$ and $T_2 = \begin{bmatrix} t_{2x} & t_{2y} \end{bmatrix}^T$ on the unit circle that minimise the contrastive loss for different settings of $I_1$ and $I_2$. Recall that the contrastive loss is formulated as:

$$\begin{aligned} L_{T\to I} &= -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp(T_i \cdot I_i/\tau)}{\sum_{j=1}^{N}\exp(T_i \cdot I_j/\tau)} \\ L_{I\to T} &= -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp(I_i \cdot T_i/\tau)}{\sum_{j=1}^{N}\exp(I_i \cdot T_j/\tau)} \\ L &= \frac{1}{2}\left[L_{I\to T} + L_{T\to I}\right] \end{aligned} \tag{A.1}$$

where $\tau$ denotes the temperature and $A \cdot B$ denotes the dot product of vectors $A$ and $B$.

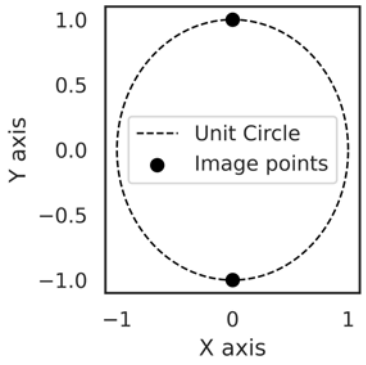



Fig. A.1 Depiction of one specific setting of the toy problem with $I_1 = \begin{bmatrix} 0 & 1 \end{bmatrix}^T$ and $I_2 = \begin{bmatrix} 0 & -1 \end{bmatrix}^T$.

Since we are working with two-dimensional points on $S^1$, the losses from Equation A.1 can be simplified to:

$$L_{I\to T} = -\frac{1}{2}[$$

$i_{1x}t_{1x} + i_{1y}t_{1y}$ Positive pair 1

$+i_{2x}t_{2x} + i_{2y}t_{2y}$ Positive pair 2

$-\log(\exp(i_{1x}t_{1x} + i_{1y}t_{1y}) + \exp(i_{1x}t_{2x} + i_{1y}t_{2y}))$ Denominator for positive pair 1

$-\log(\exp(i_{2x}t_{2x} + i_{2y}t_{2y}) + \exp(i_{2x}t_{1x} + i_{2y}t_{1y}))]$ Denominator for positive pair 2

$$L_{T\to I} = -\frac{1}{2}[$$

$i_{1x}t_{1x} + i_{1y}t_{1y}$ Positive pair 1

$+i_{2x}t_{2x} + i_{2y}t_{2y}$ Positive pair 2

$-\log(\exp(i_{1x}t_{1x} + i_{1y}t_{1y}) + \exp(i_{2x}t_{1x} + i_{2y}t_{1y}))$ Denominator for positive pair 1

$-\log(\exp(i_{2x}t_{2x} + i_{2y}t_{2y}) + \exp(i_{1x}t_{2x} + i_{1y}t_{2y}))]$ Denominator for positive pair 2

$$L = -\frac{1}{4}[$$

$2i_{1x}t_{1x} + 2i_{1y}t_{1y} + 2i_{2x}t_{2x} + 2i_{2y}t_{2y}$ Numerator terms

$-\log(A) - \log(B) - \log(C) - \log(D)]$ Denominator terms

where

$$\begin{aligned}
A &= \exp(i_{1x}t_{1x} + i_{1y}t_{1y}) + \exp(i_{1x}t_{2x} + i_{1y}t_{2y}) \\
B &= \exp(i_{2x}t_{2x} + i_{2y}t_{2y}) + \exp(i_{2x}t_{1x} + i_{2y}t_{1y}) \\
C &= \exp(i_{1x}t_{1x} + i_{1y}t_{1y}) + \exp(i_{2x}t_{1x} + i_{2y}t_{1y}) \\
D &= \exp(i_{2x}t_{2x} + i_{2y}t_{2y}) + \exp(i_{1x}t_{2x} + i_{1y}t_{2y})
\end{aligned}$$

Equipped with this loss formulation, we can now find the values of $T_1$ and $T_2$ that minimize the loss. However, since we are working on the unit circle, we need two additional norm constraints on $T_1$ and $T_2$:

$$\begin{aligned}
t_{1x}^2 + t_{1y}^2 &= 1 \\
t_{2x}^2 + t_{2y}^2 &= 1
\end{aligned}$$

We capture these two constraints in our loss optimisation by using two Lagrange multipliers. The final objective that has to be minimised with respect to $T_1$ and $T_2$ is:

$$O = L - \lambda(t_{1x}^2 - t_{1y}^2 - 1) - \theta(t_{2x}^2 + t_{2y}^2 - 1)$$

where $\lambda$ and $\theta$ are the corresponding lagrange multipliers.

To do the minimisation, we take partial derivatives of $O$ with respect to each of $t_{1x}, t_{1y}, t_{2x}, t_{2y}, \lambda$ and $\theta$. We first compute the partial derivatives of $L$:

$$\begin{aligned}\frac{\partial L}{\partial t_{1x}} = -\frac{1}{4}\Big[2i_{1x} \\ -\frac{\exp(i_{1x}t_{1x} + i_{1y}t_{1y})i_{1x}}{A} \\ -\frac{\exp(i_{2x}t_{1x} + i_{2y}t_{1y})i_{2x}}{B} \\ -\frac{\exp(i_{1x}t_{1x} + i_{1y}t_{1y})i_{1x} + \exp(i_{2x}t_{1x} + i_{2y}t_{1y})i_{2x}}{C}\Big]\end{aligned}$$

$$\begin{aligned}\frac{\partial L}{\partial t_{1y}} = -\frac{1}{4}\Big[2i_{1y} \\ -\frac{\exp(i_{1x}t_{1x} + i_{1y}t_{1y})i_{1y}}{A} \\ -\frac{\exp(i_{2x}t_{1x} + i_{2y}t_{1y})i_{2y}}{B} \\ -\frac{\exp(i_{1x}t_{1x} + i_{1y}t_{1y})i_{1y} + \exp(i_{2x}t_{1x} + i_{2y}t_{1y})i_{2y}}{C}\Big]\end{aligned}$$

$$\begin{aligned}\frac{\partial L}{\partial t_{2x}} = -\frac{1}{4}\Big[2i_{2x} \\ -\frac{\exp(i_{1x}t_{2x} + i_{1y}t_{2y})i_{1x}}{A} \\ -\frac{\exp(i_{2x}t_{2x} + i_{2y}t_{2y})i_{2x}}{B} \\ -\frac{\exp(i_{1x}t_{2x} + i_{1y}t_{2y})i_{1x} + \exp(i_{2x}t_{2x} + i_{2y}t_{2y})i_{2x}}{D}\Big]\end{aligned}$$

$$\frac{\partial L}{\partial t_{2y}} = -\frac{1}{4}\Big[2i_{2y} - \frac{\exp(i_{1x}t_{2x} + i_{1y}t_{2y})i_{1y}}{A} - \frac{\exp(i_{2x}t_{2x} + i_{2y}t_{2y})i_{2y}}{B} - \frac{\exp(i_{1x}t_{2x} + i_{1y}t_{2y})i_{1y} + \exp(i_{2x}t_{2x} + i_{2y}t_{2y})i_{2y}}{D}\Big]$$

The partial derivatives of $O$ using the above computed partial derivatives of $L$ are:

$$\begin{aligned}
\frac{\partial O}{\partial \lambda} &= 1 - t_{1x}^2 - t_{1y}^2 \\
\frac{\partial O}{\partial \lambda} &= 1 - t_{2x}^2 - t_{2y}^2 \\
\frac{\partial O}{\partial t_{1x}} &= \frac{\partial L}{\partial t_{1x}} - 2\lambda t_{1x} \\
\frac{\partial O}{\partial t_{1y}} &= \frac{\partial L}{\partial t_{1y}} - 2\lambda t_{1y} \\
\frac{\partial O}{\partial t_{2x}} &= \frac{\partial L}{\partial t_{2x}} - 2\theta t_{2x} \\
\frac{\partial O}{\partial t_{2y}} &= \frac{\partial L}{\partial t_{2y}} - 2\theta t_{2y}
\end{aligned}$$

We now have 6 variables and 6 equations, and can therefore find the optimal solution(s) for $T_1$ and $T_2$. We use scipy's `optimize.fsolve` function for finding these solutions programmatically.

# Appendix B

# Simulation Experiment details and Extended Results

## B.1 Experimental Details

In Table B.1, we provide ranges of the different values of each factor that we run simulations for in Section 4.4. We run simulations with all possible combinations of the different factors listed in the table leading to a total of 2500 different simulation runs.

| Factor | Controlled by | Range |
|---|---|---|
| *Batch Size* | $N$ | $\{256\}$ |
| *Dimension* | $d$ | $\{2, 10, 25, 100, 256\}$ |
| *Temperature* | $\tau$ | $\{0.01, 0.04, 0.1, 0.25, 1.0\}$ |
| *Mismatch Ratio* | $M$ | $\{0, 25, 50, 75, 90\}$ |
| *Alignment* | $\theta$ | $\{0, 30, 60, 90, 180\}$ |
| *Uniformity* | $\kappa$ | $\{1, 10, 100, 1000\}$ |

Table B.1 Settings of the different factors for the simulation experiment in Section 4.4

## B.2 Extended Results

In this section, we showcase further plots corroborating the main results of Sections 4.3, 4.4 and 4.5.

| Mismatch Ratio ($M$) | Alignment ($\theta$) | Concentration ($\kappa$) | Expected Loss |
|---|---|---|---|
| 0 | 90° | 1 | 1.279 |
| 0 | 0° | 1 | 1.280 |
| 0 | 180° | 1 | 1.287 |
| 0 | 30° | 1 | 1.287 |
| 0 | 60° | 1 | 1.288 |
| 0 | 90° | 10 | 1.296 |
| 0 | 60° | 10 | 1.296 |
| 0 | 0° | 10 | 1.298 |
| 0 | 30° | 10 | 1.300 |
| 0 | 180° | 10 | 1.307 |

Table B.2 Lowest 10 losses from the training simulation experiment (Section 4.4.1) and their corresponding settings for $\tau = 0.01$

| Mismatch Ratio ($M$) | Alignment ($\theta$) | Concentration ($\kappa$) | Expected Loss |
|---|---|---|---|
| 0 | 180° | 1 | 2.442 |
| 0 | 30° | 1 | 2.444 |
| 0 | 60° | 1 | 2.446 |
| 0 | 90° | 1 | 2.447 |
| 0 | 0° | 1 | 2.448 |
| 0 | 180° | 10 | 2.451 |
| 0 | 60° | 10 | 2.453 |
| 0 | 30° | 10 | 2.456 |
| 0 | 0° | 10 | 2.456 |
| 0 | 90° | 10 | 2.456 |

Table B.3 Lowest 10 losses from the training simulation experiment (Section 4.4.1) and their corresponding settings for $\tau = 0.04$

| Mismatch Ratio ($M$) | Alignment ($\theta$) | Concentration ($\kappa$) | Expected Loss |
|---|---|---|---|
| 0 | 60° | 1 | 3.996 |
| 0 | 180° | 1 | 3.997 |
| 0 | 30° | 1 | 3.997 |
| 0 | 90° | 1 | 3.997 |
| 0 | 0° | 1 | 3.998 |
| 0 | 0° | 10 | 3.998 |
| 0 | 180° | 10 | 3.999 |
| 0 | 90° | 10 | 3.999 |
| 0 | 60° | 10 | 3.999 |
| 0 | 30° | 10 | 4.001 |

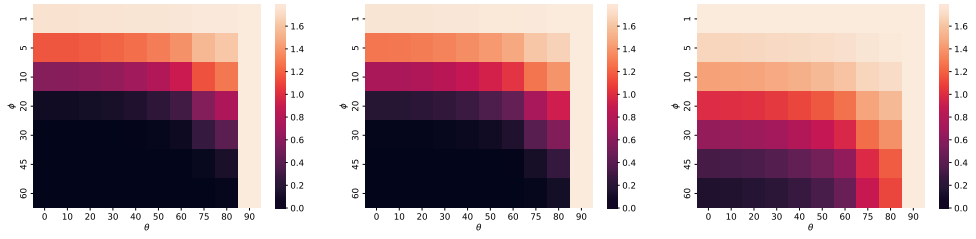
Table B.4 Lowest 10 losses from the training simulation experiment (Section 4.4.1) and their corresponding settings for $\tau = 0.1$

| **Mismatch Ratio** ($M$) | **Alignment** ($\theta$) | **Concentration** ($\kappa$) | **Expected Loss** |
|---|---|---|---|
| 0 | 60° | 1 | 4.876 |
| 0 | 90° | 1 | 4.876 |
| 0 | 0° | 1 | 4.877 |
| 0 | 30° | 1 | 4.877 |
| 0 | 180° | 1 | 4.878 |
| 0 | 0° | 10 | 4.878 |
| 0 | 180° | 10 | 4.879 |
| 0 | 90° | 10 | 4.879 |
| 0 | 30° | 10 | 4.879 |
| 0 | 60° | 10 | 4.880 |

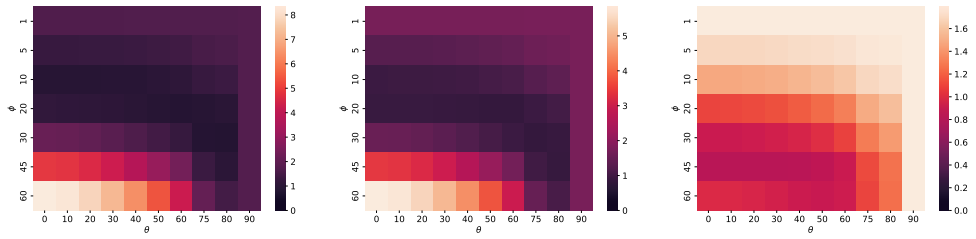
Table B.5 Lowest 10 losses from the training simulation experiment (Section 4.4.1) and their corresponding settings for $\tau = 0.25$

| **Mismatch Ratio** ($M$) | **Alignment** ($\theta$) | **Concentration** ($\kappa$) | **Expected Loss** |
|---|---|---|---|
| 0 | 90° | 1 | 5.372 |
| 0 | 30° | 1 | 5.372 |
| 0 | 180° | 1 | 5.372 |
| 0 | 0° | 1 | 5.372 |
| 0 | 60° | 1 | 5.372 |
| 0 | 60° | 10 | 5.372 |
| 0 | 0° | 10 | 5.372 |
| 0 | 30° | 10 | 5.372 |
| 0 | 180° | 10 | 5.373 |
| 0 | 90° | 10 | 5.373 |

Table B.6 Lowest 10 losses from the training simulation experiment (Section 4.4.1) and their corresponding settings for $\tau = 1.0$

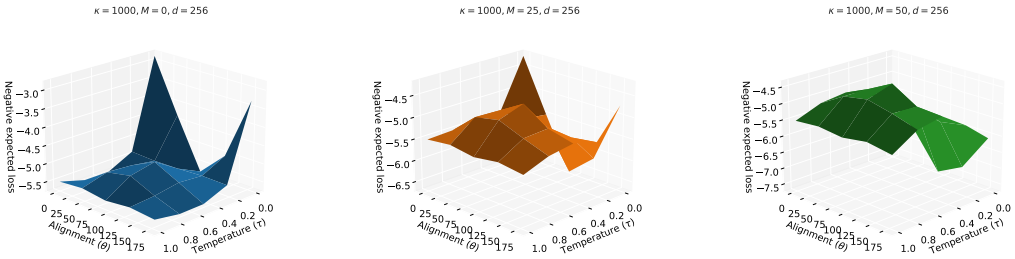


(a) $\tau = \frac{1}{50}$, no mismatches (b) $\tau = \frac{1}{35}$, no mismatches (c) $\tau = \frac{1}{5}$, no mismatches

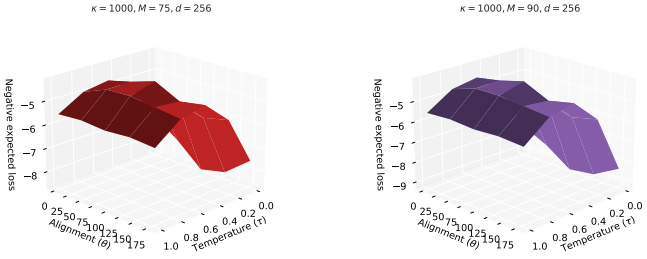


(d) $\tau = \frac{1}{50}$, with mismatches (e) $\tau = \frac{1}{35}$, with mismatches (f) $\tau = \frac{1}{5}$, with mismatches

Fig. B.1 Further loss heatmaps depicting $\phi$ v/s $\theta$ with and without mismatches at $\tau = \frac{1}{5}$, $\tau = \frac{1}{35}$ and $\frac{1}{50}$. These plots extend the results of Section 4.3. Darker cells denote smaller loss values and lighter cells denote larger loss values. Therefore, darker is better in these plots.

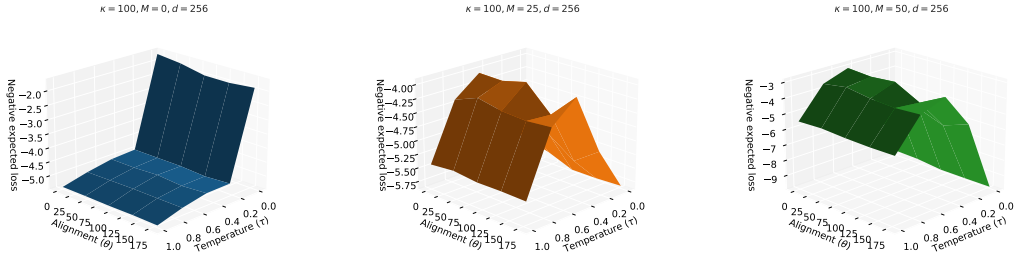



(a) $M = 0$ (b) $M = 25$ (c) $M = 50$

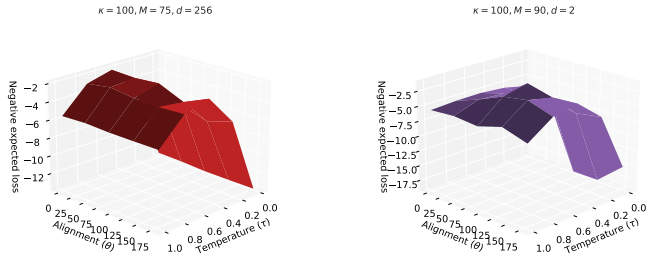



(d) $M = 75$ (e) $M = 90$

Fig. B.2 Loss landscape at low *Uniformity* ($\kappa = 1000$)

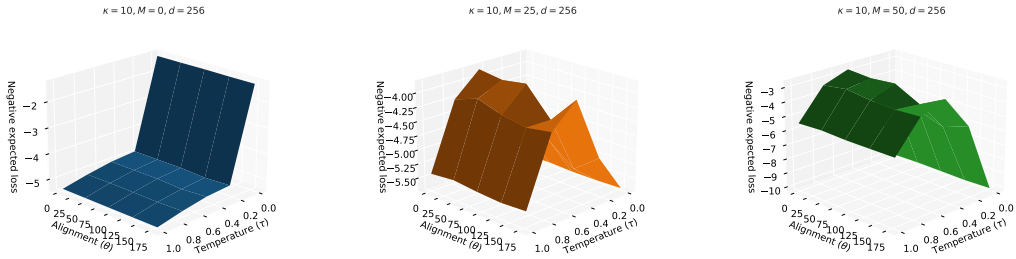



(a) $M = 0$ (b) $M = 25$ (c) $M = 50$

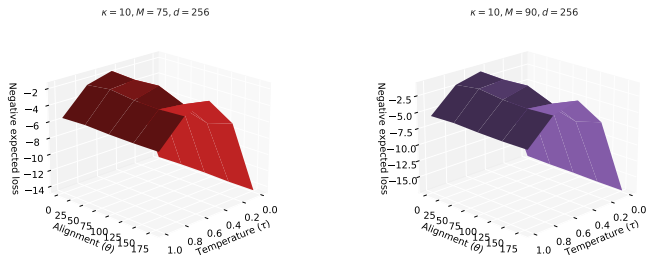



(d) $M = 75$ (e) $M = 90$

Fig. B.3 Loss landscape at moderately low *Uniformity* ($\kappa = 100$)

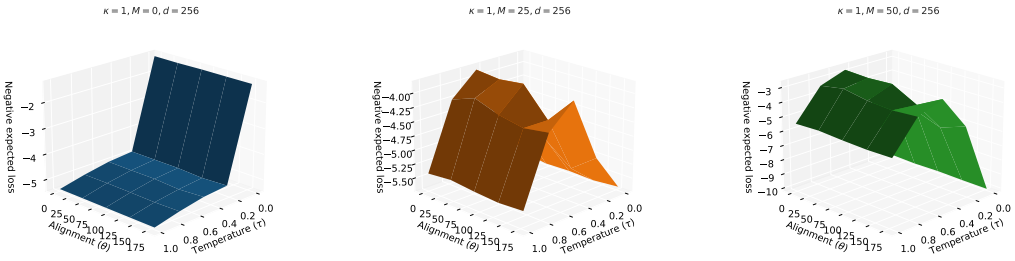



(a) $M = 0$ (b) $M = 25$ (c) $M = 50$

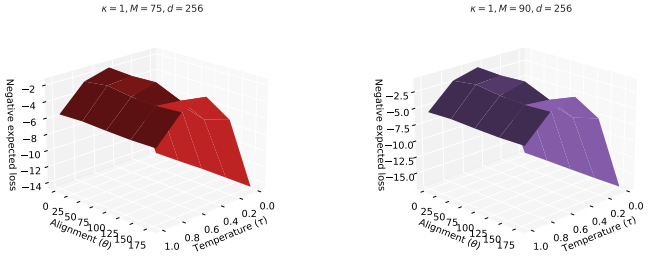



(d) $M = 75$ (e) $M = 90$

Fig. B.4 Loss landscape at moderately high *Uniformity* ($\kappa = 10$)

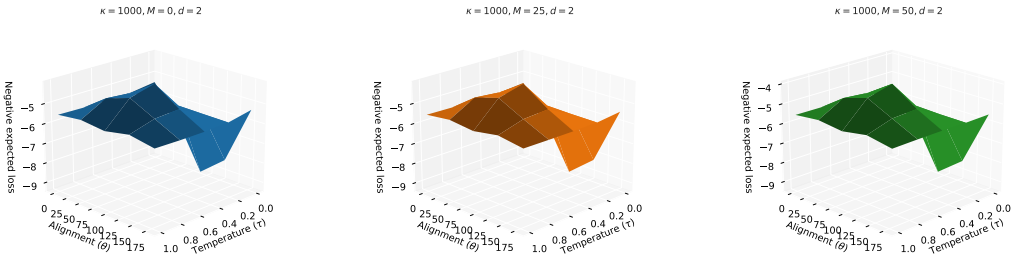



(a) $M = 0$ (b) $M = 25$ (c) $M = 50$



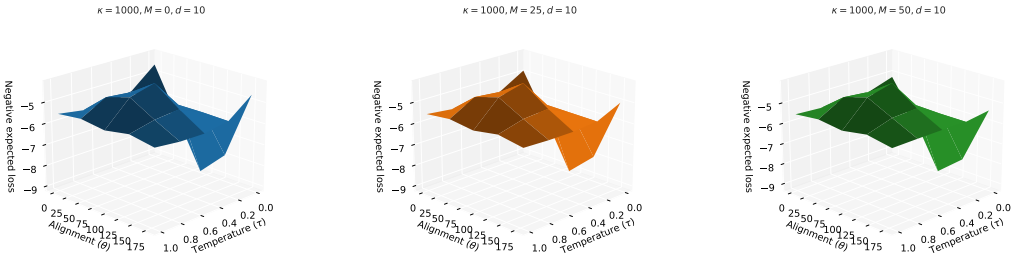

(d) $M = 75$ (e) $M = 90$

Fig. B.5 Loss landscape at high *Uniformity* ($\kappa = 1$)



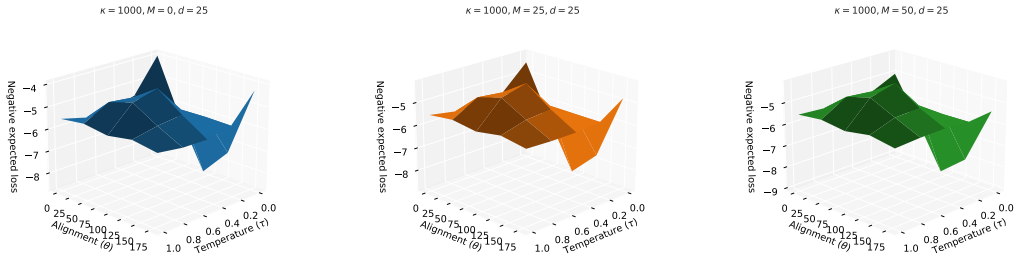

(a) $M = 0$ (b) $M = 25$ (c) $M = 50$

Fig. B.6 Loss landscape at low *Uniformity* ($\kappa = 1000$) for $d = 2$



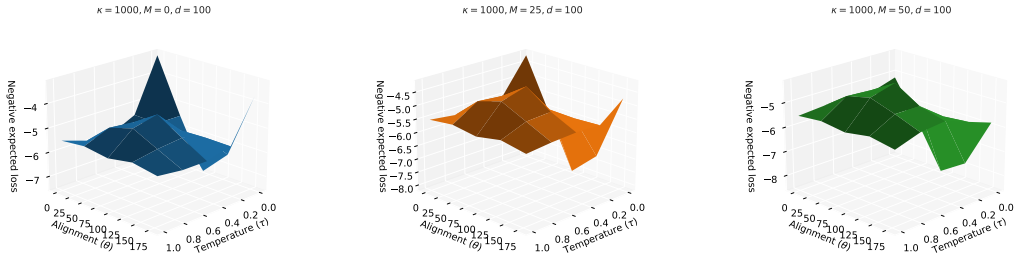

(a) $M = 0$ (b) $M = 25$ (c) $M = 50$

Fig. B.7 Loss landscape at low *Uniformity* ($\kappa = 1000$) for $d = 10$



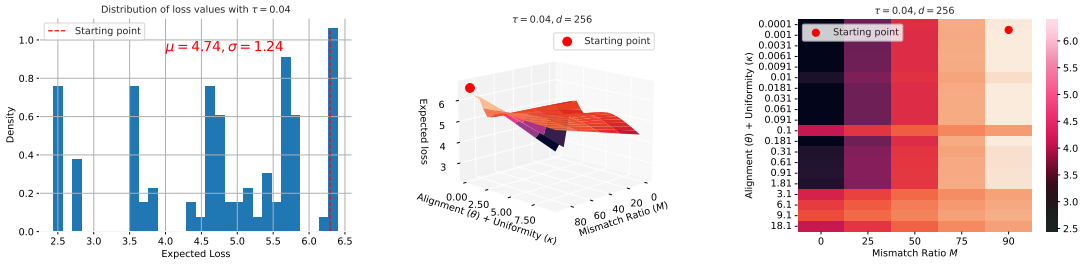

(a) $M = 0$ (b) $M = 25$ (c) $M = 50$

Fig. B.8 Loss landscape at low *Uniformity* ($\kappa = 1000$) for $d = 25$



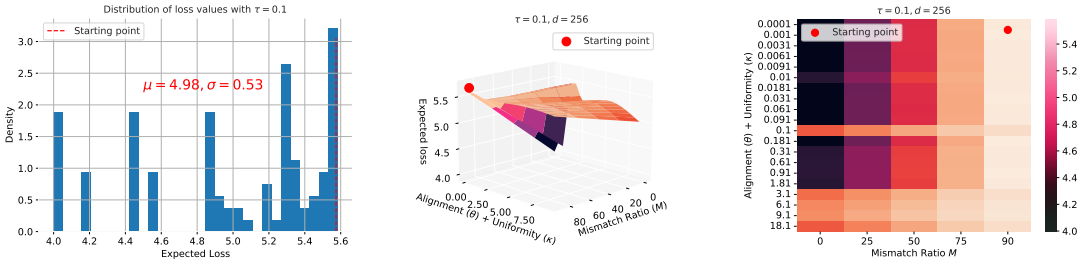

(a) $M = 0$ (b) $M = 25$ (c) $M = 50$

Fig. B.9 Loss landscape at low *Uniformity* ($\kappa = 1000$) for $d = 100$



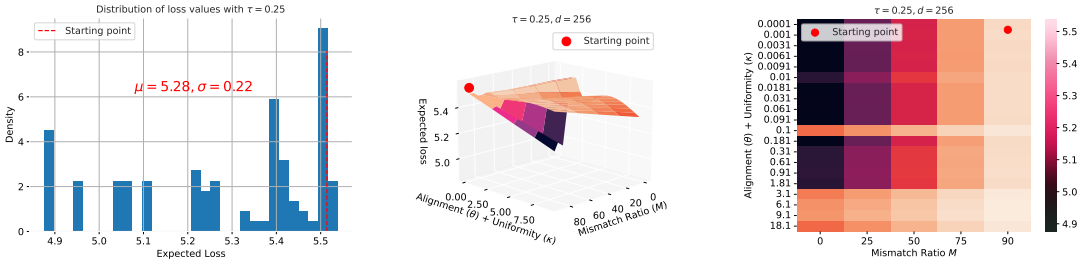

(a) Loss distribution (b) Loss landscape (Azimuth view) (c) Loss landscape (Top view)

Fig. B.10 Expected loss dynamics at $\tau = 0.04$




(a) Loss distribution (b) Loss landscape (Azimuth view) (c) Loss landscape (Top view)

Fig. B.11 Expected loss dynamics at $\tau = 0.1$




(a) Loss distribution (b) Loss landscape (Azimuth view) (c) Loss landscape (Top view)

Fig. B.12 Expected loss dynamics at $\tau = 0.25$