Aligning Vision and Language: Harnessing Language Semantics for Efficient Vision Models

Mayug Maniparambil, B.Tech, M.Tech.

Prof. Noel E. O'Connor and

Dr. Kevin McGuinness

A Dissertation submitted in fulfilment of the requirements for the award of Doctor of Philosophy (Ph.D.)

School of Electronic Engineering

Dublin City University

April 2025

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy is entirely my own work, and that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed : Mayug Maniparambil **Student number:** 20213326

Date : 10th of April 2025

Acknowledgements

To begin, I want to express my deepest gratitude to my supervisors, Dr. Kevin McGuinness and Prof. Noel O'Connor, for their invaluable guidance and unwavering support throughout this journey. Kevin, your patience and insight in unraveling the complexities of machine learning have been nothing short of essential. Your mentorship has profoundly shaped my growth, and the wisdom you've shared will stay with me for years to come. Noel, I am truly grateful for your patience, insightful research discussions, help with writing, and steadfast encouragement throughout this journey. Without your support, this thesis would not have been possible. Beginning my PhD amid the COVID-19 pandemic brought a unique set of challenges, especially as I found myself stuck at home, unable to travel to Dublin for close to a year. Despite these obstacles, I focused on building a productive remote work routine, leveraging virtual meetings, and maintaining consistent communication with my supervisors, which helped me overcome these challenges and stay on track. Your consistent support—through weekly Zoom calls and your always-open door—made a world of difference.

I also want to thank my closest collaborators - Raiymbek Akshulakov, Yasser Dahou, Chris Vorster and Sidra Aleem for the stimulating discussions that sparked numerous interesting ideas and projects. My heartfelt thanks go to Sanath Narayan and other collaborators at TII and UC Berkeley for our early morning meetings on Thursdays, which provided crucial feedback and helped refine the direction of my research in the final year of my PhD.

To my parents and my sister, thank you for your constant support and encouragement. To my girlfriend Leysi, thank you for your love, patience, and unwavering belief in me.

I would like to dedicate this thesis to the memory of Dr. Kevin McGuinness. His contributions will never be forgotten, and this work is a testament to his enduring influence and the profound impact he has had on all of us.

Contents

List of Figures

List of Tables

Abstract

Aligning Vision and Language: Harnessing Language Semantics for Efficient Vision Models

Mayug Maniparambil

This thesis explores methods to enhance the efficiency, flexibility, and generalization of vision models by leveraging semantic information from language. Inspired by human multi-modal perception, the research conducted addresses key limitations in current models: high data and compute demands, limited generalization to new concepts, and suboptimal unimodal features. The thesis begins by exploring how structured semantic information, such as domain-expert knowledge, can enhance the few-shot learning of visual concepts. A novel algorithm, BaseTransformers, is proposed to integrate semantic information, enabling computer vision models to learn new concepts with minimal labeled data by associating them with semantically similar and well-represented base concepts. Extensive evaluations on benchmark datasets highlight improvements over few-shot vision models that do not leverage semantic information. Recognizing the scalability challenges of curated semantics, this thesis introduces a strategy to leverage large language models (LLMs) as a scalable source of semantic knowledge. The proposed VDT-Adapter learns to dynamically select and aggregate LLM-generated semantic information, supporting zero-shot and few-shot domain transfer of CLIP models, as validated through evaluations on 12 benchmark datasets. The research further identifies challenges faced by CLIP models regarding compute and data requirements for pretraining, as well as issues with flexibility and generalization due to suboptimal unimodal features in the joint embedding space. To address these challenges, recent advancements in unimodal vision and language encoders are leveraged, with an analysis of these models conducted through the perspective of representational similarity. Motivated by the hypothesis that vision and language encoders model the same physical reality, this thesis studies their semantic similarity, revealing that their representations often share a high degree of alignment, comparable to that of aligned vision-language encoders. Building on this insight, a lightweight framework that aligns pre-trained, strong unimodal encoders using simple projection transformations is developed. This approach is significantly more compute/data efficient while outperforming CLIP on 0-shot domain transfer to classification/retrieval tasks. Furthermore, the framework's flexibility and generalization across diverse tasks like multi-lingual retrieval/classification, 0-shot localization, and long-context retrieval are demonstrated. The findings pave the way for flexible, efficient, and generalizable solutions for open-world understanding, contributing to broader applications of multi-modal systems. Finally, the conclusion of this thesis summarizes the contributions and future research directions

Publications

Publications arising directly from this thesis

- Maniparambil, M., Akshulakov, R., Djilali, Y. A. D., Narayan, S., Singh, A., & O'Connor, N. E. (2024). From Unimodal to Multimodal: Scaling up Projectors to Align Modalities. arXiv preprint arXiv:2409.19425 (accepted IEEE/CVF Conference on Computer Vision and Pattern Recognition 2025)

- Maniparambil, M., Akshulakov, R., Djilali, Y. A. D., El Amine Seddik, M., Narayan, S., Mangalam, K., & O'Connor, N. E. (2024). Do Vision and Language Encoders Represent the World Similarly? In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2024.

- Maniparambil, M., Vorster, C., Molloy, D., Murphy, N., McGuinness, K., & O'Connor, N. E. (2023). Enhancing CLIP with GPT-4: Harnessing Visual Descriptions as Prompts. In Proceedings of International Conference on Computer Vision 2023

- Maniparambil, M., McGuinness, K., & O'Connor, N. E. Base Transformers: Attention over base data-points for One Shot Learning. (2022) In Proceedings of British Machine Vision Conference 2022.

Other publications by the author

- Sirotkin, K., Escudero-Viñolo, M., Carballeira, P., Maniparambil, M., Barata, C., & O'Connor, N. (2024). Pinpoint Counterfactuals: Reducing social bias in foundation models via localized counterfactual generation. (under review CVPR 2025).

- Aleem, S., Wang, F., Maniparambil, M., Arazo, E., Dietlmeier, J., Curran, K., O'Connor, N. E., & Little, S. (2024). Test-Time Adaptation with SaLIP: A Cascade of SAM and CLIP for Zero-shot Medical Image Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 5184–5193).

- Aleem, S., Maniparambil, M., Little, S., O'Connor, N., & McGuinness, K. (2023). An Ensemble Deep Learning Approach for COVID-19 Severity Prediction Using Chest CT Scans. Irish Machine Vision and Image Processing Conference 2023.

- Boulogne, L. H., Lorenz, J., Kienzle, D., Schon, R., Ludwig, K., Lienhart, R., Jegou, S., Li, G., Chen, C., Wang, Q., Shi, D., Maniparambil, M., Muller,

D., Mertes, S., Schroter, N., Hellmann, F., Elia, M., Dirks, I., Bossa, M. N., Berenguer, A. D., Mukherjee, T., Vandemeulebroucke, J., Sahli, H., Deligiannis, N., Gonidakis, P., Huynh, N. D., Razzak, I., Bouadjenek, R., Verdicchio, M., Borrelli, P., Aiello, M., Meakin, J. A., Lemm, A., Russ, C., Ionasec, R., Paragios, N., van Ginneken, B., Revel Dubois, M. P. (2023). The STOIC2021 COVID-19 AI challenge: applying reusable training methodologies to private data. arXiv preprint arXiv:2306.10484. Medical Image Analysis 2024

Chapter 1

Introduction

1.1 Motivation

Humans are innately multi-modal beings. Our perception of the world is not confined to a singular sensory channel; rather, we rely on an intricate fusion of sights, sounds, smells, and touch to build a rich mental model of our environment and the physical world. Vision and language, in particular, play pivotal roles in this multi-faceted perception, as the former is our primary way of gathering information about our environment while the latter is our main way to articulate our thoughts and understanding of the world and communicate it to other humans. Some primary characteristics of human multi-modal perception are outlined in the following.

Integration of information: The human brain integrates information from multiple modalities using a process called re-entry which describes the ongoing, recursive signalling among different brain regions [42, 43]. This refers to the idea that the sensation of one modality can invoke the sensation of another [165, 185] and that the human brain has similar high-level semantic representations for the same concepts perceived through different modalities [35, 16].

Sensory Compensation / Degeneracy: When one sense is impaired, other senses can become more acute, demonstrating the brain's flexibility and adaptability in sensory processing [182, 105]. The human brain's ability to reorganize itself by forming new neural connections throughout life, known as neuroplasticity, plays a crucial role in this process [105, 159]. This reorganization is particularly pronounced in individuals who experience sensory deprivation from birth or early in life, leading to the takeover of the deprived cortex by other sensory modalities [131]. In short, humans can make use of other modalities to construct their mental model of the physical world when a modality is unavailable.

Cross-Modal Interaction: Sensory modalities often interact with each other [157, 169, 40, 119]. For instance, the McGurk effect demonstrates how visual information

(lip movements) can alter what we hear [103]. These interactions show that sensory processing is not isolated but interdependent.

The research reported in this thesis is underpinned by this understanding that vision and language are the primary modalities with which humans understand and communicate with their environment and that our perception and interaction of their representations form a major part of our intelligence. Hence, our quest for a human-like thinking machine or an Artificial General Intelligence has driven research communities to focus on Computer Vision (CV) and Natural Language Processing (NLP) to develop efficient and generalizable methods of modeling vision and language. Historically, the computer vision and natural language processing research communities evolved independently. However, the effectiveness of multi-modal perception in humans has inspired researchers to develop machine learning models that utilize cross-modal dependencies and fusion to tackle diverse tasks. Inspired by the three primary characteristics of human multi-modal perception, we are interested in the general research question that semantic information from language represents the same physical world as visual information and is complementary to the visual information in modeling the physical world. More specifically, we are interested in: 1. integrating this semantic knowledge into computer vision models to efficiently learn novel concepts and enhance performance in label-constrained scenarios; and 2. investigating how similar vision and language representations are, given that they model the same physical world, and exploring how this similarity can be exploited to develop more flexible and efficient computer vision models capable of performing well on any conceivable concept (also known as open world understanding).

Computer vision models have achieved remarkable progress in solving closed-world tasks, where the set of visual concepts is predefined and annotated datasets are readily available. This progress has been driven by supervised learning methods that rely heavily on large-scale annotated datasets. However, the reliance on annotated data introduces significant limitations. Annotating datasets is costly, time-consuming, and often impractical for specialized or evolving domains. While self-supervised learning (SSL) methods have reduced the dependency on labels for pretraining, these methods still require annotated data for fine-tuning and transfer learning to specific downstream tasks.

To address the challenges of limited labeled data, researchers have turned to few-shot learning, which aims to generalize to new tasks with only a few examples per concept. Despite progress, few-shot learning has predominantly focused on leveraging visual information alone, while the use of semantics, such as meanings, relationships, and structured representations of the world has remained underexplored. Semantics from domain experts offer a robust representation of the world that can complement visual data, particularly in low-data scenarios. Incorporat-

ing semantics into few-shot learning could significantly enhance generalization by providing a richer context for understanding novel concepts.

Given the abundance of free but noisy image-caption pairs available on the internet, the focus of few-shot and zero-shot learning research has recently shifted to pre-training vision-language models using language as an API to adapt to novel concepts. Recent research has leveraged vast amounts of image-caption pairs from the internet to extend vision models from closed-world to open-world settings. This approach, exemplified by models like CLIP [132] uses a contrastive loss to effectively align visual and textual features, with both the visual and textual encoders initialized from scratch. This enables zero-shot domain transfer to classification tasks by constructing the classification prototypes using natural language prompts – marking a significant step forward in moving away from closed-world assumptions towards a more open-world approach. However, despite its promise, CLIP suffers from several key limitations that hinder its utility across a range of applications.

While CLIP is easy to adapt to new datasets using natural language description of the classes, challenges similar to the original few-shot and zero-shot research persist, primarily because the base data (known as the pretraining dataset) cannot encompass all possible concepts [176] due to Zipf's law. The 'long-tail' phenomenon or Zipf's Law [218] refers to the fact that many types of data studied in the world can be approximated with a Zipfian distribution which is characterized by a decreasing number of examples available for a large number of real-world concepts. Due to the 'long-tail' effect, it becomes nearly impossible to ensure that data corresponding to every conceivable concept is present in the pre-training dataset leading to a domain gap between the train data and the downstream datasets. This leads to CLIP's performance on fine-grained datasets (e.g., specific species in bird classification) being suboptimal raising the need for research into suitable adaptation of CLIP models to downstream tasks where labeled data is limited or not available. In this thesis, we also explore how **semantic information** can be used to enable this adaptation of CLIP models in a label-efficient manner but without the use of curated semantic data, so that the overall approach is scalable to novel concepts.

Another potential challenge that CLIP models face is that of compute and data requirements. Training CLIP models is resource-intensive, requiring vast datasets and substantial computational resources. For instance, the original CLIP model from OpenAI [132] was trained on 400 million image, caption pairs, requiring over 20k GPU hours. This limitation has spurred interest in lightweight training methods, such as LiT (Locked-image Tuning) [207] to make vision-language models more accessible, where strong unimodal encoders are used to initialize the vision encoder of a CLIP model and kept fixed in contrast to training from scratch, potentially halving the compute requirements. Similarly [71], studied locking both the image

and text encoders, but the limited scale of their training datasets and random choice of encoder pairs limited the transferability of the framework to downstream tasks.

It has also been shown that CLIP models have suboptimal unimodal features that eventually lead to suboptimal cross-modal generalization in scenarios that are different from CLIP's pre-training setup [171, 203, 69, 86]. For instance, CLIP's vision encoder struggles with localization tasks, such as identifying specific objects within complex scenes [86]. This limitation has motivated the development of visual representation learning methods that improve localization, such as SPARC loss [12] instead of the contrastive loss, which enhances attention to spatial details. Similarly, the sub-optimal text encoder features limit CLIP's applicability to novel scenarios. As applications expand to real-world use cases, CLIP's text encoder has shown limitations in adapting to multilingual [19] [180] and long-context scenarios [177]. Efforts like multilingual CLIP [19] and adaptations for long-context understanding [177] aim to overcome these barriers. Both these issues are currently solved by re-training a CLIP model from scratch which is compute/data-intensive, limiting their accessibility while completely ignoring the strong advances in unimodal performances in both vision and language domains. For instance, DinoV2 [121] models have demonstrated strong local and global features by incorporating both iBOT[214] and DINO [17] losses during pre-training on a large vision dataset. Similarly, multiple Sentence-Transformers [139] models have improved performances on several sentence tasks including multi-lingual and long context abilities on the MTEB benchmark [111]. All of this begs the question: Why not utilize robust unimodal encoders to create multimodal dual-encoders like CLIP? This approach would address the compute/data efficiency challenge by potentially training only 1 % of the parameters, allowing for rapid iterations. It would also leverage strong uni-modal text features to enhance flexibility. Hence we propose examining the representation spaces of uni-modal encoders to assess their similarity and then using these insights to develop a framework for aligning these encoders.

To summarize, in this thesis, we aim to address challenges of data/compute efficiency, flexibility, and generalization of vision and vision language models by leveraging semantics from language and examining the semantic representations they induce. Specifically, we focus on:

- Few-Shot Learning with Semantics: We investigate how semantic information from domain experts—structured representations of the world—can improve few-shot learning by providing robust, context-rich guidance, particularly in scenarios with limited labeled data.

- Enhancing CLIP for Fine-Grained Tasks: We explore methods to integrate semantics into CLIP's architecture to improve its performance on fine-grained

datasets, specifically how LLM-generated semantics could be used to adapt CLIP to novel domains in a scalable manner.

- Semantic Similarity of Unimodal Embedding Spaces. To address the challenges with the CLIP model detailed above we propose to develop a framework where strong unimodal encoders are aligned using projection transformations. To study the feasibility of this we investigate the similarity of unimodal vision and language representations

- Improving the Compute/Data Efficiency and SubOptimal Unimodal features / Flexibility Challenges in CLIP Models: To address the resource-intensive nature of CLIP training, we develop a framework that aligns pre-trained vision and language encoders through lightweight transformations, reducing the need for large-scale joint training. This also indirectly addresses the CLIP model's lack of flexibility to adapt to novel tasks by preserving the strong unimodal text or vision features in the joint-embedding space.

By systematically addressing these challenges, this thesis advances the understanding of vision and language encoders and, drawing from these insights enhances the capabilities of vision-language models, making them more efficient, flexible, and generalizable across a wide range of tasks. The following hypotheses and research questions outline our approach to achieving these goals.

1.2 Hypothesis and Research Questions

- **Hypothesis 1 (H1):** Several few-shot learning algorithms [161, 48] have explored the use of a large base dataset of images to facilitate computer vision models to learn new concepts with few examples. Semantic information in the form of domain expert data like wordnet graphs or class-level descriptions of bird body parts can often be considered as robust representations of the world, and can provide relationship information between the base and novel concepts which can facilitate label efficient learning of computer vision algorithms. However, this is under-explored and this leads to our first hypothesis. **Semantic information about the world is suited for improving generalizability and sample efficiency of vision models, because they are a more robust representation of the world that can guide learning of visual representations when the visual data is limited.**

  - The above hypothesis leads to our first research question **(RQ1)** : *How can we incorporate semantic information to improve the performance of few-shot learning? More specifically, can we use base to novel concept*

relationships from domain expert annotated semantic information to facilitate learning novel concepts with few examples?

- **Hypothesis 2 (H2):** Our work [101] and several others [154, 115, 212] have shown that semantic information about the world from domain experts can be used to structure the learning of vision algorithms under label constraints but semantics from domain experts is costly to scale to novel concepts. Recent advances in Large Language Models (LLMs) have shown that LLMs consume almost all text information on the internet during training and can serve as a proxy domain expert for semantics making it easily scalable to novel concepts. However, the use of semantic information from LLMs to improve vision algorithms is under-explored and this leads to our second hypothesis. **LLMs have a good world model and text generated by LLMs can be a good proxy for extracting semantic information about the world that can be used to guide the generalizability and sample efficiency of CLIP models.** We explore this hypothesis in the task of improving 0-shot and few-shot domain transfer of vision language foundational models like CLIP to novel fine-grained datasets. This hypothesis gives rise to the following research questions.

  - **RQ2** *How can LLMs be used to generate semantic information useful for computer vision models in a scalable manner?*
  - **RQ3** *How can LLM generated semantic information be used to the improve 0-shot and few-shot domain transfer performance of vision language foundation models like CLIP?*

- **Hypothesis 3 (H3):** Recent work [104, 102] has demonstrated that CLIP models, despite their large-scale vision-language pre-training, can leverage semantic information generated by LLMs to enhance their zero-shot and few-shot domain transfer performance. However, incorporating LLM-generated data directly into CLIP's input space is a suboptimal method of providing CLIP's text encoder with novel context because it explains novel concepts only in terms of base concepts that the text encoder already understands. This limitation arises due to the text-encoder's limited concept coverage as it is trained on web-scale image-caption pairs, where the caption data is often noisy and limited in coverage of concepts. By contrast, language encoders [139] trained on much larger text-only datasets exhibit superior performance in concept coverage, compositionality, multilingual capabilities, and long-context reasoning [111]. Additionally, CLIP's vision encoder has poor localization capabilities

[86] due to its global joint training strategy, which matches pooled image representations to text representations. Another challenge of CLIP's framework is the computational and data requirements for from-scratch joint training. These limitations could potentially be addressed by a framework that aligns frozen unimodal encoders. To develop such a framework, we begin by investigating the similarity between vision and language representations, grounded in the hypothesis that both encoders model the same physical reality [66]. If the encoders are effective at capturing this reality, their representations should exhibit meaningful similarities. This hypothesis provides the foundation for our next exploration.

Given that vision and language are trying to model the same physical reality, the representations from vision and language encoders should demonstrate high similarity in terms of semantics. This leads to the following research questions

  - **RQ4** *How similar are vision and language representations given that their encoders are trying to model the same physical reality?*
  - **RQ5** *Is there a way to connect semantically similar vision and language representations in a training-free manner?*

- **Hypothesis 4 (H4):** Recent work including ours [100, 66] have shown that well-trained vision and language embeddings exhibit high similarity in their representation spaces. A framework where frozen vision and language encoders can be aligned could result in flexibility due to, stronger multi-modal models via strong unimodal embeddings and compute/sample efficiencies due to a low number of trainable parameters. This leads to our next hypothesis. **Given that well-trained vision and language encoders have highly similar representation spaces, simple projection transformations might be sufficient to bridge them.** This leads to our final set of research questions.

  - **RQ6** *Can semantically similar unimodal representations be bridged using simple projection transformations?*
  - **RQ7** *How can we scale up training of simple projection transformations to align unimodal encoders and achieve performant, flexible and compute/data efficient CLIP models?*

1.3 Thesis Structure

The remainder of this thesis is structured as follows. Chapter 2 provides the necessary technical background about vision-language models, 0-shot learning, few-shot

learning, and representational similarity measures to understand the research presented in this thesis, as well as a high-level overview of the related work.

Chapter 3, investigates **RQ1**, and presents a novel method for using semantic information from various sources such as WordNet-graphs or domain experts to improve few-shot learning in computer vision. We find that semantic information from curated sources is robust enough to be used to relate novel concepts to base concepts and hence improve the learning of novel concepts in a few-shot manner. We achieve this by using a cross-attention mechanism that attends to the most semantically similar regions of the well-trained base feature space and improves the novel support instance prototypes resulting in improvements compared to the SOTA in on-shot learning across various settings. This chapter concludes by demonstrating that curated semantic information can indeed help improve the generalizability and data efficiency of computer vision models.

Chapter 4 investigates a limitation of the curated semantic information used in Chapter 3, which is that to scale to novel concepts requires costly human expert annotations. However, LLMs have consumed most textual data on the internet and can act as efficient semantic information retrieval engines motivating us to explore the use of LLMs to generate semantic information that could be used to improve the 0-shot and few-shot domain transfer performances of vision language foundational models thereby addressing **RQ2** and **RQ3**. The Chapter concludes by demonstrating that LLM-generated semantics are scalable and can be used to improve the generalizability of CLIP to fine-grained datasets in a label-efficient manner.

Chapter 5 focuses on the first step towards developing a framework that would address several of the limitations of the CLIP framework that we observe in Chapter 4 like training compute, limited flexibility, and weak unimodal features. A framework that aligns powerful unimodal vision and language encoders would solve these but requires us to understand how similar the representation spaces are of the said encoders in the first place. This motivates us to investigate **RQ4** and **RQ5**. We conclude this Chapter by establishing that well-trained vision and language encoders have highly similar semantic structures and that they can be connected in a training-free manner by graph-matching based optimization techniques at test time.

In Chapter 6, we investigate **RQ6** and **RQ7** by studying whether there is a correlation between Ease of Alignment between two uni-modal encoders and the semantic similarity in their representation spaces. We find that it's possible to bridge strong vision and language encoders using simple Projection Transformations and design a novel framework that scales up projector training to achieve compute/data efficient CLIP models that can be flexibly generalized to several tasks like multilingual retrieval, 0-shot segmentation, and long context retrieval.

Finally, Chapter 7 provides a summary of the research conducted in this thesis. The results and findings are discussed by relating them to the hypotheses and research questions presented in this Chapter. We conclude with suggestions for future work and some general remarks.

Chapter 2

Background

2.1 Introduction

In this chapter, we present the basics of neural network training, transformer networks, and contrastive learning, all of which are required to understand the current crop of vision-language models. We also discuss limited-label training scenarios commonly found in the machine learning literature and describe other works that have explored incorporating semantic information in computer vision models.

2.2 Neural Networks and Training

Deep neural networks (DNNs) have fundamentally transformed the field of computer vision over the last decade. Historically, computer vision tasks primarily relied on handcrafted features, which were painstakingly engineered based on domain knowledge. However, the advent of DNNs, and in particular Convolutional Neural Networks (CNNs), marked a paradigm shift from handcrafted to learned representations, leading to significant improvements in various vision tasks. Before the dominance of DNNs, the performance of vision systems largely depended on the quality of these handcrafted features. One popular descriptor is the Scale-Invariant Feature Transform (SIFT) introduced in [94]. Another notable one is the Histogram of Oriented Gradients (HOG) proposed in [32]. While they showed impressive results for tasks like image matching and object detection, these features could not be generalized across a vast range of problems and lacked the representational power of neural networks. The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [145] has played a pivotal role in propelling DNNs to the forefront of computer vision. In 2012, a model named AlexNet [77] dramatically outperformed traditional methods, reducing the error rate by over 10 percent. This success highlighted the potential of DNNs for computer vision and ignited the current era of research and development

dominated by such approaches.

We begin by going over the different types of neural networks and a general approach to training.

Multi-layer perceptrons (MLPs) are a type of neural network consisting of multiple layers of nodes in a directed graph. Every node in a layer connects to every other node in the subsequent layer, making it "fully connected". While they can theoretically represent any function, MLPs aren't spatially hierarchical, which makes them inefficient for tasks like image recognition that have spatial hierarchies. In this section, we describe the forward equations that govern the operation of an MLP. Let x be the input vector of size n. The input layer simply passes the input values to the next layer:

$$a^{(1)} = x$$

An MLP can have one or more hidden layers, each consisting of multiple neurons. For each hidden layer l (where $l = 2, 3, \ldots, L$), the following calculations are performed:

$$z^{(l)} = W^{(l)} \cdot a^{(l-1)} + b^{(l)}$$

$$a^{(l)} = f(z^{(l)})$$

Here, $W^{(l)}$ represents the weight matrix for the l-th layer, $b^{(l)}$ is the bias vector for the l-th layer, and $f()$ is the activation function applied element-wise to the weighted sum $z^{(l)}$.

The output layer is the final layer of the MLP. It computes the weighted sum and applies an activation function:

$$z^{(L+1)} = W^{(L+1)} \cdot a^{(L)} + b^{(L+1)}$$

$$a^{(L+1)} = f(z^{(L+1)})$$

The output of the output layer $a^{(L+1)}$ is typically the final prediction or output of the MLP.

In summary, an MLP processes input data through a series of layers, with each layer applying a linear transformation (weighted sum) followed by a non-linear activation function. These forward equations allow the network to learn complex mappings from inputs to outputs, making it a versatile tool for various machine-learning tasks.

While the forward equations describe how an MLP computes outputs from inputs, the training process involves adjusting the weights and biases ($W^{(l)}$ and $b^{(l)}$) to minimize a chosen loss function. This process, often referred to as backpropagation, uses techniques like gradient descent to update the parameters and improve

the model's performance.

Convolutional Neural Networks (CNNs) are a category of deep neural networks explicitly designed for handling visual data. They exploit the spatially local correlation present in images by applying a set of learnable convolutional filters at each layer. The forward equations for a CNN can be summarized as follows: Let X be the input data, typically representing an image with dimensions H (height), W (width), and C (channels). The input layer passes the input values to the next layer:

$$A^{(1)} = X$$

A CNN typically consists of one or more convolutional layers, followed by pooling layers and fully connected layers. For each convolutional layer l (where $l = 2, 3, \ldots, L$), the following calculations are performed:

$$Z^{(l)} = W^{(l)} * A^{(l-1)} + B^{(l)}$$

$$A^{(l)} = f(Z^{(l)})$$

Here, $W^{(l)}$ represents the weight tensor for the l-th layer, $B^{(l)}$ is the bias tensor for the l-th layer, and $f()$ is the activation function applied element-wise to the output of the convolution operation $*$.

In Figure 2.1 upper panel, we illustrate the convolutional operation for a single layer on a 3-channel input image. The convolution value is calculated by taking the dot product of the corresponding values in the Kernel and the channel matrices. The current path is indicated by the red-colored, bold outline in the Input Image volume.

$$\begin{aligned}
\text{AM[1][2]} &= \text{Red Channel Matrix Contribution(CMC)} + \text{Green CMC} + \text{Blue CMC} \\
&= (3, 9, 2) \cdot (-9, -1, 0) + (18, 32, 0) \cdot (3, -8, -6) + (39, 17, 16) \cdot (5, 2, 0) \,/2 \\
&= (\text{From-red-channel}) + \text{rcc2} \\
&= [-27 - 9 + 0 + 54 + 256 + 0 + 195 + 34 + 0] \,/2 + \text{rcc2} = 503/2 + \text{rcc2}
\end{aligned}$$

where rcc2 refers to the contribution from the remainder channels.

Spatial Pooling layers are often used to reduce the spatial dimensions of the feature maps while retaining important information. For each pooling layer l, the following calculations are applied:

$$A^{(l)} = \text{pooling}(A^{(l-1)})$$

The pooling can be max-pooling where only the maximal activation in a local

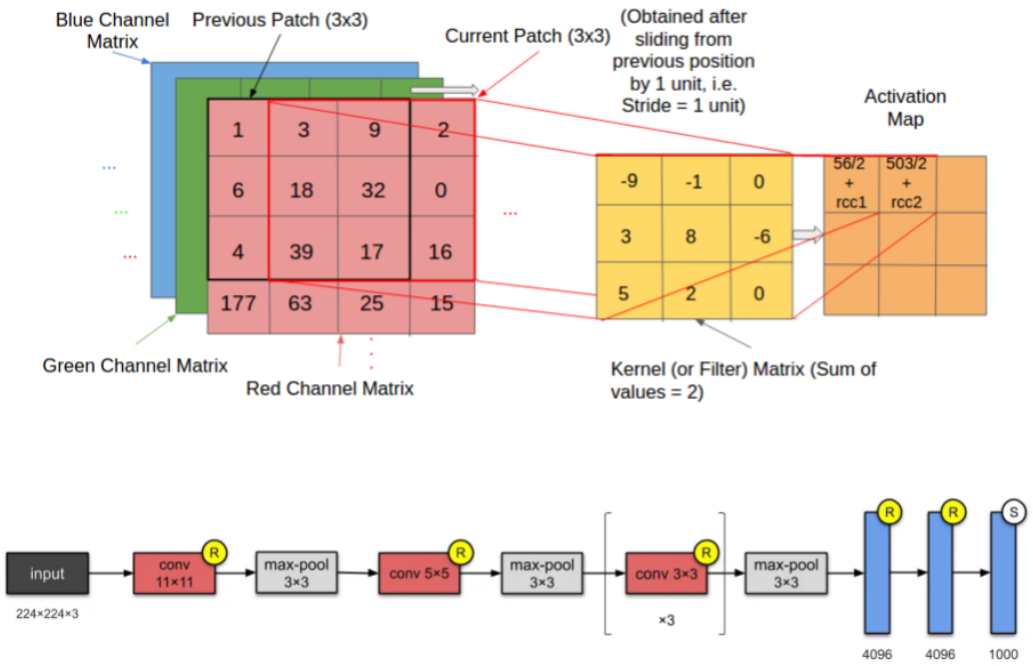


Figure 2.1: Above Convolution operation. The convolution value is calculated by taking the dot product of the corresponding values in the Kernel and the channel matrices. The current path is indicated by the red-colored, bold outline in the Input Image volume. Below: An illustration of convolutional and max-pooling layers in AlexNet. Image and explanation from [147]

group goes to the next layer or average pooling where the average of a local group goes to the next layer.

After convolutional and pooling layers, one or more fully connected layers may be used to process the flattened feature maps. These layers are similar to those in an MLP. For each fully connected layer l, the following calculations are performed:

$$Z^{(l)} = W^{(l)} \cdot A^{(l-1)} + B^{(l)}$$

$$A^{(l)} = f(Z^{(l)})$$

The output of the final fully connected layer is typically the final prediction or output of the CNN, usually of the dimension $N \times 1$, where N is the number of classes in a classification task. The training of CNNs involves adjusting the weights ($W^{(l)}$ and $B^{(l)}$) to minimize a chosen loss function using backpropagation and gradient descent.

In summary, CNNs process grid-like input data through a series of convolutional, pooling, and fully connected layers. These layers perform convolutions, non-linear activations, and spatial pooling to capture hierarchical features from the input data. The arrangement of different layers in AlexNet is illustrated in Figure 2.1. The convolutional filters, combined with spatial pooling layers that subsample the outputs

of convolutional filters to reduce the spatial size of features, allow CNNs to achieve translation invariance and reduce the number of parameters, making them more efficient than MLPs for image data. Prominent architectures like VGG [160], ResNet [61], and Inception [168] have achieved progressively better ImageNet performance by introducing deeper layers, residual connections, and multi-scale feature extraction techniques.

Deep networks learn representations by iteratively adjusting their parameters to minimize a loss function which is a measure of the difference between the predicted and true labels. Gradient descent, and its variants like stochastic gradient descent (SGD), is a crucial optimization algorithm used for this purpose. The gradients, which represent the direction and rate of the fastest decrease of the error, are computed using backpropagation [144], guiding the update of network parameters in the direction that minimizes the error. This direction is determined by the gradient of the loss with respect to the parameters.

Given a loss function $L(\theta)$, where θ represents the parameters of the model, the gradient descent update rule is:

$$\theta_{t+1} = \theta_t - \alpha \nabla L(\theta_t) \tag{2.1}$$

Here, α is the learning rate, and $\nabla L(\theta_t)$ represents the gradient of the loss function with respect to θ at iteration t.

To compute these gradients the chain rule of calculus is used, especially when dealing with deep architectures with multiple layers. Consider a neural network with a loss function L and an output y that is a function of an intermediate layer h, which itself is a function of the input x and the parameters θ. The chain rule can be used to compute the gradient of the loss L with respect to the parameters θ:

$$\frac{\partial L}{\partial \theta} = \frac{\partial L}{\partial y} \frac{\partial y}{\partial h} \frac{\partial h}{\partial \theta} \tag{2.2}$$

In the context of neural networks, this process of recursively applying the chain rule from the output layer to the input layer is termed "backpropagation". It facilitates the efficient computation of gradients, which are then used in the SGD process (or its variants) to update the model's parameters.

In essence, the combination of the chain rule via backpropagation and the SGD optimization technique allows neural networks to learn intricate patterns from data by iteratively reducing error and refining model parameters.

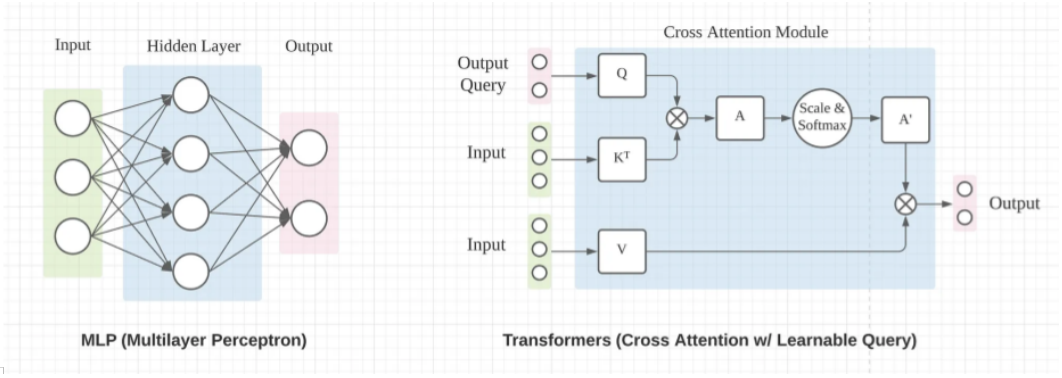


Figure 2.2: Left: An illustration of Multi-Layer Perceptron; Right: An illustration of Cross Attention Mechanism. Image from [80]

2.3 Transformer Models

The transformer architecture, introduced in [178], revolutionized NLP by showcasing a novel self-attention mechanism that allowed models to weigh the relevance of different parts of an input sequence relative to each other. This mechanism enabled the parallel processing of sequences and a more flexible representation of contextual relationships within the data.

In the realm of NLP, transformers led to the development of state-of-the-art models such as BERT [36], GPT [133], and their subsequent iterations. BERT, for instance, introduced the concept of bidirectional training of transformers, enabling models to predict missing words in a sequence by considering both left and right context. This approach showcased unparalleled performance in a variety of NLP tasks, from question answering to sentiment analysis.

Beyond NLP, the transformer architecture has found application in computer vision as well. The authors of [39] introduced Vision Transformers (ViT) which demonstrated that transformers could achieve comparable or even superior performance to traditional convolutional neural networks (CNNs) in image classification tasks. Instead of processing an image through convolutional layers, ViT divides the image into fixed-size patches, linearly embeds them, and processes the sequence of embedded patches with the transformer. In the folllowing, we briefly describe self-attention, and cross-attention which are the building blocks of all Transformer models.

2.3.1 Self-Attention

Self-Attention is a mechanism that enables a model to capture relationships and dependencies within a single input sequence effectively. It operates by computing attention scores between different elements within the sequence and using those

scores to create weighted combinations of values. Here are some key components and details of self-attention:

- **Query (Q), Key (K), and Value (V) Matrices:** Self-attention relies on three matrices: Query, Key, and Value. These matrices are linear projections of the input sequence and are learned during training. They serve different roles:

$$Q = XW_Q$$
$$K = XW_K$$
$$V = XW_V$$

  where X is the input sequence, and W_Q, W_K, and W_V are learnable weight matrices. Here the Query matrix Q is responsible for capturing the information that the model considers important or relevant. It determines what parts of the input sequence should receive more attention. The Key matrix K helps in establishing relationships between different elements in the input sequence. It provides a way to associate each element with other elements, indicating their importance in relation to each other.The Value matrix V holds the information content associated with each element in the input sequence. The attention weights obtained from the Query and Key matrices are used to weigh the values, and the weighted values are then summed to produce the final output.

- **Attention Scores:** Self-attention calculates attention scores between query and key vectors. These scores represent the similarity or relevance of each element in the sequence to others and are computed as follows:

$$\text{Attention Scores} = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)$$

  Here, d_k is the dimension of the keys, and the softmax function normalizes the scores to create a probability distribution.

- **Weighted Sum:** The attention scores are used to create weighted combinations of the value vectors, producing the output of self-attention:

$$\text{SelfAttention}(Q, K, V) = \text{Attention Scores} \cdot V$$

  In summary, the Query matrix identifies what to focus on, the Key matrix establishes relationships, and the Value matrix holds the information content. The attention mechanism uses these matrices to assign different weights to

different parts of the input sequence, allowing the model to focus on relevant information when making predictions.

- **Multi-Head Attention:** In practice, self-attention is often employed with multiple sets of query, key, and value matrices, known as "attention heads." Each head captures different relationships in the data, and their outputs are concatenated and linearly transformed to form the final output.

2.3.2 Cross-Attention

Cross-Attention extends the concept of self-attention to consider relationships between elements in different input sequences. It is commonly used in tasks like machine translation, where the model needs to align words in the source and target languages. The forward equations for cross-attention are similar to self-attention but involve query, key, and value matrices from different input sequences. This process is illustrated in the right panel of Figure 2.2

$$\text{CrossAttention}(Q_{\text{source}}, K_{\text{target}}, V_{\text{target}}) = \text{softmax}\left(\frac{Q_{\text{source}} K_{\text{target}}^T}{\sqrt{d_k}}\right) V_{\text{target}}$$

Cross-attention allows the model to capture dependencies and alignments between different parts of the input. A more in-depth description of the self and cross-attention mechanism in transformers is provided in the background of Chapter 4.

Transformers have reshaped the landscape of machine learning, leading to what are now termed "foundational models". These models, rooted in the transformer architecture, are characterized by their immense size, having billions or even trillions of parameters, and their ability to generalize across a multitude of tasks. These models, pre-trained on vast amounts of data, serve as foundational layers that can be fine-tuned or adapted to diverse specific applications. Their pretraining phase allows them to encapsulate a broad spectrum of knowledge from various domains. Consequently, instead of creating models from scratch for each new task, researchers and practitioners can now leverage these foundational models as starting points, achieving state-of-the-art performance with less data and fewer computational resources for task-specific training. The rise of transformers has thus not only propelled performance benchmarks but has also instilled a paradigm shift towards building and efficiently utilizing these foundational models in diverse domains of artificial intelligence.

2.4 Low shot learning

Deep learning-based neural networks have revolutionized image recognition and localization tasks, but a lot of data is required to train these models. Data scarcity and imbalance can occur because of several possible reasons: 1. Cost and effort of data collection; 2. Privacy and legal constraints; 3. it is not always possible to acquire enough images for each concept, particularly rare concepts, because of Zipf's law [218]; 4. evolving or dynamic environments where the object of interest may change over time can result in outdated or insufficient data. Due to these practical considerations, learning computer vision models from limited data has been an aim of researchers for a long time. It is also driven by the observation that we as humans can learn brand-new concepts from few examples or zero examples sometimes. For example, humans can recognize new bird species often by seeing a few images of the bird or even from language descriptions of the bird. The first case where we learn a model from a few examples of a class is called few-shot learning and learning an image model from language descriptions or other auxiliary semantics is called zero-shot learning.

Recent machine learning literature on few-shot learning can be classified into meta-learning and transfer learning-based methods. Meta-learning aims to extract common useful knowledge for classifying novel classes by emulating few-shot tasks during training time. They are either optimization-based methods or metric learning-based methods. In optimization methods, the objective is to meta-learn a good initialization of parameters or optimization or a combination of both, while in metric learning the objective is to learn an embedding space where similar instances are embedded together so that during test time a simple nearest neighbor classifier can be used. Transfer learning-based methods, on the other hand, take a more straightforward approach. They train a network to classify the base classes, then use the trained feature encoder and learn a classifier on top of it using the novel support instances. The work reported in [186] showed that this simple baseline works as well as several complex meta-learning methods of the time. Some recent works [154, 2, 212] use semantic information from wordnet-graphs and language embeddings to improve few-shot learning performance. In Chapter 3, we present our approach to few-shot learning where we make use of semantically similar feature representations from the robust base feature space to construct robust feature representations for novel instances and hence improve few-shot performance. We also provide an in-depth literature review of the few-shot literature.

In 0-shot learning, the aim is to adapt a base model to identify novel class instances based on some auxiliary information about the novel classes that encode the visual properties of the objects. Usually, this auxiliary information is class-level de-

scriptions from sources like domain experts, the internet, or even pre-trained LLMs. This area has seen a plethora of methods in the past decade but mostly follows one of 3 approaches. Attribute prediction networks like [79, 142, 141], take a 2-stage approach where attributes of the image are inferred first, and then its class label is inferred by finding the class with the most matching set of attributes. Joint embedding space methods like learning either a linear [50, 6] or non-linear mapping [162, 191] of the images and attributes into a common semantic embedding space using the base dataset, and then the classes of novel instances are inferred by projecting the image into the common embedding space and finding the closest novel class prototype. Generative models [148], model the seen class-attribute distributions and then use these models to generate visual instances of features of novel classes conditioned on their attributes. While most of this research focused on adapting ImageNet-trained base models to relatively small datasets like Animals With Attributes 2 (AWA2) [192], Caltech- UCSD Birds (CUB) [184] dataset, using the base split of each of these datasets, recent research into foundation models like CLIP has surpassed all these works by a large margin by training a large vision language model on 400 million image-text pairs scraped from the internet, in a contrastive manner. These models now form a solid starting point for few-shot and zero-shot adaptation to downstream datasets and hence recent research in both few-shot and zero-shot has focused on adapting these models to downstream tasks and datasets. In Chapter 4, we describe our method for improving the 0-shot and few-shot performance of CLIP to downstream datasets by making use of visually descriptive textual information that can be generated from LLMs like GPT-4 in a scalable manner. Efforts to achieve parameter-efficient alignment include LiT (Locked Image Tuning) [207], which reduces parameters by keeping the vision encoder fixed. However, LiT still requires significant compute and large-scale data for effective alignment. On the other hand, LiLT (Locked Image Locked Text) [71] keeps both vision and text encoders fixed and aligns random encoder pairs but performs well only on retrieval tasks, struggling with zero-shot domain transfer due to the limited scale of the alignment dataset. LiLT uses light-weight adapters [65] in intermediate layers and unlocked projectors for alignment. In contrast, our proposed framework, discussed in Chapter 6, simplifies the process by aligning unimodal encoders using only projectors, achieving alignment more efficiently.

2.5 Contrastive Learning and Vision Language Models

Self-supervised learning (SSL) is a prominent paradigm in computer vision that leverages unlabeled data to train deep neural networks. In SSL, the network learns meaningful representations from the data itself by generating pseudo-labels or supervisory signals. This approach has gained significant attention due to its ability to mitigate the need for extensive labeled datasets. Several popular self-supervised losses have been proposed to drive the learning process effectively. Self-supervised learning often relies on specific loss functions to train neural networks. Some popular self-supervised losses used in computer vision include:

1. **Contrastive Loss:** Contrastive loss encourages the model to project similar data samples closer in the embedding space while pushing dissimilar samples farther apart. It has been widely used in SSL, with variations like SimCLR [21] and MoCo [60] achieving remarkable results.

2. **Triplet Loss:** Triplet loss aims to minimize the distance between anchor-positive pairs while maximizing the distance between anchor-negative pairs. This loss has been instrumental in learning embeddings for tasks like face recognition [151].

3. **Rotation Loss:** The rotation loss requires the network to predict the rotation angle applied to an input image. By predicting the correct rotation angle, the model learns to capture fine-grained features and object representations [56].

4. **Contrastive Predictive Coding (CPC):** CPC is an information-theoretic loss that encourages the model to predict future patches from current ones. It has been applied to various SSL tasks and has shown success in unsupervised representation learning [120].

Contrastive learning gained popularity as an instance discrimination loss for self-supervised learning (SSL) [21, 23], outperforming several SSL losses of the time. In the contrastive loss, the objective is to minimize the distance between positive samples while maximizing the distance between negative samples in the joint embedding space, given by:

$$\mathcal{L}_{\text{InfoNCE}}(z, \hat{z}) = -\log\left(\frac{\exp(\text{sim}(z, \hat{z}))}{\sum_{i=1}^{N}\exp(\text{sim}(z, z_i))}\right)$$

Here $(z, \hat{z})$ are the positive pairs in the numerator while the denominator is calculated over the positive examples and all other negatives (z_i) to bring the positive

pairs together in the embedding space while pushing the negative pairs apart. This loss is summed over all the data points in a batch to get the average loss per batch.

For learning self-supervised image representations, an image and its augmentations are considered positive samples, while different images are considered negative. Subsequently, the same loss was used to jointly train image and text encoders on a large dataset of 400 million image-text pairs scraped from the internet where the positive pairs were from different modalities but shared similar semantic meanings. This resulted in strong visual encoders that surpass the existing performance of several ImageNet-trained encoders on a plethora of downstream datasets and tasks. The so-called CLIP [132] model is now termed a foundational VLM because of its impressive zero-shot image recognition abilities and downstream transferability.

More recent works, such as ALIGN [68], make use of a 1 billion noisy image-caption dataset to train a VLM that performs better than CLIP without the data curation efforts associated with acquiring the cleaner 400 million dataset used for training CLIP. Recently open-source efforts to recreate CLIP's performance have resulted in the collection of LAION 400 million [153] and 5B datasets [152] and Open-CLIP efforts have resulted in VLM's trained on much more data. However, it has been observed that the quality of the dataset also matters for performance – the performance of Open-CLIP [67] trained on 2B data points being inferior to that of OpenAI's CLIP with 400 million data points. With the advent of Vision-Language Models (VLMs) and the emergence of CLIP as a strong baseline, research in image recognition with limited labels—such as few-shot and zero-shot tasks—has increasingly shifted to using CLIP as the de facto starting point [53, 216, 215].

In Chapter 4, we present a more detailed literature review of Vision Language models and show how our method makes use of semantic knowledge from LLMs to improve the 0-shot and few-shot domain transfer performance of foundational VLMs.

2.6 Semantic information for Label Efficiency in Computer Vision

Auxiliary semantic information has traditionally been used in computer vision in zero-shot image recognition tasks, by virtue of the problem statement where it's required to identify images of novel classes without any training images. This has been achieved in a myriad of ways that make use of the auxiliary information of the novel and base classes in the form of language descriptions and domain expert annotations.

One of the first papers to use semantics, [154] makes use of external semantic

information from the class name, domain expert attributes, and/or natural language descriptions with an embedding model to combine with the visual features to improve the prototypes for few-shot classification. The authors of [2] follow a similar approach where they make use of a transformer-based forward and backward decoding mechanism to decode the semantic embeddings conditioned on a hybrid prototype of support and query instances. The work reported in [212] uses semantics from BERT embeddings [139] of class names and image prototypes to construct graphs of foods and use them for improved few-shot classification of images of food items. [198] uses language-based semantics to generate attention that's applied to visual prototypes to improve few-shot performance. More recent work, [95] directly uses attributes to construct attribute prototypes. The approach in [91] constructs cross-modality graphs of both the visual and semantic prototypes of support instances and uses these for few-shot inference. Orthogonally, the approach described in [124] uses semantics for both query and support samples to reduce the imbalance, and does so by learning a meta-network that predicts the query semantic feature given the support features.

In Chapter 3, we describe our method of using semantics to relate novel image concepts to already-known image concepts using semantic relationships and hence learn them more efficiently.

2.7 Representational Similarity Measures

Representational similarity measures like Canonical Correlation Analysis (CCA) [134], Centered Kernel Alignment (CKA) [75] has been explored in prior research in the context of comparing internal representations of uni-modal encoders trained from different initializations [75] and trained with different losses [74], architectures [135] etc. More recently [66] makes use of semantic similarity measures to study vision / language representations and proposes the Platonic Representation Hypothesis, which suggests that strong vision and language encoders converge toward similar representations. They introduce an alternative to the CKA metric, called Centered Kernel Nearest Neighbor Alignment (CKNNA), which calculates alignment based only on the k-nearest neighbors of each point. The authors claim that this approach effectively reduces noise in CKA measurements, resulting in improved trend clarity. Other learning-based approaches for measuring representation similarity include model stitching [9, 83]. This method introduces trainable stitching layers to swap components between different networks. If the stitched model performs well, it indicates high semantic similarity between the representations of the original networks. A more detailed comparison of semantic simlarity measures is available in 5 and [66].

We use CKA to measure the similarity across vision and language representations in 5 and 6. While in chapter 5, we use CKA as an optimization objective to establish a connection between vision and language encoders in a training free manner, in Chapter 6, we study CKA as a measure of ease-of-alignment between uni-modal encoder spaces. We provide a brief introduction to CKA in 2.7.1, but a more detailed preliminary is available in Chapters 5, 6.

2.7.1 Centered Kernel Alignment (CKA)

Centered Kernel Alignment (CKA) has emerged as a powerful tool for understanding and comparing the representations encoded by neural networks [75]. CKA quantifies the similarity between the representations learned by different layers of a network or across networks. Formally, CKA operates on two data representations $X \in \mathbb{R}^{p \times N}$ and $Y \in \mathbb{R}^{q \times N}$, computing their corresponding kernel matrices $K = k(X^\top, X) \in \mathbb{R}^{N \times N}$ and $L = \ell(Y^\top, Y) \in \mathbb{R}^{N \times N}$, where k and ℓ are kernel functions (e.g., linear or RBF kernels).

The CKA value is then calculated as:

$$\mathrm{CKA}(K, L) = \frac{\mathrm{HSIC}(K, L)}{\sqrt{\mathrm{HSIC}(K, K) \cdot \mathrm{HSIC}(L, L)}}, \tag{2.3}$$

where $\mathrm{HSIC}(K, L)$ denotes the *Hilbert-Schmidt Independence Criterion (HSIC)* [57, 97]. HSIC is computed as:

$$\mathrm{HSIC}(K, L) = \frac{1}{(N-1)^2} \mathrm{tr}(KCLC), \tag{2.4}$$

with $C = I - \frac{1}{N}\mathbf{1}\mathbf{1}^\top$ being a centering matrix that ensures the kernel matrices are mean-centered.

CKA provides a normalized measure of similarity that is invariant to orthogonal transformations and isotropic scaling of the embeddings, making it particularly suited for comparing neural network representations. Its application ranges from analyzing the alignment of representations across layers to evaluating how downstream fine-tuning tasks influence representation similarity [41]. For further properties and insights into CKA, readers are directed to [75].

2.8 Conclusion

Building upon the work and findings presented here, in Chapter 3 we present our approach to improve few-shot performance by constructing robust prototypes of novel classes from semantically similar base classes. Our work in Chapter 4 makes

use of semantic information for improving 0-shot and few-shot domain adaptation of VLMs. We also demonstrate a straightforward way of generating semantic information in a manner easily scalable to any dataset. These results suggest that both language and vision models have a high degree of representational similarity, but are complementary so that one can be used to improve the sample efficiency in the other modality.

The principles of re-entry and information integration in neuroscience illustrate that experiencing a concept through one sensory modality often triggers a corresponding sensation in another, indicating that the human brain likely forms similar representations for identical concepts across different modalities. This concept sparks our interest in examining the degree of similarity between vision and language encoders, which are separately trained on extensive datasets of images and text. Given that both modalities aim to model the same physical world, a high level of semantic similarity is anticipated. We investigate this in detail in Chapter 5 by developing methods to measure this similarity between vision and language and leveraging this connection to efficiently link unimodal vision and language encoders. Finally, in Chapter 6, we align semantically similar vision and language encoders to achieve flexible, and sample/compute efficient CLIP models. This area of study is garnering substantial interest as research labs strive to equip foundational large language models with multimodal capabilities.

Chapter 3

BaseTransformers: Attention over base data-points for One Shot Learning

This chapter provides a detailed description of the contributions of this research to answer **RQ1** *How can we incorporate semantic information to improve performance in few-shot learning?*. Specifically, we improve few-shot image classification performance by making use of semantic information from domain experts, semantic graphs like WordNet, and/or language models (**RQ1**). Few-shot classification aims to learn to recognize novel categories using only limited samples per category. Most current few-shot methods use a base dataset rich in labeled examples to train an encoder that is used for obtaining representations of support instances for novel classes. Since the test instances are from a distribution different from the base distribution, their feature representations are of poor quality, degrading performance. Here we use semantic relationships between the novel concept and the concepts the model already knows, i.e. the base concepts, to compose robust representations for the novel concepts and hence learn them in a label-efficient manner. More specifically, we propose to make use of the well-trained feature representations of the base dataset that are semantically closest to each support instance to improve its representation during meta-test time. To this end, we propose BaseTransformers, that attend to the most relevant regions of the base dataset feature space and improve support instance representations. Experiments on three benchmark data sets show that our method works well for several backbones and achieves state-of-the-art results in the inductive one-shot setting. This work was presented as a conference paper at the British Machine Vision Conference in London, UK in November 2022.

The following section provides an overview of few-shot learning methods before foundational Vision Language Models (VLMs) and a high-level overview of our

method for estimating novel prototypes using semantic relationships. Section 3.2 provides a detailed literature review of few-shot learning methods and how semantic information has been used in few-shot learning. Section 3.3 provides a detailed explanation of Prototypical Networks, Cross-Attention, and our method BaseTransformers which performs cross-attention over semantically similar base samples. Section 3.3 provides an overview of the training process and the semantic querying functions adopted for each of the datasets. Section 3.4 provides experiments and results on 3 widely used few-shot benchmark datasets. Finally, in Section 3.5 we summarize our results, discuss the limitations of our work, and lay the groundwork and motivation for the research undertaken in the next chapter.

3.1 Introduction

The development of few-shot learning models is important for the real-world deployment of artificial vision systems outside of controlled scenarios. Most previous works focus on developing stronger models, while scant attention has been paid to the properties of the data itself and the fact that as the number of data points increases, the ground truth distribution can be better uncovered. Estimating the prototype for a novel class using a single instance is fundamentally ill-posed, resulting in poor one-shot performance. [199] has shown that this can be alleviated by modeling the class conditional distribution as a Gaussian and sampling a large number of features from this distribution to train a classifier or estimate the prototype. They show that distributions of semantically similar classes in the base dataset have similar mean and variance to the distributions of the novel class. Therefore, the statistics of the class conditional distributions of novel classes are transferred from those of base classes which have been estimated with several examples (over 600) per class. This method assumes that the class conditional feature distributions are uni-modal Gaussian and that the transferable statistics are only global and not local to each base instance or its spatial locations.

We propose a novel method for estimating prototypes of unseen classes using the base dataset without making any assumptions on the distribution of the base data feature space or the transferability of the instance level or spatial level information. Our proposed method, BaseTransformers, is an end-to-end learnable cross attention mechanism that estimates a robust, base aligned prototype for novel categories by learning local part based correspondences between the support instance and semantically similar base instances. This is based on two key ideas: (i) the base dataset images are composed of semantically meaningful parts that could be reused during the classification of novel images; and (ii) since the base data features are estimated using many shots, the features corresponding to these parts are less noisy represen-

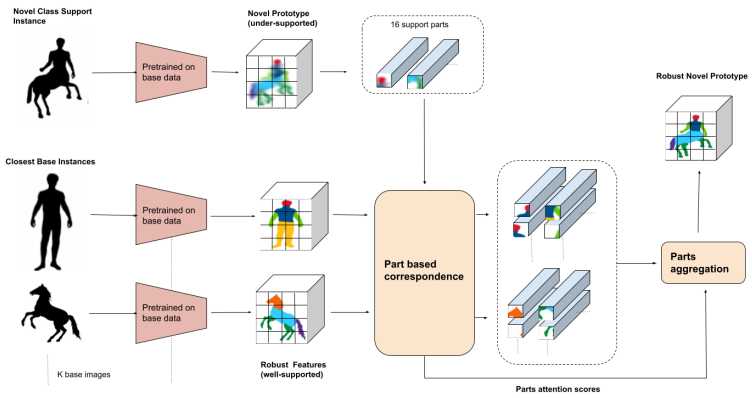


Figure 3.1: BaseTransformers construct robust novel class prototypes by attending to and aggregating semantically similar regions of the well supported base data feature space instead of using the noisy novel prototype as in Prototypical Networks [161].

tations, closer to the ground truth distribution. The concept is illustrated in Fig 1, where a novel 'centaur' class has an undersupported prototype in the feature space of an encoder pretrained on base-data. However a robust prototype of a centaur can be constructed by taking the head, torso of a human and the body and legs of horse base classes which are individually well supported in the feature space.

We hypothesize that semantically similar parts of a well represented base data feature space can be used to estimate a novel prototype that is effectively a part based composition of the well estimated base data regions. To enable this BaseTransformers allow for: (i) spatial part based comparison between the support instance and similar base instances to select the semantically meaningful regions of the robust base data feature space; and (ii) aggregation of the semantically similar parts of the base instances to estimate a novel prototype that is a composition of robust meaningful base regions. Taking inspiration from [38] we instantiate a cross attention mechanism on the feature space of the pretrained encoder to enable this. We perform this adaptation of the support instance using the base instances in the feature space and not the original pixel space, as the feature space has lower dimensions and semantically meaningful structures that are more easily transferable between the base and novel domains.

For each episode, the BaseTransformer takes the 2D feature spaces of the support instance as query and the closest base instances as the key and value. The BaseTransformer is trained end-to-end using the meta learning paradigm to identify the most relevant regions in the base data feature space and use them to compose a more robust novel class prototype. Our approach starts with a pretraining stage using cross entropy and contrastive losses on the base dataset to produce a robust encoder bypassing supervision collapse [38, 99]. This is followed by a meta training stage,

in which the encoder and BaseTransformer are jointly trained to adapt the support instances using instances from the closest base classes. To identify the closest base classes we propose using the class label information of the support instances, and making queries on the base dataset based on semantic similarity. We show that the proposed method beats the current state-of-the-art in 3 different datasets (70.88%, 72.46%, 82.27% on mini-ImageNet, tiered-ImageNet and CUB respectively) in the inductive one shot setting.

Our novel contributions are: i) We identify that robust novel prototypes in one shot learning can be obtained by part based composition of semantically similar base features; ii) We design BaseTransformer that improves the 1-shot prototypes by learning to attend to the robust 2D feature space of base instances and aggregate these to compose the novel prototype; iii) We evaluate our method on two backbones and three benchmarks to show its effectiveness in the one shot inductive setting of few shot learning.

3.2 Related Work

Meta Learning aims to extract common useful knowledge for classifying novel classes by emulating few shot tasks during training time, and are usually optimization based or metric learning based. In optimization based methods, the objective is to meta-learn a good initialization of weights [48, 146, 136, 217] or the optimization process [137, 88, 113, 197] or a combination of both [125, 8]. In metric learning methods [179, 161, 167, 201] the objective is to develop an embedding space where similar instances are close to each other in some distance sense so that a simple nearest neighbour classifier can be used during meta test time. Our method is similar to metric learning, specifically prototypical networks, as we only have an extra transformer stage to adapt the support instances to form more robust prototypes.

Transfer Learning methods train a network to classify base classes, followed by finetuning the classifier on the novel instances whilst keeping the encoder fixed. [186, 22] has shown that this simple strategy performs surprisingly well, beating/matching several complex meta learning algorithms. We follow works such as [200] and have a pretraining stage in which the encoder is trained on a combination of cross entropy and self supervised loss. Other works [166, 55, 99, 90] have shown that addition of self supervision losses in the pretraining stage provides more robust features, resulting in improved few shot performance. We use the InfoNCE loss [21] as an auxiliary loss during the pretraining stage.

A *Base Dataset* has been used explicitly during meta test time in previous works, such as in [199, 3]. The approach of [199] models the feature space of each class as a Gaussian and transfers statistics from well estimated base class distributions to

novel class distributions, and sample from this to train a classifier. In our approach, we do not assume that the class feature space follows a Gaussian distribution, but use a parametric function- a transformer to improve the prototype representation by means of attention over the feature space of base examples. The approach reported in [3] aligns the feature space of the novel instances to that of the closest base instances by reducing an adversarial alignment loss during the test time, while we do not tune any parameters of the transformer network during meta test time. Both methods make use of cosine similarity in the feature space to query the closest base classes. While this works well for us for shallow encoders, we find that making use of semantic information from the class labels results in semantically closer base classes.

Transformers have also been investigated in a similar context. Previous works like [200, 64] make use of transformer based adaptation on the feature space to improve few shot performance. The approach in [200] uses self attention over the prototypes to adapt them in a task specific manner, while the approach of [38] builds a classifier that aligns the prototypes and the queries spatially. Similarly [70, 85, 190, 59] use different forms of self-correlation and cross-correlation mechanisms to improve the relational comparison between the prototypes and the query instances. We differ from these methods, in that we explicitly attend over all spatial locations of a base data subset to improve the support instance features. To our knowledge, our work is the first to apply attention over the base data points for few shot learning.

3.3 Method

In this section we first introduce the setup of few shot classification in section 3.3.1 followed by description of our proposed method in sections 3.3.2 through 3.3.4.

3.3.1 Preliminaries

We follow the inductive setting for few shot learning. A few shot task is an N way M shot classification problem, with N classes sampled from novel classes C_n with M examples per class. $D_s = \{x_i, y_i\}_{i=1}^{M \times N}$ refers to the support set sampled from novel classes C_n. Test instances x_q are sampled from a a query set $D_q = \{x_i, y_i\}_{i=1}^{Q}$ and the goal is to find a function f that classifies x_q via $\hat{y} = f\left(x_q \mid D_s\right)$. In the few shot learning literature M is usually 1 or 5 referring to the 1-shot or 5-shot task.

Finding f from the very few examples in the support set is very difficult, so a base dataset is provided consisting of base classes C_b such that $C_b \cap C_n = \emptyset$. In the meta-learning paradigm, f is learnt by sampling several N-way M-shot tasks D_s^b and corresponding query sets D_q^b from the base dataset to emulate the test time

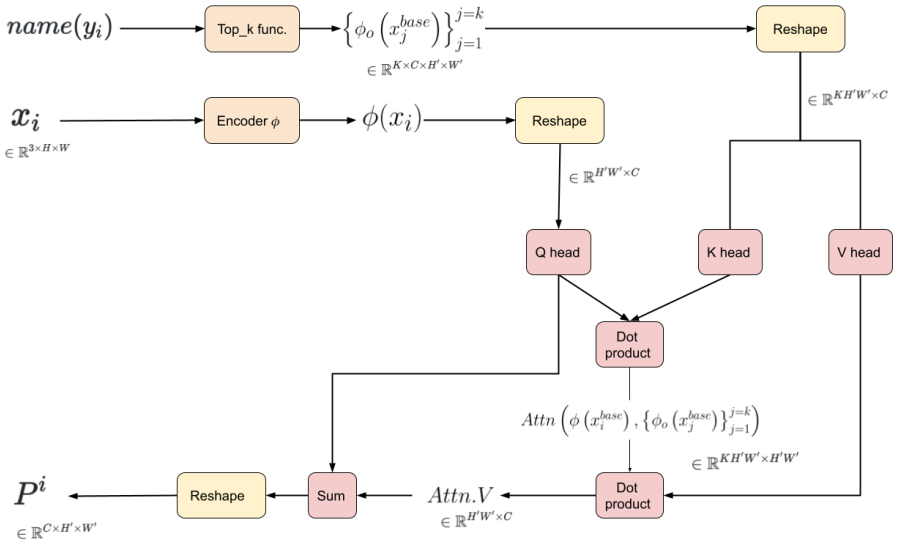


Figure 3.2: Support instance feature $\phi(x_i)$ is reshaped and projected by query head Q to obtain queries q_m^i where m corresponds to spatial locations in the support instance. q_m^i is then compared with the keys k_n^j from all spatial locations n of base instances to get attention scores $attn_{mjn}^i$, which are used to aggregate the values v_n^j and summed with original support feature $\phi(x_i)$ to obtain the base adapted prototype.

scenario. In each sampled task, f is learnt to minimize the average error on D_q^b:

$$f^* = \arg\min_f \sum_{(x_q^b, y_q^b) \in D_q^b} \ell\left(f(x_q^b \mid D_s^b),\, y_q^b\right), \tag{3.1}$$

where ℓ can be any loss that measures the discrepancy between prediction and true label.

During meta test time the optimal f^* is applied on tasks sampled from C_n. The performance of the model is evaluated on multiple tasks sampled from the novel classes C_n. For example, in prototypical networks, f consists of an embedding network E and a nearest neighbour classifier:

$$\phi_x = E(x) \in \mathbb{R}^d, \qquad \hat{y}_q = f(\phi_{x_q}; \{\phi_{x_s}^c\}), \tag{3.2}$$

where $\{\phi_{x_s}^c\}$ is the set of prototypes. Here, each prototype is given by:

$$\phi_{x_s}^c = \sum_{y_i \in c} E(x_i), \qquad (x_i, y_i) \in D_s. \tag{3.3}$$

Table 3.1: 5-way 1-shot and 5-way 5-shot classification accuracy (%) on miniImageNet dataset using ResNet-12 and Conv4-64 backbones. 95% confidence intervals reported. The numbers in bold are the best performing methods for the corresponding setting.

Setups	1-shot		5-shot	
Backbone	Conv4-64	Res12	Conv4-64	Res12
ProtoNets[161]	49.42±0.78	60.37±0.83	68.20±0.66	78.02±0.57
SimpleShot[186]	49.69±0.19	62.85±0.20	66.92±0.17	80.02±0.14
CAN[64]	-	63.85±0.48	-	79.44±0.34
FEAT[200]	55.15±0.20	66.78±0.20	71.61±0.16	82.05±0.14
DeepEMD[208]	-	65.91±0.82		82.41±0.56
IEPT[209]	56.26±0.45	67.05±0.44	**73.91**±0.34	82.90±0.30
MELR[46]	55.35±0.43	67.40±0.43	72.27±0.35	83.40±0.28
InfoPatch[90]	-	67.67±0.45	-	82.44±0.31
DMF[195]	-	67.76±0.46	-	82.71±0.31
META-QDA[211]	56.41±0.80	65.12±0.66	72.64±0.62	80.98±0.75
PAL[96]	-	69.37±0.64	-	**84.40**±0.44
BaseTransformer	**59.37**±0.19	**70.88**±0.17	73.40±0.18	82.37±0.19

3.3.2 BaseTransformer

Given a support instance x_i and its closest base instances $\left\{x_i^{base}\right\}_{i=1}^{k}$ the BaseTransformer aims to learn a representation that enables part-based adaptation of x_i by attending over all the spatial locations of all base instances in $\left\{x_i^{base}\right\}_{i=1}^{k}$.

First, an image representation of support instance $\phi\left(x_i\right)$ is obtained using the encoder ϕ, while the class name corresponding to the support instance is used to get the k closest instances in the base dataset. The top k function is described in detail in Section 3.3.3. The features of the closest base instances are passed through a fixed encoder ϕ_0 whose weights are the weights obtained after the pre-training stage on the base dataset. These representations are then used by the Transformer to establish correspondences between support instances and base instances to produce the adapted prototype. Finally, similar to prototypical networks, the Euclidean distance is used to classify the query feature $\phi\left(x_{test}\right)$ by making use of adapted prototypes $\{P_i\}_{i=1}^{N}$. Prototypical networks use 1D feature embedding while, BaseTransformers use 2D embeddings as input to allow the model to make part based soft correspondences between support and base instances, and use these to weigh the most relevant regions of base instances to estimate the prototype of a support instance as a composition of robust base parts.

More concretely, we consider a CNN without the final fully connected or pooling layers, such that $\phi\left(x_i\right) \in \mathbb{R}^{C \times H' \times W'}$. Top k function uses the pre-trained encoder phi_0 to provide the closest base instances features set $\left\{\phi_0\left(x_i^{base}\right)\right\}_{i=1}^{k}$ where $\phi_0\left(x_i^{base}\right) \in \mathbb{R}^{C \times H' \times W'}$. During meta training care is taken so as to exclude the class of the support feature itself from this set of base features so as to force the

BaseTransformer to learn to compose novel prototypes using only instances from different classes. These features are reshaped such that the attention would be between spatial locations of $\phi(x_i)$ and spatial locations of the $\phi_0\left(x_i^{base}\right)$. Key-value pairs of base instance features $K\phi_0\left(x_i^{base}\right)$, $V\phi_0\left(x_i^{base}\right)$ are obtained using two independent linear layers K, V while the transformer's queries $Q\phi(x_i)$ are obtained by using linear mapping Q on the support instance features. Here, we distinguish between a query (or test) set sample and the query of the transformer by explicitly referring to the latter as transformer's query. The dot product between transformer's query and key features results in an attention map between support features and base features. This is followed by a softmax over all spatial locations and k base instances. The computed attention is then used to aggregate the values and a residual connection from the transformer's query features is added to obtain the adapted prototype. Figure 3.2 illustrates this process. We follow the mathematical notation outlined in [38]. Let $q_m^i = Q\phi(x_{im})$ be the transformer queries i.e., the support features projected by Q, where i is the index of the support instance and m is the spatial location and $k_n^j = K\phi_0(x_{jn}^{base})$ are the key features, i.e., the base features projected by K where j is the index of the base instance and n is the index of the spatial location. An attention map $\widetilde{\text{attn}}$ between support features and base features is calculated as:

$$\widetilde{\text{attn}}_{mjn}^{i} = \frac{\exp(\text{attn}_{mjn}^{i})}{\sum_{mjn}\exp(\text{attn}_{mjn}^{i})}, \quad \text{where} \quad \text{attn}_{mjn}^{i} = \langle k_n^j, q_m^i \rangle. \tag{3.4}$$

Next the base adapted prototype P_m^i at spatial location m is obtained as follows:

$$P_m^i = q_m^i + \sum_{jn} \langle \widetilde{\text{attn}}_{mjn}^{i}, v_n^j \rangle. \tag{3.5}$$

For a test instance x_{tm}^{test}, logits are obtained by calculating the similarity and averaging over the spatial and channel locations as,

$$\text{sim}(\phi(x_t^{test}), p^i) = -\frac{1}{H'W'}\sum_m \left\|\phi(x_{tm}^{test}) - P_m^i\right\|_2^2. \tag{3.6}$$

Here we do not update the features of the base instances during training so as to not corrupt the base data features that have been learnt using several examples per class. The features of a random subset of base instances are computed using the pretrained encoder f_0 and stored in a memory bank, which is then queried by the top-k querying function described in Section 3.3.3.

Table 3.2: 5-way 1-shot and 5-way 5-shot classification accuracy (%) on tieredImageNet dataset for ResNet-12. The numbers in bold are the best performing methods for the corresponding setting.

Setups	1-shot	5-shot
ProtoNets[161]	65.65	83.40
SimpleShot[186]	69.75	85.31
FEAT[200]	70.80	84.79
CAN[64]	69.89	84.23
DeepEMD[208]	71.16	86.03
IEPT[209]	72.24	86.73
MELR[46]	72.14	**87.01**
InfoPatch[90]	71.51	85.44
DMF[195]	71.89	85.96
META-QDA[211]	69.97	85.51
PAL[96]	72.25	86.95
BaseTransformer	**72.46**	84.96

Table 3.3: Test accuracy over number of shots for BaseTransformer and SupportTransformer

shot	1	2	3	4	5
BT	70.8	74.61	78.1	80.23	82.37
ST	66.34	73.12	77.33	79.8	82.01

Table 3.4: 5-way 1-shot and 5 way 5-shot classification accuracy (%) on CUB dataset. The numbers in bold are the best performing methods for the corresponding setting.

Setups	1-shot		5-shot	
Backbone	Conv4-64	Res12	Conv4-64	Res12
ProtoNets[161]	64.42	-	81.82	-
FEAT[200]	68.87	-	82.90	-
DeepEMD[208]	-	75.65	-	88.69
IEPT[209]	69.97	-	84.33	-
MELR[46]	70.26	-	**85.01**	-
BaseTransformer	**72.15**	**82.27**	82.12	**90.64**

3.3.3 Querying function

We use a semantic similarity based querying function, which uses the label name of the support instance and finds the 5 closest base classes in a semantic space that varies according the dataset. Then base instances are sampled randomly from these classes such that they sum up to k. For mini-Imagenet dataset the semantic similarity is equal to the LCH-similarity[81] of the labels in the WordNet graph[108]. LCH similarity between class labels do not work well for tiered-ImageNet because the class splits were made using higher up nodes in the WordNet hierarchy resulting in very similar LCH similarity scores between a test class label and many base class labels. Hence, we use BERT[37] embeddings of the word labels concatenated with their hypernyms from WordNet to find more semantically similar base classes. For CUB, category-level attributes describing the visual features of each bird species are already available. Similar to [156], we use the cosine similarity between normalized category attribute vectors to query the closest base classes.

3.3.4 Training

Following [38, 99, 90] we note that we require base embeddings that contain more information than just information regarding base classes to be effective for adapting novel classes. To restrict supervision collapse, we train our encoder with an auxiliary contrastive loss in the pretraining stage. We follow a version of InfoNCE loss from

[22], where the distance measure is Euclidean instead of cosine distance.

$$l(i,j) = -\log\left(\frac{\exp(s_{i,j})}{\sum_{k=1}^{2N} 1_{k!=i}\exp(s_{i,k}))}\right), \tag{3.7}$$

$$L_{\text{infoNCE}} = \frac{1}{2N}\sum_{k=1}^{N}\left[l(2k-1,2k) + l(2k,2k-1)\right], \tag{3.8}$$

where $s_{i,j} = -\|f_i - f_j\|_2^2$ and f_i, f_j are features of SimCLR [21] style augmented images in a minibatch. Concretely, the pretraining is a N_b way classification task where N_b is the number of classes in the base dataset. It is evaluated on a 16-way 1-shot classification task on the validation set. The complete pretraining objective is:

$$L_{\text{pretraining}} = L_{\text{classification}} + b \times L_{\text{InfoNCE}}, \tag{3.9}$$

where b is a hyperparameter balancing the auxiliary loss and $L_{\text{classification}}$ is a N_b way cross-entropy loss.

After pretraining, we train the transformer and the encoder end to end in a meta-learning fashion similar to [200]. Because the feature encoder is pretrained on base dataset, a lower learning rate(factor of 10) is used for the feature encoder to ensure convergence. Similar to the pretraining stage we use unsupervised InfoNCE loss as an auxiliary loss along with the cross entropy loss during meta training stage to restrict supervision collapse.

3.4 Experiments

We evaluate our method on three different datasets, namely mini-Imagenet, tiered-Imagenet and CUB [184]. Mini-Imagenet and tiered-Imagenet are subsets of the Imagenet dataset designed specifically for few shot learning. Mini-Imagenet dataset consists of 60,000 images across 100 classes of which train, validation, and test have 64, 16, and 20 classes respectively. We follow the split specified in [137] with 64 classes in the base dataset. Tiered-Imagenet is a larger dataset consisting of 351, 97, and 160 categories for model training, validation, and evaluation, respectively. We follow the split specified in [200]. In addition to this, we also look at a more fine grained few shot classification task using the CUB dataset that consists of images of various species of birds. CUB dataset contains 11,788 images split into 100, 50, and 50 classes for train, validation, and test. For all images in CUB dataset, we use the provided bounding box to crop all the images as a preprocessing step [173]. We follow the split specified in [200]. Similar to [200, 146], we use 10,000 randomly sampled few shot tasks for testing as well as report the average accuracy and 95%

Table 3.5: 1 shot results using oracle querying function

Querying Setup	mini-ImageNet	tiered-ImageNet
Visual	67.40±0.20	71.05±0.18
Semantic	70.88±0.17	72.46±0.19
Oracle	72.38±0.18	76.55±0.17

Table 3.6: Ablation study of BaseTransformer

SimCLR-pre	Querying	BT	1-shot
No	NA	No	51.65
Yes	NA	No	52.68
No	Visual	Yes	54.46
Yes	Visual	Yes	57.38
Yes	Semantic	Yes	59.37

confidence intervals.

3.4.1 Implementation details

We test our method with two networks popularly used in the few shot learning literature, namely Conv4-64 – a 4 layer convolution network [179, 161, 173, 200] and ResNet-12 – a 12-layer residual network [82, 200]. As mentioned above we have an additional pretraining stage over the base dataset before the meta training stage. We use images resized to input resolution of 84×84 for both networks.

In pretraining stage, we use SGD with momentum with an initial learning rate of 0.1 which is decayed by 0.1 using a custom schedule for both networks, similar to [200]. For weighing the auxiliary contrastive loss, we use balance $b = 0.1$.

In the meta learning stage, we use SGD with momentum with an initial learning rate of 0.002 and $\gamma = 20$ for Conv4-64 and an initial learning rate of 0.0002 and $\gamma = 40$ for ResNet-12. We follow the standard implementation of multi-headed self attention as presented in [178]. In meta training stage, the temperature hyperparameter used for softening the logits is critical for convergence to a good solution. We set the temperature as 0.1 for both networks. The optimal value for k is set to 30 after a hyperparameter search.

The memory bank consists of features of 200 randomly sampled instances per base class computed using the trained encoder f_0. The value of k was fixed to be 20 after trying out values of k from 2 to 30 and choosing the best performing value on 1-shot classification on mini-ImageNet.

3.4.2 Results

We report the results of BaseTransformer and other methods for mini-ImageNet in Table 3.1 and tiered-ImageNet and CUB in Table 3.2 and 3.4 respectively. We can see that one shot performance of BaseTransformers is better than all competing methods. For fairness, we have excluded comparisons with works that use larger encoders or extra image data [199]. We make the following observations: 1) BaseTransformers are effective in improving 1 shot performance on all considered backbones and benchmarks; 2) In comparison to other works [200, 64] that use

transformers for prototype adaptation, we show improvements of 4.1%, 1.66%, and 3.28% on mini-ImageNet, tiered-ImageNet, and CUB dataset in the 1-shot setting; 3) We do not see the strong improvements in 1-shot reflected in the 5-shot setting. We hypothesize that this could be because the prototypes in 5-shot setting are already a good estimate of the true prototype. We investigate this phenomenon in 3.4.5. Results with the oracle top-k querying function are reported in 3.4.6. We also compare against other methods that use semantic knowledge in 3.4.7.

3.4.3 Ablation studies

Table 3.6 provides detailed ablation study of the various parts of our method for the Conv4-64 encoder. We can see that performance without BaseTransformer and SimCLR-pretraining is similar to that of Prototypical Networks. Including just InfoNCE as the auxiliary loss in the pretraining stage improves performance by 1.3%. Applying BaseTransformers with visual querying on Prototypical Networks further improves one shot accuracy to 54.46%. Using SimCLR in the pretraining stage with BaseTransformers improves accuracy further to 57.38%. This shows that the SimCLR loss in the pretraining stage is necessary to prevent supervision collapse and provide the BaseTransformer with robust base features. Finally, applying semantic querying gives a further improvement of $\sim 2\%$.

3.4.4 5-shot variations

For the 5-shot case, we experiment with two different ways of averaging the support instances to form a prototype. Pre-avg averages the support instances before the BaseTransformer. The closest base instances in this case are sampled randomly from the 5 closest base classes using semantic similarity as described in Section 3.4 in the main paper. In contrast, for post-avg we adapt each support instance and its corresponding set of closest base instances independently and the prototype is obtained by averaging the adapted support instances after the BaseTransformer. Table 3.7 reports the results for both pre-avg and post-avg for 5-shot classification on the mini-ImageNet dataset using a ResNet12 encoder. Here we can see that pre-avg works much better than post-avg for the ResNet12 encoder. We believe that this could be because averaging the support instances results in a more robust input to the BaseTransformer, aiding in its training.

3.4.5 5-shot results

We believe that the performance improvements from using base dataset is only significant in the 1-shot to 3-shot domain. We ran experiments comparing Base-

Table 3.7: Results for different setups considered for averaging of support instances in 5-shot setting.

Setup	5-shot
Pre-avg	78.38±0.23
Post-avg	82.05±0.19

Table 3.8: Comparison with semantic knowledge baselines

Method	mini 1-shot	CUB 1-shot	tiered 1-shot
RS_FSL [2]	65.33±0.83	65.66±0.90	-
MS [154]	67.3	76.1	-
AM3 [194]	65.30±0.49	74.1	69.08±0.47
KTN [128]	64.42	-	-
BT (Ours)	70.88±0.17	82.27±0.19	72.46±0.19

Transformer (BT) with semantic querying to SupportTransformer (ST), a variant of BT where the $Q = \sum_{y_i \in c} \phi(x_i)$ and $K = V = \{\phi(x_i)\}$ where $y_i \in c$, keeping all other hyperparameters same. Here Q is the prototype of class c and $K = V$ are the support instances of class c. Test accuracy of ST approaches that of BT as the number of shots approaches 5, showing that the prototypes from 5 different support instances of the novel class become as good as the prototype computed using base instances queried via a semantic query (Table 3.3).

3.4.6 Oracle querying

Table 3.5 reports 1-shot classification results using visual, semantic, and oracle querying for the mini-ImageNet and tiered-ImageNet datasets. Oracle querying uses the ResNet-12 encoder trained on both seen and unseen classes in the dataset. Then the closest base classes are found by the Euclidian similarity between the class prototypes estimated using all the instances in the class. We see that by improving the querying function, BaseTransformers can improve 1-shot accuracy by a significant margin, especially for the tiered-ImageNet dataset where the classes are distributed into seen and unseen classes with limited semantic overlap [123]. This shows that the 1-shot performance of the BaseTransformer architecture is limited by the querying function. We leave the search for an optimal querying function for the future.

3.4.7 Comparison with Semantic knowledge baselines

Previous methods that use semantic knowledge [154, 2, 194, 128] use it explicitly to structure the feature space, while we use it only for querying. Despite this, our method outperforms all these methods (see Table 3.8).

3.4.8 Visualization of learnt attention over base data points

We visualize the attention maps learnt by the BaseTransformer in Fig. 3.3. These are obtained by overlaying the resized attention map over the corresponding image

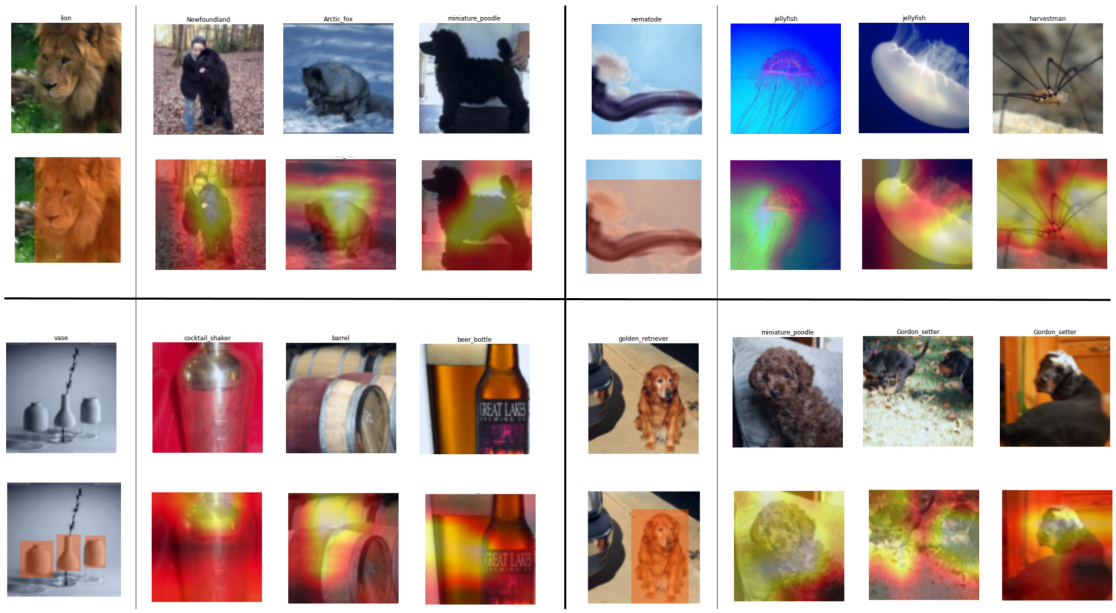


Figure 3.3: Left: support instance; right: the three closest base instances (top) and attention maps overlaid over the closest base instances (bottom). It can be seen that BaseTransformers learns to select visually similar features from the base feature space using the learnt part-based correspondences. Warmer color corresponds to higher attention weight.

of base instance selected by the querying function. We can see that for each support image, BaseTransformer has learned to attend to visually similar regions of base instances. For example (Fig. 3.3 quadrant 2), for support instance nematode, the BaseTransformer learns to attend to the tentacle of jellyfish or the legs of harvestman to improve the prototype representation. It is also worth noting that in some cases the BaseTransformer is successful in identifying multiple visually similar features in base instance images when there are multiple instances of the class in one image. For example (Fig. 3.3 quadrant 4), for golden retriever, the BaseTransformer attends to two instances of gordon setter without being explicitly trained to identify multiple gordon setters.

3.5 Conclusion

In this chapter, we show that semantic relationships can be utilized to learn novel concepts using less data by relating them to well-learned base concepts. We propose a novel method that answers **RQ1** *How can we incorporate semantic information to improve performance in few-shot learning?*. Specifically, we propose that the one-shot performance of metric learning-based few-shot approaches is hindered by bias in the estimation of the prototype. We show that prototype estimation can be improved by relating to concepts that the model already knows and reusing the features of

well-supported base instances. Our proposed method, BaseTransformers, adapts the prototype by making use of learned correspondences between the support instance and the semantically closest base class instances. Extensive experiments on three benchmarks and two encoders show the effectiveness of our method advancing the state of the art at the time in the inductive one-shot setting. A primary limitation of the work presented in this chapter is that acquiring semantic information in the form of domain expert descriptions of bird species for the CUB [184] dataset, and WordNet graph [108] for the mini-ImageNet and tiered-ImageNet dataset, is a time-consuming and costly process and is necessary for the success of our method. We tackle this limitation in the next chapter by treating LLMs trained on large amounts of data as approximate world models and querying them for acquiring semantic information in a flexible and scalable manner.

Soon after the publishing of the work presented in this chapter, there was a significant paradigm shift in few-shot and zero-shot research with the arrival of foundation models like CLIP [132], ALIGN [68] which have been trained on Billion scale image-caption pairs showcasing impressing 0-shot classification accuracies on several downstream datasets like ImageNet [145] or Caltech101 [47]. A more detailed analysis of the classification results of CLIP shows that performance on certain specialized and fine-grained datasets/domains like DTD [26], FGVC Aircraft [98], and CUB [184] datasets seem to be lacking in comparison to natural datasets like ImageNet. We believe that even though it is possible to collect webscale datasets there would always be a long tail of concepts with low occurrence due to the 'long-tail' phenomenon or Zipf's Law [218]. We believe this to be the reason for the poor performance of CLIP on fine-grained and specialized datasets like FGVC-Aircraft [98], DTD [26], CUB [184], etc, and propose that including semantic descriptions of the classes in each of these datasets could be used to improve the 0-shot and few-shot domain transfer of CLIP-like foundation models. In the next chapter, we define the kind of semantic information that's most useful for visual classification tasks as well as how to generate it in a scalable manner for improving the generalizability of foundation models like CLIP.

Chapter 4

Enhancing CLIP with GPT-4: Harnessing Visual Descriptions as Prompts

In this Chapter, we detail our attempt at answering research questions **RQ2**, and **RQ3**. More specifically we present a detailed description of our contributions to understanding what type of semantic information from language is useful for improving the 0-shot and few-shot domain transfer performance of CLIP-like models (**RQ3**)and how this type of semantic information can be generated in a flexible manner scalable to any dataset without any human annotation effort from domain experts (**RQ2**). We show that GPT-4 can be used to generate text that is visually descriptive and how this can be used to adapt CLIP to downstream tasks. We show considerable improvements in 0-shot transfer accuracy on specialized fine-grained datasets like EuroSAT ($\sim$ 7%), DTD ($\sim$ 7%), SUN397 ($\sim$ 4.6%), and CUB ($\sim$ 3.3%) when compared to CLIP's default prompt. We also design a simple few-shot adapter that learns to choose the best possible sentences to construct generalizable classifiers that outperform the recently proposed CoCoOP by $\sim$ 2% on average and by over 4% on 4 specialized fine-grained datasets. The work in this Chapter was presented as a Proceedings track paper at the 'What's Next in MultiModal foundation models' workshop at the International Conference of Computer Vision held in Paris, France in October 2023.

The following section presents an overview of foundational VLMs like CLIP and our method. Section 4.2 provides a detailed literature review of Vision Language Models, Few-shot adaptation techniques for VLMs and a discussion on extracting semantic information from LLM's. Section 4.3 provides an overview of different adapter baselines, our prompt design for the LLM to generate visually descriptive sentences for the classes, 0-shot adaptation strategy and our few-shot adapter that

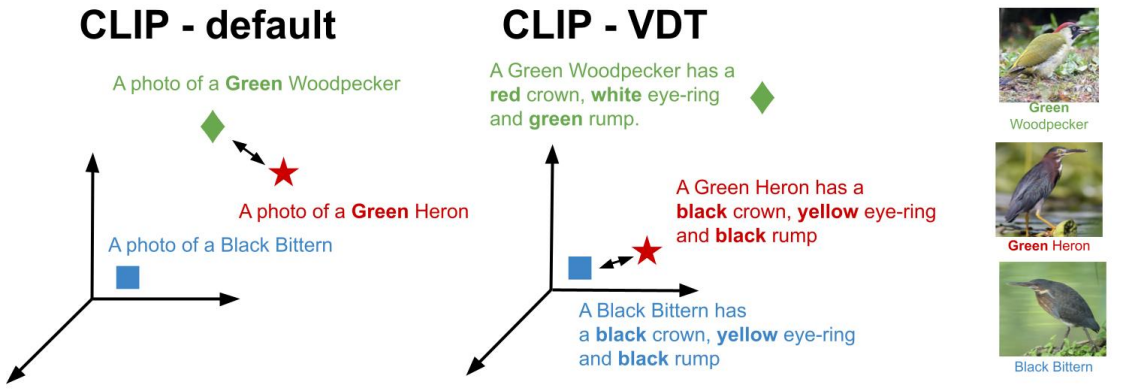


Figure 4.1: An example showing three birds, Green Heron, Green Woodpecker, and Black Bittern. Green Heron and Green Woodpecker have close-by classification prototypes by virtue of not having enough details in the prompt template. Only the text-encoder's embedding space is visualized. Here we see that adding visual descriptions to the prompt resolves this issue and moves the classification prototypes in the word-encoder's space such that classification prototypes for visually similar birds (Green Woodpecker and Black Bittern) lie together.

uses this semantic information. Section 4.4 presents the results of our methods on a diverse benchmark of 12 datasets covering various natural fine-grained specialized image classification settings. Finally we provide a summary and discussion of the limitations of this work and VLMs in Section 4.5.

4.1 Introduction

Contrastive pre-training of large-scale VLMs has demonstrated remarkable image classification performance on open-set classes. Models like CLIP [132] and ALIGN [68] are pretrained on web-scale datasets consisting of image-text pairs (over 400 million and 1.8 billion respectively), resulting in a highly generalizable model with competent 0-shot domain adaptation capabilities. While vanilla supervised training is performed on a closed set of concepts or classes, CLIP pretraining uses natural language. This results in a joint text-vision embedding space that is not constrained to a fixed set of classes. In CLIP, the classifier is constructed by plugging the class name into a predetermined prompt template like 'a photo of {class name}'. A straightforward way to adapt CLIP to different domains is by prompt engineering, which usually involves modifying the prompt template to include semantic information about the target task. For example, to classify bird images, one could construct a prompt 'a photo of {classname}, a type of bird'. This prompt engineering process, however, is not optimal because it: 1.) requires domain expertise in the target domain; 2.) has high variance – small changes to the prompt result in large variation

in performance; 3.) has a fixed prompt template for all the classes, therefore only the class name in the prompt provides the classification anchor, which might not contain enough information to distinguish different classes. For example, in Fig 4.1 we see an image of a Green Heron, which from the name would suggest that it is predominantly a green-colored bird and we would assume that it is similar to Green Woodpecker if we have never seen either bird. However, we can see that it is in fact a blackish-brown bird with a chestnut-colored neck and visually more similar to a bird like the Black Bittern. For 0-shot transfer to fine-grained datasets like this to work well, CLIP has to either have seen and associated images of a Green Heron to the text 'Green Heron' from its large pretraining dataset or additional information in the form of *visually descriptive textual* (VDT) information is required. Here we define VDT as a set of sentences that describe the visual features of the class under consideration including shape, size, color, environment, patterns, composition, etc. While most humans can identify many different common bird species just from their names, they would need access to an ornithology taxonomy of bird descriptions to identify more rare bird species. Similarly, we argue that CLIP's 0-shot accuracy can be improved by incorporating VDT information into the prompts. As shown, in Fig 4.1, including VDT information like *black crown* and *black rump* moves the classification prototype of Green Heron away from the classification prototype of Green Woodpecker and towards that of Black Bittern in the text-encoder's embedding space.

In this work, we first show that we can use VDT information for each class in the target domain to construct class conditional prompts that achieve performance improvements over CLIP's default prompt. We show this on the CUB dataset [184] by constructing sentences from domain experts about the bird species in Section 4.3.2 as they are readily available as part of the dataset.

However, we acknowledge that domain expert annotations are costly and time-consuming to obtain, hampering the scalability of our method to other datasets. To address this, we focus on the recent advances in *generative pretrained Large Language Models (LLMs)* like GPT-4 to construct these class conditional prompts in a manner easily scalable to other datasets. These models are a good fit for the task of constructing sophisticated prompts, because: 1) they are a condensed form of human knowledge (trained on web-scale text data) [202]; 2) they can be manipulated to produce information in any form or structure which makes compatibility with CLIP's prompt style relatively simple. Therefore we use GPT-4 to construct visually descriptive textual information about the classes with special emphasis in the GPT-4 prompts about visual cues like shape, color, structure, and compositionality. We use the generated VDT information to construct prompt ensembles that are passed through CLIP's text encoder and aggregated to generate classifiers that are then used

for 0-shot classification. Using GPT-4 circumvents the need for domain knowledge and conveniently provides class conditional prompts. Prompt ensembling the VDT sentences reduce CLIP's performance sensitivity to small changes in the prompt. We show performance improvements over vanilla CLIP with the default prompt on 12 datasets with an average improvement of 2% and even better improvements in fine-grained datasets like EuroSAT ($\sim$ 7%), DTD ($\sim$ 7%), SUN397 ($\sim$ 4.6%), and CUB ($\sim$ 3.3%). The prompts and all the auxiliary class information will be made publicly available to promote research in prompt ensembling and multi-modal adapter design.

Finally, we design a simple adapter that learns to adaptively select and aggregate the best sentences for any given dataset and show that making use of this additional VDT information improves the few-shot domain transfer performance of CLIP as well. We demonstrate the few-shot adaptation performance for the recently proposed Base-to-New setting on a benchmark of 12 datasets and outperform recent methods like CoOp [216] and CoCoOp [215] despite having fewer model parameters, shorter training time, and a simpler model architecture.

In short, our contributions are as follows:

1. We show that including visually descriptive textual (VDT) information in prompts results in better 0-shot domain transfer performance of CLIP.

2. We use GPT-4 to generate VDT sentences in a scalable manner and show consistent performance improvements over CLIP in 0-shot domain transfer.

3. We design a simple adapter network to make use of this extra information for few-shot transfer and show performance improvements over methods like CLIP-Adapter and CoCoOp [215] for few-shot domain transfer in the Base-to-New setting.

4. We release all the VDT information for all 12 datasets to promote further research in multi-modal prompt and adapter design for low-shot domain transfer of large VLMs.

4.2 Related Works

4.2.1 Vision Language Models

Recent VLMs [68, 132, 51] jointly learn the vision and language encoders from scratch and have demonstrated impressive 0-shot domain transfer performance. As mentioned in [216], this can be attributed to transformer networks [178], contrastive losses [21, 60], and web-scale training datasets [132, 68].

While our GPT-generated prompt ensembles are similar to CLIP's prompt ensembles, CLIP's prompt ensembles were constructed and tuned manually, and are class agnostic, while ours were generated by GPT models that were prompted to provide VDT information for each class.

4.2.2 Prompt Learning

CoOp [216] successfully used prompt learning in VLMs but had generalizability limitations due to overfitting on the few-shot dataset [215]. In response, CoCoOp was proposed, enhancing performance with image-conditioned prompt learning using a meta-network, albeit at a higher resource cost. We address generalizability differently by using class conditional VDT information. Our simpler and more efficient model, CLIP-A-self, outperforms CoCoOp in the Base-to-New few-shot setting.

4.2.3 Few-shot adapters for Vision Language models

CLIP-Adapter [53] (CLIP-A) offers a simpler few-shot transfer method for VLMs, utilizing an MLP trained on fixed image/text encoders. Our CLIP-A-self is different from CLIP-A in that we apply a self-attention mechanism on the set of all sentences for any class, learning to select and aggregate the best subset of VDT information for the dataset from the few-shot training set. Although Tip-adapter [210] showed superior performance on base classes with a cache model, it's inapplicable in the Base-to-New setting due to its reliance on few-shot test class examples, making it irrelevant for our comparison.

4.2.4 Semantic information from Large Language Models

Recent advancements in transformer-based language models, particularly the GPT family [15, 1], have demonstrated exceptional abilities in semantic extraction from intricate texts. Their application to vision tasks has emerged as an active area of research. [114] employs Palm540B LLM [25] to generate semantic data for unsupervised class embedding vectors in 0-shot classification, but only tests on three legacy datasets. Our research presents results on a modern benchmark of 12 datasets. Recently, [130, 104] leverage GPT-3 for class conditional prompts to enhance CLIP's 0-shot domain transfer on 6 datasets. While [104] focuses on using GPT-3 to construct visual descriptors that aid in the interpretability of CLIP's predictions during 0-shot domain transfer, we argue that 0-shot domain transfer performance improves with the inclusion of high-quality VDT information. Hence, we make use of GPT-4 for richer, more diverse, and more accurate VDT information.

While [104] utilize GPT-3, probability space ensemble, and highlight VDT's role in 0-shot transfer, our method differs. We use GPT-4 for auxiliary data collection, perform ensemble in word-encoder space, and introduce a few-shot adapter for optimal VDT selection in few-shot transfer. [175] uses GPT-3 for prompt construction in diffusion models to generate images for support sets while our work only uses GPT4 to acquire auxiliary text data. To our knowledge, we are the first to prompt GPT-4 for visually descriptive sentences to improve CLIP's 0-shot and few-shot domain transfer.

4.3 Methodology

4.3.1 Review of CLIP and CLIP-Adapter

Through contrastive pretraining on large image-text datasets, CLIP performs image classification on various concepts, aligning related images and texts in a shared embedding space, while separating dissimilar ones. After pretraining, CLIP directly performs image classification on the target dataset without any finetuning. First, we review how the CLIP model performs 0-shot classification on an open set.

The CLIP model, comprising a vision and language model, encodes an image and its corresponding caption into visual and textual embeddings, respectively. During inference, these embeddings are compared using *cosine similarity*. Given an image $I \in \mathbb{R}^{H\times W\times C}$, where H, W, C denotes the height, width, and number of channels of the image, the vision encoder transforms the image into the joint embedding space to get the image features $f \in \mathbb{R}^D$ where D represents the dimension of the features.

During inference, a prompt template such as 'A photo of {classname}' is used to generate sentences for K different classes and passed through the text-encoder to yield classifier weight matrix $W \in \mathbb{R}^{D\times K}$. Prediction probabilities are then calculated by multiplying image feature f with W and applying a softmax function:

$$f = \text{Backbone}(\mathbf{I}), \quad p_i = \frac{\exp(\mathbf{W}_i^T f)/\tau}{\sum_{j=1}^{K} \exp(\mathbf{W}_j^T f)/\tau}, \tag{4.1}$$

In CLIP [132], 0-shot domain transfer utilizes domain-specific information in the prompt template, such as 'A photo of a {class-name}, a type of bird' for bird images. [132] reports that careful prompt design and prompt ensembling are important to improve 0-shot classification accuracy. Prompt ensembling is achieved by constructing several prompts for each class and then averaging the classification vectors. In our work, we show that prompt ensembles of VDT information improve CLIP's 0-shot domain transfer.

CLIP-A [53] is a learnable MLP adapter applied to image and/or word en-

Table 4.1: Comparing visual and non-visual prompt ensembles for 0-shot domain transfer to the CUB dataset.

Prompting	Default	Non-Visual-GT	Visual-GT	Visual-GPT
Accuracy	54.7	53.0	57.7	57.4

coder features for few-shot transfer to target datasets. During few-shot transfer, given N images per class with labels, denoted as $(x_{i,k}, y_{i,k})_{i=1,k=1}^{i=N,j=K}$, K classifier weights are constructed using the prompt template H and text encoder g as $W = g(H(classname(\{y_{i,k}\})))$. The image features f and text features W pass through the learnable adapters A_v, A_t to get adapted features as follows.

$$f^{\star} = \alpha A_v(f)^T + (1-\alpha)f, \tag{4.2}$$

$$\mathbf{W}^{\star} = \beta A_t(\mathbf{W})^T + (1-\beta)\mathbf{W}. \tag{4.3}$$

The hyperparameters α and β blend CLIP's knowledge with fine-tuned knowledge to avoid CLIP-Adapter overfitting. Logits are calculated as per Eqn 4.1, and cross entropy loss over the entire training set $(x_{i,k}, y_{i,k})_{i=1,k=1}^{i=N,j=K}$ is used to optimize A_v, A_t.

In the *All* setting, few-shot transfer is tested on a hold-out dataset with images from the K classes used in training. In the Base-to-New setting, proposed by [215], the evaluation occurs on U non-overlapping classes. Our model is evaluated in the more practical Base-to-New setting.

4.3.2 Language Model Prompt Design

In this section, we show that using VDT information in the prompt template improves CLIP's 0-shot transfer capabilities and describe our approach to generate class-specific prompts using an LLM.

Visual Descriptive Sentences

[132] demonstrates that careful prompt design and prompt ensembling improve the 0-shot classification performance of CLIP. Here we ask the question: What type of information can be appended to the prompt template to improve the 0-shot domain transfer performance? We show that appending visually descriptive information to the prompt template and ensembling improves the 0-shot performance over the default prompt and prompts containing non-visual information.

Using the CUB dataset with expert annotations, we contrast the 0-shot performance of visual and non-visual prompt ensembles. For the visual prompts, we

take class attribute vectors detailing attributes like color, pattern, shape, etc. for 28 bird body parts, leading to 312 scores per bird. We use the most pronounced attribute-value pairs to form 28 visual prompts (denoted *Visual-GT*) such as 'A photo of Green Heron. Green Heron has a greenish-black head cap.' Conversely, for non-visual prompts (denoted *Non-Visual-GT*), we collect information on bird calls, migration, behavior, and habitat, yielding 12 different prompts like 'A photo of Green Heron. The green heron's bird call is a loud, harsh 'skeow" per class.

We derive classification vectors for *Visual-GT* and *Non-Visual-GT* by averaging class-level sentence embeddings within CLIP's joint embedding space, considering its 77-token limit. Table 4.1 shows no improvement using *Non-Visual-GT* prompts over the default, yet a 4% improvement with *Visual-GT*.

Prompting LLMs for visually descriptive information

In the prior section, we highlighted the use of expert VDT information in creating class-specific prompts to enhance CLIP's 0-shot performance. However, acquiring expert annotations is both expensive and time-consuming. To overcome this, we utilize GPT language models, known for their large-scale knowledge and flexibility [202]. Our approach involves using GPT-4 to generate visual descriptions for any given dataset thereby aiding in the construction of prompt ensembles for CLIP in a scalable manner.

Our prompting strategy takes inspiration from chain-of-thought prompting [187] and is as follows: First, we ask GPT-4 to list all the attributes that may be necessary to discriminate between images of the K classes under consideration. Second, we ask GPT-4 to provide the values for all these attributes for all the K classes as sentences. An example for the CUB dataset is shown in the left side of Fig 4.1.

The last row in Table 4.1 shows that the GPT-4 generated visual sentences' performance is similar to that of sentences generated from the class attribute vectors annotated by domain experts. We follow the same simple strategy for all the datasets in the benchmark suite to generate visually descriptive sentences in a scalable and flexible manner and use them to construct prompt ensembles.

4.3.3 Prompts for GPT-4

Throughout our experiments, we use a two-step prompting strategy in which we first ask the LLM to generate a list of attributes which will aid in visually distinguishing between the different classes in a particular dataset. The second prompt asks the LLM to create a description using the attributes provided by the first prompt and specifies the expected output format. We request a python dictionary as output with a list of sentences, each corresponding to one attribute. The output structure is

simple to use (downstream), preserves attribute-level detail and encourages attribute richness.

Example of first prompt for the FGVC Aircraft dataset:

> *I am creating class attributes for a zero-shot image recognition algorithm to classify different images of a diverse set of aircrafts. The attributes are part of side information about the classes. List 20 attributes that can form part of a description of the class that will aid in distinguishing between the following list of classes visually: [list of classes in dataset e.g. 'Boeing 707-320', 'Boeing 727-200',..., 'Yakovlev Yak-42']*

GPT-4 response:

> *Manufacturer: The company that produces the aircraft (e.g., Boeing, Airbus). Model Number: Differentiates different aircraft within the same manufacturer. Aircraft Function: The primary role or use of the aircraft, such as commercial (passenger), cargo (freight transport), or military (fighter, bomber, etc). Engine Count: The number of engines on the aircraft. Engine Position: The location of the engines, either under the wings or at the rear of the fuselage. Wing Position: The position of the wings: high-wing, mid-wing, or low-wing design. Tail Type: The shape and type of the tail, such as traditional, T-tail, V-tail, etc. Body Length: The length of the aircraft's fuselage. Body Shape: The overall shape of the fuselage, like narrow, wide, double-decker, etc. Wing Shape: The shape of the wings, either swept or straight. Nose Shape: The shape of the nose or front of the aircraft. Landing Gear Type: The type of landing gear, like tricycle or taildragger. Cockpit Window Shape: The shape and number of cockpit windows. Color Scheme: Typical color schemes or liveries associated with the aircraft type. Number of Decks: Either single-deck or double-deck (like the Airbus A380). Existence of Winglets: Whether or not the aircraft has winglets at the end of the wings. Existence of Canards: Whether or not the aircraft has canards, small wing-like surfaces located at the front. Propeller or Jet: Indicates if the aircraft is propeller-driven or jet-powered. Number of Seats: An estimate of the seating capacity of the aircraft. Range: The maximum distance the aircraft can travel without refueling. Unique Visual Identifier: Specific structures or features that are unique to a model or series of aircraft. Aircraft Type: Differentiates between commercial, cargo, or military aircraft.*

Example of the second prompt for the FGVC Aircraft dataset which uses GPT-4's response for the first prompt as part of the prompt:

> *Describe the following aircraft by adding one sentence about each attribute for the following aircraft: Airbus A340-200. Return the answer as a python dictionary with the aircraft variant as the key (i.e. remove the manufacturer) and the value is a list of sentences. Rewrite the attribute as a full sentence. Do not include the attributes as keys. Attributes: Manufacturer: The company that produces the aircraft (e.g., Boeing, Airbus). Model Number: Differentiates different aircraft within the same manufacturer. Aircraft Function: The primary role or use of the aircraft, such as commercial (passenger), cargo (freight transport), or military (fighter, bomber, etc). Engine Count: The number of engines on the aircraft. Engine Position: The location of the engines, either under the wings or at the rear of the fuselage. Wing Position: The position of the wings: high-wing, mid-wing, or low-wing design. Tail Type: The shape and type of the tail, such as traditional, T-tail, V-tail, etc. Body Length: The length of the aircraft's fuselage. Body Shape: The overall shape of the fuselage, like narrow, wide, double-decker, etc. Wing Shape: The shape of the wings, either swept or straight. Nose Shape: The shape of the nose or front of the aircraft. Landing Gear Type: The type of landing gear, like tricycle or taildragger. Cockpit Window Shape: The shape and number of cockpit windows. Color Scheme: Typical color schemes or liveries associated with the aircraft type. Number of Decks: Either single-deck or double-deck (like the Airbus A380). Existence of Winglets: Whether or not the aircraft has winglets at the end of the wings. Existence of Canards: Whether or not the aircraft has canards, small wing-like surfaces located at the front. Propeller or Jet: Indicates if the aircraft is propeller-driven or jet-powered. Number of Seats: An estimate of the seating capacity of the aircraft. Range: The maximum distance the aircraft can travel without refueling. Unique Visual Identifier: Specific structures or features that are unique to a model or series of aircraft. Aircraft Type: Differentiates between commercial, cargo, or military aircraft.*

The response of the second prompt constitutes the VDT information we utilise as side-information for Airbus A340-200 as an example:

> *"A340-200": ["The Airbus A340-200 is produced by Airbus, a renowned aircraft manufacturer.", "It differentiates itself from other aircraft within*

the Airbus family through its unique model number: A340-200.", "This aircraft primarily serves a commercial role, typically used for passenger transport.", "The Airbus A340-200 is equipped with four engines.", "These engines are situated under the aircraft's wings.", "The aircraft features a low-wing design, with wings positioned at the bottom of the fuselage.", "It has a traditional tail configuration, common to many large commercial aircraft.", "The A340-200 has a lengthy fuselage, extending to about 59.4 meters.", "The body of the Airbus A340-200 is wide-bodied, facilitating a larger passenger capacity.", "Its wings are swept back, a design aspect that improves fuel efficiency and performance at high speeds.", "The aircraft features a rounded nose shape, contributing to its aerodynamic design.", "The Airbus A340-200 uses a tricycle type landing gear, supporting stability during takeoffs and landings.", "Its cockpit windows are angular and include six panels, giving pilots a comprehensive view of their surroundings.", "Color schemes vary by airline, but the Airbus corporate livery features a predominantly white body with blue and teal accents.", "This model is a single-deck aircraft, focusing on width rather than height for passenger capacity.", "The A340-200 does not have winglets, differing from some newer Airbus models.", "There are no canards present on the Airbus A340-200, instead, it employs a more traditional aircraft design.", "As a jet-powered aircraft, the A340-200 uses high-speed jet engines for propulsion.", "The A340-200 typically accommodates around 260 passengers, though the exact number can vary with the configuration.", "With a range of approximately 7,800 nautical miles, the Airbus A340-200 can cover considerable distances without refueling.", "The aircraft's four-engine configuration and lengthy, wide-bodied design are unique visual identifiers of the A340-200 model.", "Classified as a commercial aircraft, the Airbus A340-200 is primarily used for passenger transportation."]

GPT-4 generally adheres to the python dictionary output requirement in the *User* prompt, but tends to return additional explanations, motivations or clarifications. To encourage the LLM to only return a Python dictionary as requested, we add the following *System* prompt:

You are ChatGPT, a large language model trained by OpenAI. Return only the python dictionary, with no explanation.

Conversely, OpenAssistant's [73] output requires manual cleaning and reformatting to get into Python dictionary format. GPT-3.5 performed slightly worse than GPT-4 in terms of adherence to the prompt, as it did not consistently return only a

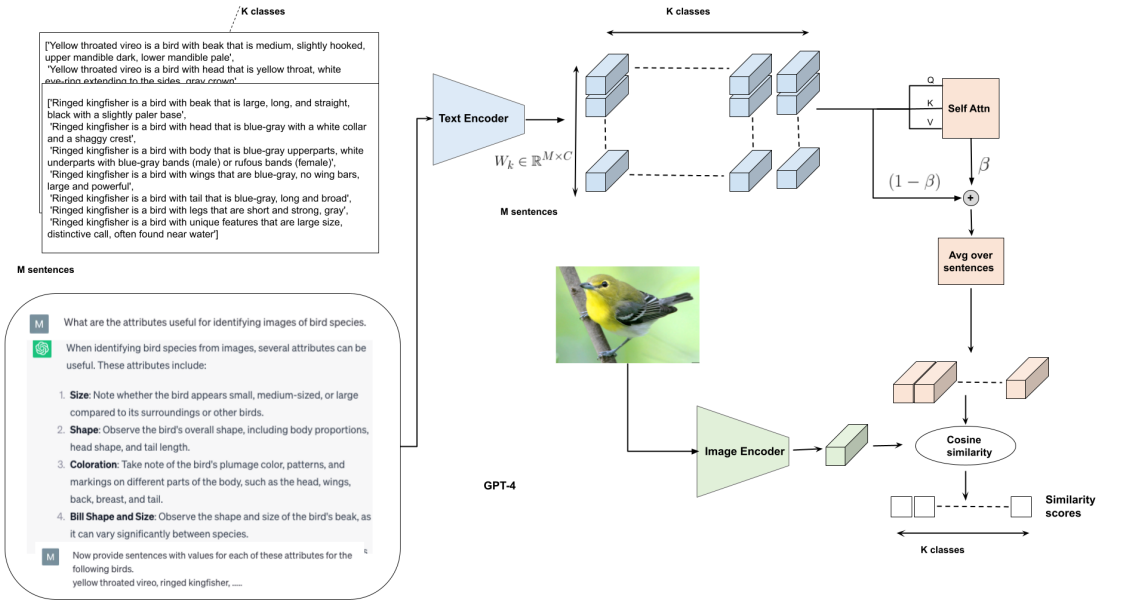


Figure 4.2: CLIP-A-self, our simple self-attention based adapter learns to select and aggregate the most relevant subset of Visually Descriptive Text (VDT) to generate more generalizable classifiers. First, we prompt GPT-4 to generate VDT, N sentences for K classes that are then passed through the text encoder to get embeddings for each of the N*K sentences. Self-attention is applied over the N sentences of each class and averaged to get K adapted classifier embeddings.

dictionary. In such cases, we simply called the API again. After repeated incorrect format responses, we manually cleaned those cases.

We primarily utilized GPT-4 via the ChatGPT Plus subscription plan at a cost of $20 since the GPT-4 API was not generally available during most of our experimentation phase. The GPT-4 API cost to create the VDT information for the SUN397 dataset was $14.90, as opposed to $1.94 using the GPT-3.5 API.

4.3.4 Simple few-shot adapters for visual sentences

We design a simple adapter that can use VDT information to improve the few-shot transfer of CLIP to the target datasets. Similar to the CLIP-A text, we append a small set of learnable parameters to the output of the word encoder and train the adapter using cross-entropy loss. Our CLIP-A-self uses a self-attention layer that applies attention over the embeddings of the different sentences for each class and averages the output to get the final classification vector.

Given we have M GPT generated sentences for each of the K classes $t_{m,k}$, we construct M prompts by appending each sentence to the prompt template like $H(classname(y_{i,k}), \{t_{m,k}\})$ and pass them through CLIP's word encoder to get $W^{sent} \in \mathbb{R}^{D \times M \times K}$.

For the self-attention adapter, we apply vanilla self-attention [178] over all the visual descriptive sentences such that during training it learns to select and aggregate the most relevant visual sentences for identifying each class. Just like before, we first obtain the classification vector for all sentences $W^s \in \mathbb{R}^{K\times M\times D}$ and pass them as the key, query, and value to the self-attention module B_{self} and average out the output tokens to get the final classification vector $W^\star$. Here the attention is applied over the M different visually descriptive sentences.

$$W_{avg} = 1/M \sum_{m=1}^{M} W^s_{m,k} \tag{4.4}$$

$$\left\{W^a_{m,k}\right\}_1^M = B_{self}(\left\{W^s_{m,k}\right\}_1^M, \left\{W^s_{m,k}\right\}_1^M, \left\{W^s_{m,k}\right\}_1^M) \tag{4.5}$$

$$W_{a-mean} = 1/M \sum_{m=1}^{M} W^a_{m,k} \tag{4.6}$$

$$W^\star = \beta \mathbf{W_{a-mean}}^T + (1-\beta)\mathbf{W_{avg}} \tag{4.7}$$

We finally obtain the new adapter classifier weights $W^\star \in \mathbb{R}^{D\times K}$ that have been adapted to focus on the most visually discriminative information among the M visually descriptive sentences for any given dataset. We make use of 4.1 to calculate the probabilities and predict the image category by selecting the class with the highest probability.

During the few-shot training only the weights of the adapter network B_{self} are trained using cross-entropy loss.

4.4 Experiments

We assess the significance of visual sentence ensembles in two scenarios: (i) we gauge visual sentence quality by comparing an ensemble of these prompts with CLIP's default prompts across 12 benchmark datasets; (ii) we contrast the performance of adapters using these visual prompts against other few-shot transfer techniques in Base-to-New class generalization within a dataset. Prior to discussing the results, we detail the datasets and experimental setup.

4.4.1 Datasets

We use 11 diverse image recognition datasets from [216] and the bird species CUB dataset [184] for both study settings, extending our suite to 12. These include generic object datasets ImageNet [34] and Caltech101 [47]; fine-grained classification datasets OxfordPets [126], StanfordCars [76], Flowers102 [117], Food101 [14] and FGVCAircraft [98]; SUN397 [193] for scene recognition; UCF101 [163] for action

recognition; DTD [26] for texture classification; EuroSAT [62] for satellite imagery; and CUB for bird identification.

For 0-shot transfer with visual sentences, we test on *All* classes across these datasets while for the Base-to-New setting, following [215], we equally sample classes for base and new sets without overlap. We use the 150-base and 50-new class split from ZSL and few-shot literature [192, 101] for CUB. Like [215], our CLIP-A-self is evaluated on the 16-shot setting for easier comparison with other methods.

4.4.2 Baselines

We compare the performance of visual sentences ensemble on 0-shot transfer against the CLIP model [132] whose default prompts for each dataset have been extensively fine-tuned using a test set. We also compare against DCLIP [104] a recent work that uses GPT-3 to generate VDT information for 0-shot transfer. We compare our CLIP-A-self against two prompt learning methods CoOp [216] which learns static prompts and CoCoOp [215] which learns a dynamic prompt that is specifically designed to improve Base-to-New transfer. We also compare our CLIP-A-self against CLIP-A [53] due to the similarity in architecture and to show that the performance improvements are from making use of the visual sentences and not from the just adapting the text features.

4.4.3 Training settings

Our implementation is based on CoOp's and CLIP-A's code. [1] We make all our comparisons on VIT CLIP backbone i.e., VIT-B/16. We take the results for CoOp and CoCoOp for all datasets (except CUB) from their respective papers, while we make use of practices from the respective papers like context length set to 4 and context initialization to "a photo of" to ensure the best results on the CUB dataset. For CLIP-A, we re-run all experiments on VIT-B/16 backbone as they were not reported in the paper. For all adapter models including ours, we only tune the residual ratio β hyper-parameter. For CLIP-A, we use the version where the MLP is applied on top of the visual encoder as it performed the best [53]. We make use of May version of GPT-4 for obtaining the auxiliary dataset.

[1] `https://github.com/KaiyangZhou/CoOp`, `https://github.com/gaopengcuhk/CLIP-Adapter`

Table 4.2: Results of including LLM generated VDT on 6 datasets for comparison with other works. We see that higher quality VDT from GPT-4 outperforms GPT-3 generated VDT on specialized datasets like DTD OxfordPets and EuroSAT.

Methods	EuroSAT	Food101	DTD	OxfordPets	CUB	ImageNet	Average
CLIP	47.69	85.97	43.09	89.07	54.70	64.51	64.17
DCLIP[104]	48.82	**88.50**	45.59	86.92	**57.75**	68.03	65.93
CLIP-GPT	**54.86**	86.43	**50.15**	**91.54**	57.43	**68.92**	**68.21**

Table 4.3: Results of 12 datasets with ViT-B/16 for 0-shot domain transfer.

Methods	EuroSAT	Caltech101	OxfordFlowers	Food101	FGVCAircraft	DTD	OxfordPets	StanfordCars	Sun397	UCF101	CUB	ImageNet	Average
CLIP	47.69	93.75	70.69	85.97	**24.81**	43.09	89.07	**65.55**	62.61	**67.54**	54.70	64.51	64.16
CLIP-GPT	**54.86**	**94.51**	**73.40**	**86.43**	23.42	**50.15**	**91.54**	65.01	**67.24**	65.51	**57.43**	**68.9**	**66.53**

4.4.4 GPT generated visual sentences improve 0-shot transfer.

We compare the performance of CLIP-GPT prompt ensemble with the default prompts of CLIP in Table 4.3. GPT-generated prompt ensemble improves upon the performance of CLIP 0-shot by 2% on average over 12 datasets. The improvement over CLIP-ZS is significant; over 5% for specialized fine-grained datasets like CUB, SUN397, EuroSAT, and DTD and over 2% for oxford-flowers and oxford-pets. This shows that CLIP does not recognize several of the classnames in these datasets and describing the class in the form of visually descriptive sentences results in better classifiers from the text-encoder and better classification accuracy. It is also worth noting that only including the visually descriptive sentences in the prompts can help improve the performance of general datasets like Imagenet (over 4%) and Caltech-101 (over 1%) too. For all other datasets, the transfer performance matches that of CLIP, with the exception being the action recognition dataset UCF-101. We inspected the sentences generated for UCF-101 and notice that several of the sentences generated by GPT involves temporal information instead of visual descriptions and we believe this could be the reason for the drop in accuracy. However, we notice in Section 4.4.5 that the self-attention module of the few-shot adapter learns to emphasize the visual sentences out of the generated sentences which might explain the improvement in the performance of few-shot adapters in the new setting in Section 4.4.5. We also compare against recent work [104] on their subset of 6 datasets for VIT-B/16 encoder in 4.2. We see that using the larger GPT-4 model over the GPT-3 model results in much higher improvements for specialized datasets like DTD ($\sim 5\%$) and EuroSAT ($\sim 6\%$). We compare the text used by [104] against our GPT4-generated VDT in Table A.1.

	Base	New	H
CLIP	68.45	73.89	71.05
CoOp	**82.39**	62.39	70.99
CoCoOp	79.35	71.89	75.37
CLIP-A	78.90	72.14	75.07
CLIP-A-self	82.12	74.20	**77.78**

Table 4.4: Comparing our CLIP-A-self against other methods on average accuracy over 12 datasets.

4.4.5 GPT-Adapters improve few-shot transfer performance.

We compare the performance of our CLIP-A-self against CLIP, CoOp, and CoCoOp on the benchmark suite of 12 datasets in the Base-to-New setting in Table 4.5 and Table 4.4. Here we see that GPT-Adapters that make use of the VDT information outperform CoCoOp by 3% in the new setting while maintaining similar performance to that of CoOp in the base setting on the average accuracy over 12 datasets. This is impressive considering that CoCoOp makes use of a meta-network and forward pass through the text encoder making it computationally intensive to train. CoCoOp takes up to 5 hours to train on 16-shot ImageNet for VIT-B/16 encoder, in comparison, our CLIP-A-self takes only 10 mins (on an RTX 3090 GPU). The Base-to-New generalization ability of our adapters is even more impressive for fine-grained, specialized datasets as evidenced by the gains over CoCoOp in Harmonic mean of base and new accuracy. For example, CLIP-A-self demonstrates gains in datasets like FGVCAircraft (7.5%), EuroSat (7.4%), DTD (5.8%), CUB (4.3%), Flowers102 (4%), Stanford Cars (2.4%) and UCF-101 (2.4%). This demonstrates that our adapters make use of semantic information in the form of visually descriptive sentences and fuse this with CLIP's 0-shot knowledge to build more generalizable classifiers that transfer well to unseen classes within the same dataset. It is also worth noting that even though the same set of VDT did not provide any improvements in 0-shot domain transfer for datasets like FGVC-Aircraft, Stanford-Cars, and UCF-101, our self-attention adapter was able to choose the most informative subset of VDT and produce few-shot classifiers that provide substantial few-shot transfer performance gains in comparison to CoCoOp. We show in Section 4.4.5 the sentences picked by the attention mechanism for these datasets to qualitatively verify this.

Attention weights Analysis

We visualize the attention weights learned by the CLIP-A-self for datasets Stanford Cars, UCF101, FGVC Aircraft, Oxford Flowers and CUB in Table 4.6. We note

Table 4.5: **Comparison of GPT-Adapters with CLIP, CoOp and CoCoOp in the Base-to-New generalization setting**. For prompt learning-based methods (CoOp and CoCoOp), their prompts are learned from the base classes (16 shots). The results strongly justify the importance of including extra visual information. H denotes Harmonic mean (to highlight the generalization trade-off [192]). Here C-A is CLIP-A and C-A-self is CLIP-A-self.

(a) CUB.

	B	N	H
CLIP	58.7	70.3	63.90
CoOp	79.2	53.3	63.71
CoCoOp	67.1	74.1	70.40
C-A	68.3	70.8	69.53
C-A-self	**78.6**	**71.3**	**74.77**

(b) Caltech101.

	B	N	H
CLIP	96.84	94.00	95.40
CoOp	98.00	89.81	93.73
CoCoOp	97.96	93.81	95.84
C-A	97.7	93.6	95.61
C-A-self	**98.3**	**95.9**	**97.09**

(c) OxfordPets.

	B	N	H
CLIP	91.17	97.26	94.12
CoOp	93.67	95.29	94.47
CoCoOp	**95.20**	**97.69**	**96.43**
C-A	94.8	97.0	95.89
C-A-self	94.4	97.0	95.68

(d) StanfordCars.

	B	N	H
CLIP	63.37	**74.89**	68.65
CoOp	**78.12**	60.40	68.13
CoCoOp	70.49	73.59	72.01
C-A	70.5	73.3	71.87
C-A-self	76.8	72.9	**74.80**

(e) Flowers102.

	B	N	H
CLIP	72.08	**77.80**	74.83
CoOp	**97.60**	59.67	74.06
CoCoOp	94.87	71.75	81.71
C-A	94.6	71.5	81.44
C-A-self	97.4	75.3	**84.94**

(f) Food101.

	B	N	H
CLIP	90.10	91.22	90.66
CoOp	88.33	82.26	85.19
CoCoOp	**90.70**	**91.29**	**90.99**
C-A	90.3	91.2	90.75
C-A-self	90.4	91.2	90.80

(g) FGVCAircraft.

	B	N	H
CLIP	27.19	**36.29**	31.09
CoOp	**40.44**	22.30	28.75
CoCoOp	33.41	23.71	27.74
C-A	34.9	33.5	34.19
C-A-self	37.8	33.0	**35.24**

(h) SUN397.

	B	N	H
CLIP	69.36	75.35	72.23
CoOp	**80.60**	65.89	72.51
CoCoOp	79.74	76.86	78.27
C-A	80.1	75.9	77.94
C-A-self	81.4	76.8	79.03

(i) DTD.

	B	N	H
CLIP	53.24	59.90	56.37
CoOp	79.44	41.18	54.24
CoCoOp	77.01	56.00	64.85
C-A	74.9	53.0	62.08
C-A-self	**81.8**	**62.3**	**70.73**

(j) EuroSAT.

	B	N	H
CLIP	56.48	64.05	60.03
CoOp	**92.19**	54.74	68.69
CoCoOp	87.49	60.04	71.21
C-A	82.5	62.4	71.06
C-A-self	88.5	70.5	78.48

(k) UCF101.

	B	N	H
CLIP	70.53	**77.50**	73.85
CoOp	**84.69**	56.05	67.46
CoCoOp	82.33	73.45	77.64
C-A	82.9	74.9	78.70
C-A-self	84.1	76.4	**80.07**

(l) ImageNet.

	B	N	H
CLIP	72.43	68.14	70.22
CoOp	**76.47**	67.88	71.92
CoCoOp	75.98	**70.43**	**73.10**
C-A	75.4	68.6	71.84
C-A-self	76.4	68.3	72.12

that even though CLIP-GPT ensembles were outperformed by CLIP default prompt on FGVC Aircraft, UCF-101, and Stanford Cars dataset, we see that CLIP-A-self outperforms CLIP-A and CoCoOp [215] on these datasets in the few-shot transfer setting. We believe that this is because, during few-shot training, the self-attention mechanism learns to select the most relevant visual sentences out of the set of visually descriptive text and helps produce generalizable classifiers. In Table 4.6 we show the top 3 and bottom 3 attributes picked by attention scores for each of these datasets and show that the sentences with the highest attention scores correspond to visually descriptive attributes in the set and vice versa for the lowest scored attributes. For example, for both Stanford Cars and FGVC it is interesting to see that the color

Dataset	Top 3 attributes selected	Bottom 3 attributes selected
FGVC	Unique visual identifier, presence of canards, tail type.	Color scheme, model number, commercial or cargo.
Cars	Body shape, fender description, spoiler description,	Interior description, brand logo description, color scheme
UCF-101	Equipment used, Posture of person, Interaction info.	Body muscles used, force involved, speed of motion
Oxford-Flowers	Shape of the flower, Color, shape and number of petals, Texture and description of veins in leaves	Stem color, Color of leaves, Description of sepals
CUB	Wings color and shape, Head color and shape, Beak color and shape	Color and description of legs, Underparts color, Tail shape and color

Table 4.6: The top 3 and bottom 3 attributes selected by the attention mechanism in GPT-A-self for 3 different datasets. For UCF101, We see that attention learns to pick visually descriptive sentences like posture and description of objects over temporal information like speed of motion and force applied.

scheme is one of the least used attributes as it's difficult to identify a car or a plane from its color or livery. For UCF-101, information like the force involved or temporal information like speed and range of motion of the action is unlikely to be encoded in the image and hence is not selected by the attention mechanism. Information regarding the subject and the object of the action, like the posture of the person, description of the object, and interaction between objects are visible in the images and hence weighted highly by the attention mechanism. We notice that the self-attention mechanism in CLIP-A-self assigns more weight to visually descriptive sentences that are most relevant for discriminating between the classes of the dataset under consideration. For instance, we see that for discriminating images of birds species (CUB dataset) and flower species (Oxford Flowers) sentences describing the color of the head and wings of birds and petals of the flowers are important but for identifying different car or aircraft models sentences describing the color or livery is one of the least important. We also see that if the information being described by the VDT sentence is not clearly visible in the image, the attention weight assigned to it by CLIP-A-self is low. For instance, in the CUB dataset, the undersides of birds or the sepals in the Oxford Flowers dataset are often not visible in the images, hence the VDT sentence corresponding to this is in the bottom 3 attributes picked by the learnt attention weights. It's also worth noting that, some of the VDT sentences do not have much variation between different classes and hence are not useful in discrimination between the classes of the dataset. For instance, in Oxford-flowers, the color of the leaves, and the color of the stem are often green for most flowers in

Prompting	ZS	Base	New	H
Default	54.7	NA	NA	NA
OpenAssistant	56.0	78.3	69.8	73.80
GPT-3.5	55.7	78.1	70.6	74.16
GPT-4	57.4	78.6	71.3	74.77

Table 4.7: Comparing different GPT models for obtaining the VDT information. We see that the larger models provide higher quality VDT information but CLIP-A-self is capable of producing generalizable classifiers even with smaller models like OpenAssistant.

the dataset, which maybe why a low attention score was learnt for this attribute.

4.4.6 Ablation over different GPT models

In this section, we see if other GPT models like GPT-3.5 and open-source model, OpenAssistant [73], are as capable as GPT-4 in generating visually descriptive information. We explore this on the CUB dataset as it is fine-grained and specialized. The results are presented in Table 4.7. We find that the performance improves with larger models which are more capable of memorizing accurate class information with less hallucination [202]. Even though we obtain decent performance with the open-source model OpenAssistant, the outputs were always inconsistent and noisy, resulting in a lot of clean-up effort in comparison to GPT-3.5 and GPT-4 where the outputs were in the form of concise sentences following a dictionary format. It is worth noting that our few-shot adapter is capable of picking out the the best VDT information even from a noisy set, pushing the Base-to-New generalization performance of OpenAssistant, and GPT-3.5 close to that of GPT-4.

Comparing our VDT with GPT3

In Table A.1, we compare the VDT generated by GPT-4 using our prompting technique with that of [104] who used GPT-3 to obtain visual descriptors for different classes of the dataset. Here we notice that including a prompt step asking the GPT-4 for visual attributes necessary for classifying between images of the classes results in a fixed number of sentences per class, a fixed order guaranteeing that every class is accompanied by as much visual information as possible. By using GPT-4 we also get much richer and more accurate visual descriptions. For example, for the class industrial, our descriptions provide information about the density of buildings, shadows in the image, road accessibility, and layout while the description used by [104] is only 'evidence of human activity'. A similar phenomenon can be observed for the DTD dataset. This explains the jump in performance for specialized datasets like DTD and Eurosat over DCLIP.

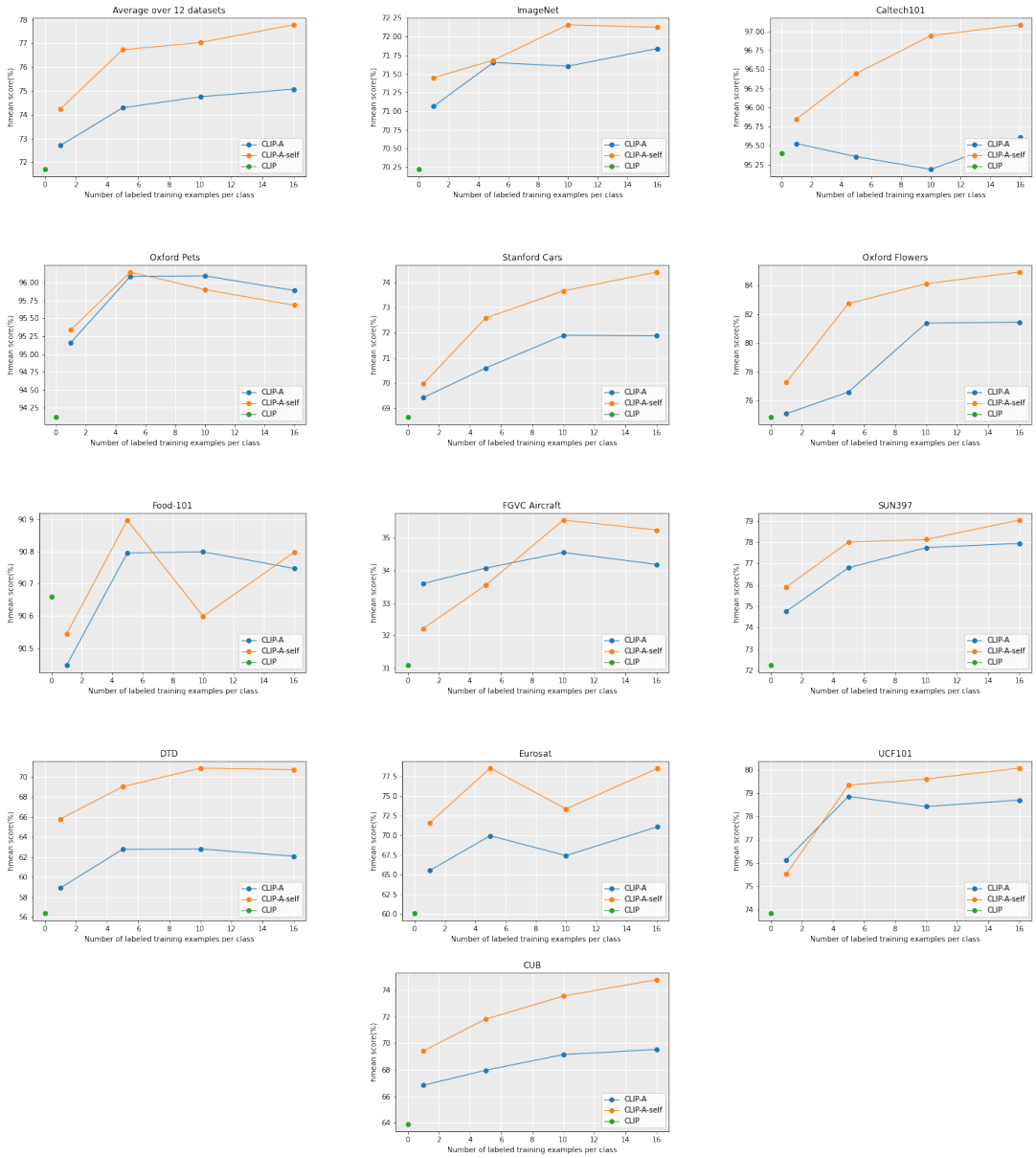


Figure 4.3: Main results of Base-to-New few shot learning on 12 datasets. CLIP-A-self consistently shows better performance over CLIP-A over different training shots, demonstrating the importance of Visually descriptive text in improving the generalizability of few-shot classifiers for CLIP.

4.4.7 Generalizability at lower shots

In Figure 4.3, we compare the harmonic mean of Base and New accuracies of CLIP-A-self with that of CLIP-A over number of shots = 1, 5, 10, 16. Our CLIP-A-self demonstrates performance improvements at lower shots, outperforming CLIP-A on average by over 1.5%/ for the 1-shot case and over 2.5%/ for the 5-shot case. Our adapter shows higher improvements over CLIP-A in the higher shot scenario because of the number of parameters and the inherent difficulty in identifying the VDT sentences that are discriminative for the current classes in the low shot scenario. For instance, identifying the class from a single image is often difficult because

of co-occuring objects, environment, background etc which can be resolved if we have more exmaple images from the same class. The largest improvements are for specialized and fine-grained datasets like Stanford-Cars, EuroSat Oxford Flowers, DTD and CUB. Oxford-pets and Food-101 results do not improve much because these datasets are relatively easy and already show good performance with default CLIP.

4.5 Conclusion

In this chapter, we show that using visually descriptive textual (VDT) information can improve the 0-shot domain transfer performance of CLIP over non-visual semantic information and the default prompts. We leverage the recent advances in LLMs and their ability to act as approximate world models for factual information retrieval in a flexible, scalable manner to provide an answer to **RQ2** *How can LLMs be used to generate semantic information useful for computer vision tasks in a scalable manner?* More specifically, we demonstrate that GPT-4 is an accurate and flexible source of VDT information by improving the 0-shot domain transfer performances on a suite of 12 benchmark datasets showcasing its potential to generate scalable semantic information adaptable to any real-world dataset. Our few-shot adapter CLIP-A-self learns to pick the best VDT information from the LLM-generated set and improves the few-shot domain transfer in the Base-to-New setting even when the quality of the generated text deteriorates with smaller LLMs. This provides a simple but effective answer to our **RQ3** *How can we use this semantic information to improve the 0-shot and few-shot transfer performance of vision language foundation models?* As part of this work, we released all prompts and VDT information for the 12 benchmark datasets to promote further research in the research direction of using LLMs for learning multi-modal adapters for vision language foundation models.

In Chapter 3 and this Chapter we make use of semantic information to improve the label efficiency and generalizability of vision models. Specifically in this Chapter, we use an LLM to explain novel concepts in terms of concepts that CLIP's text-encoder already understands. This is sub-optimal in terms of concept-coverage, multi-linguality and long-context primarily due to the CLIP text-encoder's sub-par unimodal features. This can be attributed to the contrastive pre-training on noisy image, caption pairs, where the text data has limited coverage. Similarly, CLIP's vision encoder lacks the fine-grained understanding to perform well on image-localization tasks due to the global pooling of local vision features during pre-training. Another limitation of CLIP is the significant compute/data requirements for training, as both image, and language encoders are trained from scratch on over 400 million image, caption pairs. These challenges faced by the CLIP model could

be alleviated if strong unimodal vision and language encoders can be aligned using simple projection transformations. To develop such a framework, we first study the semantic similarities of well-trained vision and language representations in the next chapter. This is grounded in the idea, that since vision and language encoders are modeling the same physical reality, then the representations from their encoders should demonstrate high similarity in terms of semantics.

Chapter 5

Do Vision and Language Encoders Represent the World Similarly?

In Chapters 3 and 4, we demonstrate how semantic knowledge from language can be harnessed in a retrieval-augmented manner to enhance the performance and generalizability of vision models. In this chapter, we dive deeper into a more fundamental research question: if language models encode semantics that are useful for vision tasks, should there not be a significant similarity between the representational spaces of vision and language models? Since both vision and language encoders represent the same physical world through different modalities, it raises the key question: How similar are the representations of these two modalities? (**RQ4**).

We define semantic similarity as the correspondence between semantic concepts in the embedding spaces of vision and language encoders and analyze the latent structure of these spaces using image-caption benchmarks and the Centered Kernel Alignment (CKA) metric. Surprisingly, we find that the representation spaces of unaligned and aligned encoders are semantically similar.

In the second half of this chapter, we address a related question (**RQ5**): *Is there a way to connect semantically similar vision and language representations in a training-free manner?* We propose leveraging semantic similarity between encoders by framing the alignment as a seeded graph-matching problem. To solve this, we introduce two methods: a Fast Quadratic Assignment Problem optimization and a novel localized CKA metric-based matching/retrieval approach. We validate these methods across several downstream tasks, including cross-lingual and cross-domain caption matching, as well as image classification. The code for this work is available at github.com/mayug/0-shot-llm-vision. This work was presented as a conference paper at the Computer Vision and Pattern Recognition (CVPR) conference held in Seattle, USA in June 2024.

The following section presents an overview of aligned vision language models like CLIP, advances in vision and language unimodal encoders and our approach

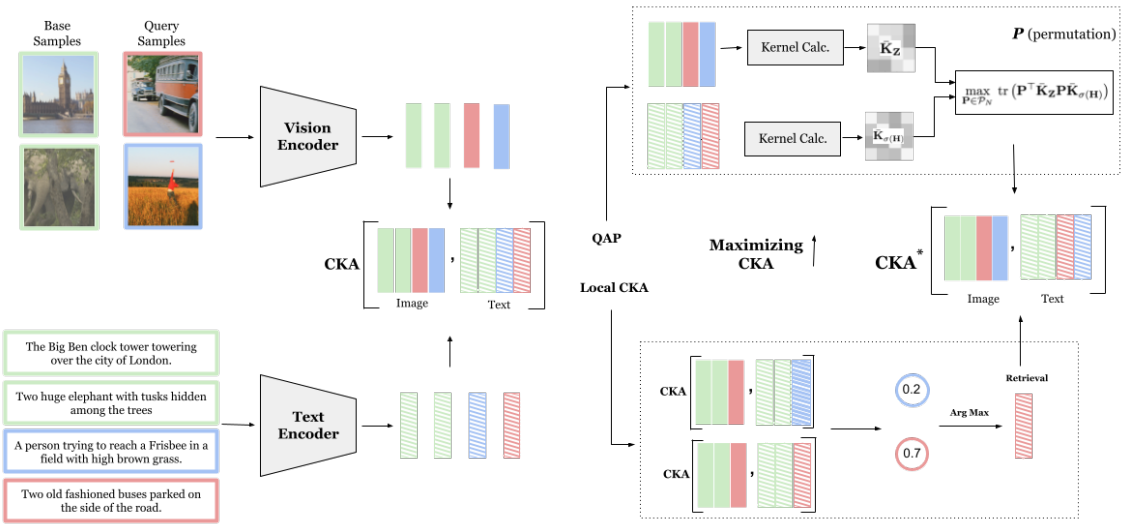


Figure 5.1: For matching, we calculate the kernels for image and text embeddings and employ QAP-based seeded matching to maximize CKA for obtaining the optimal permutation P. For retrieval, we append query embeddings to base embeddings and retrieve the best caption that maximizes the local CKA for a query image.

to investigating and exploiting the semantic similarity between unaligned encoder spaces. This is followed by 5.2 detailing a literature review on semantic similarity methods and other methods that try to connect unaligned latent spaces in a training free manner. In Section 5.3 we go over the preliminaries of CKA while in Section 5.4 we go over the 2 proposed methods used to acquire an alignment between unaligned encoder spaces. In Section 5.5 we detail our experiments, evaluation tasks and show a practical application of our method through multi-lingual image-caption retrieval task.

5.1 Introduction

The recent success of deep learning on vision-language tasks mainly relies on jointly trained language and image encoders following the success of CLIP and ALIGN [68, 132]. The standard procedure for training these models aims at aligning text and image representation using a contrastive loss that maximizes the similarity between image-text pairs while pushing negative captions away [58, 120, 21]. This achieves a statistical similarity across the two latent spaces, which is key to retrieving the closest cross-modal representations using cosine similarity. This property is not valid for unaligned encoders, hence, extra transformations are needed to bridge the gap. These transformations can be training a mapping network that captures the prior distribution over the text and image representations [106, 110, 118]. The work of [106] has shown that it is possible to train a linear mapping from the output embeddings of vision encoders to the input embeddings of language models and exhibit impressive performance on image captioning and VQA tasks. This indicates

that the representations between the unaligned uni-modal vision and language encoders are sufficiently high level and differ only by a linear transformation. However, this linear layer is trained on CC-3M [18] consisting of three million image-caption pairs.

Is this training step necessary? In an ideal scenario, we anticipate an alignment between vision and language encoders as they inherently capture representations of the same physical world. To this end, we employ Centered Kernel Alignment (CKA) [135, 28, 75], which is known for measuring representation similarity both within and between networks. As shown in Figure 5.2, we measure the CKA between a variety of unaligned vision and language encoders [39, 188, 92, 121, 17], on the image-caption pairs of the COCO [89] dataset and observe that some have comparable scores to that of aligned encoders like CLIP [132], affirmative of semantic similarities.

We then ask the question: If the unaligned image and text encoders are semantically similar, is there a way to connect them in a *zero-shot manner?* Do they build a similar representation graph over the same information coming from the two modalities? We study these questions, revealing key similarities between unaligned image and text encoders, and how these similarities can be exploited for downstream tasks. Furthermore, we devise a caption matching downstream task and show using two novel methods that latent space communication between unaligned encoders could be achieved by leveraging the semantic similarities between the cross-modal spaces. Our contributions are:

- We present a matching method that seeks to find the permutation of the captions that maximizes the CKA (see Fig. 5.1). Hence, We formulate maximizing CKA as a quadratic assignment problem and introduce transformations and normalizations that greatly improve the matching performance.

- We propose a local CKA metric and use it to perform retrieval between two unaligned embedding spaces, demonstrating superior performance with that of relative representations [110] on the COCO caption image retrieval.

- The method is benchmarked on COCO, NoCaps [5] cross-domain caption and image retrieval as well ImageNet-100 [33] classification tasks despite our method not being optimized to align the representation in any manner demonstrating zero-shot communication between the encoder's latent spaces.

- Finally, we show a practical application of our method on cross-lingual image retrieval by making use of sentence transformers trained in various languages and a CLIP vision encoder trained only in English.

5.2 Related Work

Recently, there has been an increasing consensus that good networks, when trained independently, learn general representations across different architectures and tasks. On the one hand, the works of [109, 87, 75, 13] show that these networks exhibit representation similarity by learning similar latent spaces when trained on similar tasks and data [174, 10, 183, 27, 83, 107, 7]. Specifically, [75] introduced centered kernel alignment (CKA) as a similarity metric for comparing the inner representations across networks. The CKA measure mitigates the limitation of canonical correlation analysis (CCA) [134] being invariant to an invertible linear transformation that often leads to difficulty in measuring meaningful similarities between representations. [189] uses CKA for comparing the representations from different layers of different language models and the effect of downstream task-finetuning on the representation similarities, while [13] utilizes CKA along with Procrustes similarity for understanding the ability of variational autoencoders (VAEs) [72] in learning disentangled representations. In general, these approaches study the representation similarity in unimodal models, either vision or language. Clearly, however, the use of CKA has been limited to visualization and analysis purposes, whereas we attempt at exploiting CKA as an optimization objective.

Recent works [110, 118] employ relative representations to match embeddings of unaligned encoders using the cosine similarity to a set of anchors. However, these relative representations are sensitive to the selection of anchors and noise in the original embeddings. Similarly, approaches [9, 31] analyze networks and empirically verify the “good networks learn similar representations" hypothesis by utilizing model stitching [83], which introduces trainable stitching layers to enable swapping parts of different networks. LiMBeR [106] can be seen as stitching the output of an image encoder to the input of a language model in the form of soft prompts [84]. However, these approaches involve training of stitching layers for evaluating the representation similarity between two models.

In this work, we argue that using an explicit similarity measure as done in [110, 118] is sensitive to the selection of anchors and noise in the original embeddings. One design choice is an implicit measure that captures the similarity of similarities, hence, inducing more robustness to the alignment process. Furthermore, we explore how this similarity can be leveraged for downstream cross-modal tasks in a *training-free* manner with the aid of CKA and a set of parallel anchors in the image and text latent embedding spaces.

5.3 Preliminaries

Centered Kernel Alignment (CKA) has shown its relevance in understanding and comparing the information encoded by different layers of a neural network [75]. Formally, CKA relies on two sets of data $\mathbf{X} \in \mathbb{R}^{p \times N}$ and $\mathbf{Y} \in \mathbb{R}^{q \times N}$ through their corresponding kernels $\mathbf{K} = k(\mathbf{X}^\top, \mathbf{X}) \in \mathbb{R}^{N \times N}$ and $\mathbf{L} = \ell(\mathbf{Y}^\top, \mathbf{Y}) \in \mathbb{R}^{N \times N}$ where k, ℓ are some kernel functions applied on the columns of $\mathbf{X}$ and $\mathbf{Y}$ respectively (e.g., linear or RBF kernels). Therefore, the CKA is computed in terms of $\mathbf{K}$ and $\mathbf{L}$ as:

$$\mathrm{CKA}(\mathbf{K}, \mathbf{L}) = \frac{\mathrm{HSIC}(\mathbf{K}, \mathbf{L})}{\sqrt{\mathrm{HSIC}(\mathbf{K}, \mathbf{K})\,\mathrm{HSIC}(\mathbf{L}, \mathbf{L})}}, \tag{5.1}$$

where $\mathrm{HSIC}(\cdot, \cdot)$ is the Hilbert-Schmidt Independence Criterion [57, 97] defined as:

$$\mathrm{HSIC}(\mathbf{K}, \mathbf{L}) = \frac{1}{(N-1)^2}\,\mathrm{tr}\left(\mathbf{KCLC}\right), \tag{5.2}$$

with $\mathbf{C} = \mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^\top$ the centring matrix. We refer the reader to [75] for broader properties and studies of the CKA metric on neural network representations.

5.4 Proposed Method

Consider a set of N image-caption pairs, $\mathcal{S} = \{(\boldsymbol{x}_i, \mathbf{c}_i)\}_{i=1}^N$, where $\boldsymbol{x}_i \in \mathcal{X}$ and $\mathbf{c}_i \in \mathcal{C}$ represent the i-th image and its corresponding caption, respectively. In this particular example, we are performing caption-to-image retrieval, but it is applicable for the reverse as well. Let $\boldsymbol{f} : \mathcal{X} \mapsto \mathbb{R}^{d_1}$ and $\boldsymbol{g} : \mathcal{C} \mapsto \mathbb{R}^{d_2}$ denote some vision and language encoders respectively. The image-caption pairs are mapped into their corresponding sets of representations $\mathbf{Z} = [\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N] \in \mathbb{R}^{d_1 \times N}$ and $\mathbf{H} = [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_N] \in \mathbb{R}^{d_2 \times N}$, where $\boldsymbol{z}_i = \boldsymbol{f}(\boldsymbol{x}_i)$ and $\boldsymbol{h}_i = \boldsymbol{g}(\boldsymbol{c}_i)$.

As shown in Table 5.1, the maximum CKA score is obtained on the ground-truth ordering of the representations $\mathrm{CKA}_{\max} = \mathrm{CKA}(\mathbf{K_Z}, \mathbf{K_H})$, where $\mathbf{K_Z}$ and $\mathbf{K_H}$ are the kernels for the image and text representations, defined respectively as $\mathbf{K_Z} = k(\mathbf{Z}^\top, \mathbf{Z})$ and $\mathbf{K_H} = k(\mathbf{H}^\top, \mathbf{H})$. We find that the CKA is sensitive to the data ordering. Specifically, we shuffle x% of data to obtain wrong matches while keeping the remaining 100-x% aligned, measure the CKA on each new data set, and observe that it monotonically decreases with random shuffling. This motivates our methodology for finding an optimal permutation of the image data that maximizes the CKA.

Formally, let σ be some permutation of the set $\{1, \cdots, N\}$ and denote $\sigma(\mathbf{Z}) = [\boldsymbol{z}_{\sigma(1)}, \cdots, \boldsymbol{z}_{\sigma(N)}] \in \mathbb{R}^{d_1 \times N}$ the set of permuted image representations by σ. If σ is not identity, it disrupts the original ordering of the image representations leading to a

Table 5.1: **CKA reduces with shuffling.** We measure the CKA score between DINOv2 [121] and All-Roberta-large-v1 [92] on the 5k COCO [89] image-caption representations pairs of the valset. The exact ordering yields the best score, whereas randomly shuffling the representations reduces the CKA score.

Shuffling (%)	0	20	40	60	80	100
CKA Score	0.72	0.46	0.27	0.13	0.04	0.01

lower CKA score as shown in Table 5.1. Therefore, our goal is to find a permutation σ^* that maximizes the CKA. Formally:

$$\sigma^* = \arg\max_{\sigma} \text{CKA}(\mathbf{K}_{\sigma(\mathbf{Z})}, \mathbf{K}_{\mathbf{H}}). \tag{5.3}$$

The solution to this problem seeks to realign the permuted set of images in a way that maximizes the CKA, potentially recovering the ground-truth pairing between images and their corresponding captions.

To solve the aforementioned optimization problem, we explore two main approaches (visualized in Fig. 5.1): the Quadratic Assignment Problem (QAP) algorithm and Local CKA-based retrieval and matching. The QAP algorithm provides a global matching solution, seeking the optimal permutation across the query set considered. On the other hand, Local CKA-based retrieval and matching focuses on aligning images and captions using a localized metric, facilitating retrieval on a more granular level. This approach is more suitable where a single query image is given for a set of captions or *vice versa*.

5.4.1 QAP Matching

For some random permutation σ, the optimization problem in Equation 5.3 can be reformulated as a quadratic optimization problem [181] which reads as:

$$\max_{\mathbf{P}\in\mathcal{P}_N} \text{tr}\left(\mathbf{P}^\top \bar{\mathbf{K}}_{\sigma(\mathbf{Z})} \mathbf{P} \bar{\mathbf{K}}_{\mathbf{H}}\right), \tag{5.4}$$

where $\mathcal{P}_N$ is the set of all permutation matrices of size N and $\bar{\mathbf{K}} = \text{HSIC}(\mathbf{K}, \mathbf{K})^{-\frac{1}{2}}\mathbf{KC}$ stands for the centered and re-scaled kernel. In principle, maximizing the above objective is a relaxation of a graph-matching problem. Moreover, finding a global maximum of Equation 5.4 is NP-hard due to the combinatorial nature of the problem and therefore optimizing it can lead to sub-optimal or approximate solutions.

To overcome the NP-hardness of QAP, in practice, we suppose that we have access to a base set $\mathcal{B} = \{(\boldsymbol{z}_i^b, \boldsymbol{h}_i^b)\}_{i=1}^M$ of image-caption representations pairs and solve an equivalent objective to Equation 5.4 only partially on some unmatched

query set $\mathcal{Q} = \{\boldsymbol{z}_i^q\}_{i=1}^N \times \{\boldsymbol{h}_i^q\}_{i=1}^N$ using a seeded version of the fast QAP algorithm [49]. Formally, let $\mathbf{Z} = [\boldsymbol{z}_1^b, \cdots, \boldsymbol{z}_M^b, \boldsymbol{z}_1^q, \cdots, \boldsymbol{z}_N^q] \in \mathbb{R}^{d_1 \times (M+N)}$ and $\mathbf{H} = [\boldsymbol{h}_1^b, \cdots, \boldsymbol{h}_M^b, \boldsymbol{h}_1^q, \cdots, \boldsymbol{h}_N^q] \in \mathbb{R}^{d_2 \times (M+N)}$ be the matrix concatenating all base and query representations of images and captions respectively, and denote by $\bar{\mathbf{K}}_{\mathbf{Z}}, \bar{\mathbf{K}}_{\mathbf{H}} \in \mathbb{R}^{(M+N)\times(M+N)}$ the corresponding centered and re-scaled kernels. The partial matching for aligning the query samples is then performed by solving the following:

$$\max_{\mathbf{P} \in \mathcal{P}_N} \operatorname{tr}\left((\mathbf{I}_M \oplus \mathbf{P})^\top \bar{\mathbf{K}}_{\mathbf{Z}} (\mathbf{I}_M \oplus \mathbf{P}) \bar{\mathbf{K}}_{\mathbf{H}}\right), \tag{5.5}$$

where $\mathbf{I}_M \oplus \mathbf{P} \in \mathbb{R}^{(M+N)\times(M+N)}$ stands for the block-diagonal matrix having diagonal blocks $\mathbf{I}_M$ and $\mathbf{P}$.

5.4.2 Local CKA based Retrieval and Matching

The concept of a global CKA metric is extended to derive local similarity measures suitable for retrieval. This process begins with a base set $\mathcal{B} = \{(\boldsymbol{z}_i^b, \boldsymbol{h}_i^b)\}_{i=1}^M$ consisting of aligned pairs of images and captions representations. The objective is to facilitate caption-image retrieval/matching within an unaligned query set $\mathcal{Q} = \{\boldsymbol{z}_i^q\}_{i=1}^N \times \{\boldsymbol{h}_i^q\}_{i=1}^N$.

A local CKA score, denoted as localCKA$(\boldsymbol{z}^q, \boldsymbol{h}^q)$ for a couple $(\boldsymbol{z}^q, \boldsymbol{h}^q) \in \mathcal{Q}$ is calculated by computing a global CKA score for the image-caption pairs in $\mathcal{B}$, augmented with the query pair $(\boldsymbol{z}^q, \boldsymbol{h}^q)$. The local CKA is computed as follows:

$$\text{localCKA}(\boldsymbol{z}^q, \boldsymbol{h}^q) = \text{CKA}(\mathbf{K}_{[\mathbf{Z}, \boldsymbol{z}^q]}, \mathbf{K}_{[\mathbf{H}, \boldsymbol{h}^q]}), \tag{5.6}$$

where $[\mathbf{M}, \boldsymbol{v}]$ denotes the concatenation of the matrix $\mathbf{M}$ and the vector $\boldsymbol{v}$ column-wise and $\mathbf{Z} = [\boldsymbol{z}_1^b, \cdots, \boldsymbol{z}_M^b] \in \mathbb{R}^{d_1 \times M}$ and $\mathbf{H} = [\boldsymbol{h}_1^b, \cdots, \boldsymbol{h}_M^b] \in \mathbb{R}^{d_2 \times M}$. In essence, a correctly matched image-caption pair in $\mathcal{Q}$ would exhibit a higher degree of alignment with the base set $\mathcal{B}$ in terms of the CKA score, resulting in an elevated localCKA score. This metric can be used to calculate a score between one source query and N target queries enabling effective retrieval. Furthermore, this framework allows for the use of linear sum assignment [78] for matching tasks.

5.4.3 Stretching and Clustering

The choice of base samples and the spread of the representations in each embedding space affect the performance of the QAP and Local CKA algorithms. To spread the representations out in each domain for matching, we introduce a stretching matrix that normalizes the features of each dimension by the variance calculated from the query and base sets. Given $\mathbf{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_d]^\top \in \mathbb{R}^{d \times N}$, the stretched matrix $\mathbf{X}_s$ is

computed as $\mathbf{X}_s = \mathbf{S}\mathbf{X}$, where the stretching matrix $\mathbf{S} \in \mathbb{R}^{d \times d}$ is a diagonal matrix with inverse empirical standard deviation of the feature dimension as entries, *i.e.*,, $\mathbf{S} = \text{diag}\left(\frac{1}{\text{std}(\boldsymbol{x}_1)}, \cdots, \frac{1}{\text{std}(\boldsymbol{x}_d)}\right)$ and $\boldsymbol{x}_i \in \mathbb{R}^N$ is the i^{th} row of $\mathbf{X}$.

This stretching operation is performed for both the image and text before calculating the kernels for both QAP and local CKA matching algorithms. For picking the most effective base samples, we find that the simple k-means clustering on the image embeddings works best. An ablation on how these affect the QAP and local CKA matching and retrieval accuracies is provided in Sec 5.7.

5.5 Experiments

We assess the performance of the proposed method using various vision and language encoders on a set of downstream tasks. We first detail the encoders, datasets, downstream tasks, and the baselines used.

5.5.1 Vision and Language Encoders

The experimental setup covers vision encoders of different architectures, such as ViTs [39] and ConvNeXt [93], trained in various ways: supervised, language-supervised, and self-supervised, across different training data regimes. For the language encoder, an encoder capable of producing a global embedding for a caption is essential. This includes encoders of multiple architectures varying in size, languages, and training data sizes. The Huggingface's sentence-transformers [139] library is utilized, where each sentence transformer is first pre-trained on the masked language modeling task using a large text corpus, followed by a finetuning stage on a sentence pairs dataset with a contrastive loss. It's not straightforward to acquire a global sentence embedding from decoder-only models like GPT models [133, 15], hence we did not study the semantic alignment of these class of models to vision encoders.

The CKA and Matching Score (MS) of the various combinations of vision and language encoders are reported in supplementary. The findings indicate that the All-Roberta-large-v1 [92] demonstrates the best CKA/MS across all vision models, establishing it as the primary language encoder for subsequent tasks, unless specified otherwise.

5.5.2 Baselines

Here, we briefly describe three baselines that we compare our methods against for caption matching/retrieval, image classification, and cross-lingual tasks.

Linear Regression: We propose a baseline that learns a linear transformation from the image embedding space to the text using M aligned base examples and apply

the transformation to the query image embeddings. Concretely, given query image embeddings $\mathbf{Z}^q = [\boldsymbol{z}_1^q, \cdots, \boldsymbol{z}_N^q] \in \mathbb{R}^{d_1 \times N}$ and text embeddings $\mathbf{H}^q = [\boldsymbol{h}_1^q, \cdots, \boldsymbol{h}_N^q] \in \mathbb{R}^{d_2 \times N}$, and a set of aligned base samples $\mathbf{Z}^b = [\boldsymbol{z}_1^b, \cdots, \boldsymbol{z}_N^b] \in \mathbb{R}^{d_1 \times M}$ and $\mathbf{H}^b = [\boldsymbol{h}_1^b, \cdots, \boldsymbol{h}_N^b] \in \mathbb{R}^{d_2 \times M}$, we first construct a linear transformation between $\mathbf{Z}^b$ and $\mathbf{H}^b$ by minimizing the MSE loss as $\mathbf{W} = \arg\min_{\mathbf{W}} \|\mathbf{W}^\top \mathbf{Z}^b - \mathbf{H}^b\|_F^2$. Then we use $\mathbf{W}$ to transform the query image embeddings $\mathbf{Z}^q$ to the text domain as $\hat{\mathbf{H}}^q = \mathbf{W}^\top \mathbf{Z}^q$. Cosine similarity on $\hat{\mathbf{H}}^q$ and $\mathbf{H}^q$ can be used to perform caption retrieval.

Relative Representations [110]: enable latent space communication between unaligned encoders by representing each query point relative to an aligned base set. Concretely, let ℓ_2-normalized embeddings for image and text queries be $\mathbf{Z}^q = [\boldsymbol{z}_1^q, \cdots, \boldsymbol{z}_N^q] \in \mathbb{R}^{d_1 \times N}$ and $\mathbf{H} = [\boldsymbol{h}_1^q, \cdots, \boldsymbol{h}_N^q] \in \mathbb{R}^{d_2 \times N}$, respectively. Utilizing a set of aligned base sample ℓ_2-normalized embeddings $\mathbf{Z}^b = [\boldsymbol{z}_1^b, \cdots, \boldsymbol{z}_M^b] \in \mathbb{R}^{d_1 \times M}$ and $\mathbf{H}^b = [\boldsymbol{h}_1^b, \cdots, \boldsymbol{h}_M^b] \in \mathbb{R}^{d_2 \times M}$, we can construct relative image and text query representations as $\mathbf{Z}_{rel}^q = (\mathbf{Z}^b)^\top \mathbf{Z}^q$ and $\mathbf{H}_{rel}^q = (\mathbf{H}^b)^\top \mathbf{H}^q$. Relative representations are a single vector of dimension M for each query specifying the cosine similarity of a query sample with all the base samples. Now we can use the cosine similarity on the relative representations to perform retrieval. Sec D in appendix provides a further comparison with our method.

CLIP [132]: We also compare against CLIP which has been contrastively trained to obtain a joint embedding space- as an upper limit on performance for both retrieval and matching tasks. We perform retrieval using cosine similarity

For all 3 methods, caption matching can be achieved by constructing a cost matrix using cosine similarities and using linear sum assignment to find the permutation matrix.

5.5.3 Downstream Tasks

Caption Matching: Given N query images and their corresponding captions, a query set is constructed by shuffling the captions. The task involves finding the correct permutation over captions for perfect matching. In *Retrieval*, the objective is, given one caption, to retrieve the correct image from the overall set of N images. The alignment between unaligned vision and text encoders is investigated using our methods on the COCO and NoCaps validation sets.

The COCO dataset [89] comprises over 120,000 images with multiple captions per image. It is used for testing unimodal representation quality via a caption-matching task, utilizing a validation set of 5,000 image-caption pairs. The NoCaps dataset [5] is designed for testing image captioning models on unseen objects, with 166,100 captions for 15,100 images from OpenImages. Its validation set includes novel concepts absent from COCO.

Table 5.2: **Caption matching and retrieval task performance comparison in cross-domain and in-domain settings.** Base samples from COCO are utilized for matching/retrieval tasks on queries from NoCaps (cross-domain) and COCO (in-domain). CLIP-V denotes the vision encoder of CLIP [132]. We use the Large version of all vision encoders.

Method	**Vision Model**	**NoCaps** [5]		**COCO** [89]	
		Matching accuracy	Top-5 retrieval	Matching accuracy	Top-5 retrieval
Cosine Similarity*	CLIP [132]	99.5	99.6	97.1	96.1
Linear regression	CLIP-V [132]	29.3	44.7	42.7	59.1
	ConvNeXt [188]	19.0	28.5	31.3	46.1
	DINOv2 [121]	38.1	50.3	45.1	65.4
Relative representations [110]	CLIP-V [132]	61.3	37.6	61.6	41.3
	ConvNeXt [188]	25.5	17.8	38.6	34.1
	DINOv2 [121]	46.0	46.4	47.7	52.3
Ours: QAP	CLIP-V [132]	**67.3**	-	**72.3**	-
	ConvNeXt [188]	46.7	-	66.1	-
	DINOv2 [121]	57.7	-	66.0	-
Ours: Local CKA	CLIP-V [132]	65.1	60.5	71.9	69.9
	ConvNeXt [188]	43.7	44.4	64.8	65.5
	DINOv2 [121]	58.7	**61.8**	64.3	**70.5**

Cross-lingual Caption Matching/Retrieval: The task mirrors prior matching and retrieval but uses multilingual captions, say German. Given N images and shuffled German captions, the objective is to match each image with the correct caption. In retrieval, the goal is to select the most fitting German caption for a given query image from the set.

The XTD-10 dataset [4] enhances COCO2014 with 1,000 human-annotated multilingual captions in ten languages for cross-lingual image retrieval and tagging, serving as a zero-shot model benchmark.

ImageNet-100 Classification. The task setup is similar to the conventional classification task with small differences to account for the methods used. Given N query images and their corresponding classes, image representations are obtained by processing them through a vision encoder. In parallel, textual representations are generated in a multi-step process. Initially, several text captions are derived from the class-associated Wordnet synsets' lemmas, definitions, and hypernyms. These captions are then passed through the language encoder and averaged to get the text representations. The classification task is performed by retrieving the closest text representations to each image representation using our local CKA metric. We employ the ImageNet-100 dataset. This dataset is a subset of the larger ImageNet dataset, featuring only 100 classes. It includes 130,000 training images, 50,000 validation images, and 100 classes.

Table 5.3: **Reverse Caption Retrieval Results for COCO and NoCaps**. In this setting, the retrieval objective is, given one image, to retrieve the correct caption from the overall set of N captions. The matching objective remains quite similar but instead of shuffling the captions, this time, the images are shuffled.

Method	Vision Model	NoCaps [5]		COCO [89]	
		Matching accuracy	Top-5 retrieval	Matching accuracy	Top-5 retrieval
Cosine Similarity*	CLIP [132]	99.5	99.6	97.1	98.5
Linear regression	CLIP-V [132]	63.6	70.1	72.6	83.9
	ConvNeXt [188]	22.8	38.9	43.8	65.7
	DINOv2 [121]	46.8	59.9	56.2	75.9
Relative representations [110]	CLIP-V [132]	61.3	3.0	61.6	2.9
	ConvNeXt [188]	25.5	2.7	38.6	12.9
	DINOv2 [121]	45.9	38.1	47.7	43.7
Ours: QAP	CLIP-V [132]	67.3	-	72.8	-
	ConvNeXt [188]	45.9	-	65.1	-
	DINOv2 [121]	58.5	-	65.9	-
Ours: Local CKA	CLIP-V [132]	65.1	65.9	71.9	80.5
	ConvNeXt [188]	44.8	33.0	63.8	74.3
	DINOv2 [121]	55.7	64.2	64.3	76.0

5.5.4 Results

Importance of Good Initialization:

For all tasks, we make use of a set of base samples of size S that is kept fixed at 320 samples. The size of the query set is analogously fixed at 500 samples (see Sec 5.7.2 for more details). These base samples are selected after clustering the image embeddings and choosing one closest sample to each of the S cluster centers. By aligning the initial samples with the diverse cluster centers, we ensure sufficient coverage of the sample space. This enhances the accuracy of the matching process, as the initial alignment closely mirrors the inherent structure and variability within the data. In the case of linear regression, uniform sampling is employed to select the base samples. For relative representations [110], the same clustering methodology is applied to select base samples, ensuring a fair and consistent comparison between all methods.

COCO and NoCaps Caption Matching:

We present the results of cross-domain and in-domain caption matching/retrieval, as detailed in Table 5.2. We tested each baseline against three different vision models, while employing a consistent language model—specifically, the all-roberta-large-v1. The vision models utilized are OpenAI's CLIP ViT-L/14, the ConvNeXT-Base model (trained on the ImageNet-22k dataset at a resolution of 224x224), and the ViT-L/14 model trained using the DINOv2 method. It is important to note that the first row of the results table features vision and language models both being OpenAI's CLIP ViT-L/14. To effectively analyze cross-domain capabilities, our experiment design involved the use of the COCO validation set as the source of

the base set and the NoCaps validation set for querying. Additionally, in-domain results are shown, when using COCO validation for both base and queries. We uniformly sample the query set and average the results over three different seeds. Although CLIP's cosine similarity metric emerges as the most robust due to the training paradigm inherent in CLIP models, our methods demonstrate commendable performance without necessitating any training. The DINOv2 model, trained solely through self-supervision, demonstrates the formation of semantic concepts independently of language supervision. This is evident in its remarkable top-5 retrieval scores of 70.5% and 61.8% on COCO and NoCaps datasets when coupled with an unaligned language encoder through our Local Kernel CKA method. However, the best-performing vision encoder is CLIP's vision encoder which has been trained using language supervision.

While the performance on the image retrieval task was reported above, here in Table 5.3, we show the NoCaps and Coco caption retrieval results in the reverse setting. In this configuration, the retrieving objective shifts to finding the correct caption from a pool of N captions when given a single image. The matching objective remains consistent, but, instead of shuffling the captions, the images themselves are shuffled. While the matching accuracies express minimal changes in this setting, the retrieval accuracies display notable discrepancies.

A plausible explanation for the reduced retrieval scores associated with the relative representation method is the heightened semantic variability inherent in the image domain compared to the caption domain. A considerable number of images share very similar captions, leading to a compressed semantic space for the captions. Consequently, caption embeddings become more closer to one another, making the retrieval a lot harder.

ImageNet-100 Classification:

In Table 5.7, we detail the performance of our methods on the ImageNet-100 classification task. Mirroring our approach in cross-domain matching and retrieval, we evaluated three different vision models for each method. Notably, the first row of the table highlights the performance using CLIP's embedding cosine similarity. The results are averaged over three different seeds for sampling the query set. A significant observation from this table is the comparatively narrower performance gap between the CLIP's cosine similarity and our methods, as well as the baseline linear regression method, in contrast to the results observed in cross-domain caption matching/retrieval tasks.

It is interesting that ConvNeXt encoder trained on ImageNet has a classification top1 accuracy improvement of over 14% compared to CLIP and Dinov2 while on the caption matching task DinoV2 and CLIP perform much better.

Table 5.4: **Cross-Lingual caption matching and retrieval performance comparison.** Using QAP and local CKA-based methods we are able to do cross-lingual caption matching/retrieval using CLIP's ViT-L vision encoder and a multi-lingual sentence transformer paraphrase-multilingual-mpnet-base-v2. While CLIP performs well on the Latin languages, it degrades on non-Latin languages. In comparison, our QAP and Local-CKA-based methods perform comparably in Latin languages while outperforming non-Latin languages, highlighting the efficacy of our training-free transfer approach.

Language		**Kernel CKA**		**Matching Accuracy**				**Retrieval @ 5**	
		CLIP	Ours	CLIP	Relative[110]	Linear	Ours (QAP)	CLIP	Ours (Local)
Latin	de	0.472	0.627	41.8	35.0	34.0	39.6	65.1	56.7
	en	0.567	0.646	81.5	52.5	40.9	51.6	92.5	69.0
	es	0.471	0.634	50.2	37.8	31.7	41.4	68.5	61.6
	fr	0.477	0.624	49.4	37.5	30.7	40.2	68.7	57.6
	it	0.472	0.638	41.0	37.2	34.9	38.5	61.3	59.7
Non-Latin	jp	0.337	0.598	13.2	28.3	23.5	30.5	30.0	49.4
	ko	0.154	0.620	0.50	30.4	23.5	30.9	3.30	53.4
	pl	0.261	0.642	5.40	36.6	30.2	40.2	18.8	59.5
	ru	0.077	0.632	0.80	31.9	30.7	35.1	4.10	53.2
	tr	0.301	0.624	4.30	35.8	29.6	38.9	15.2	59.3
	zh	0.133	0.641	2.70	36.5	31.1	40.3	8.90	57.8
	Avg.	–	–	26.4	36.3	30.9	**38.8**	39.6	**57.9**

Cross-lingual Caption Retrieval:

The results of cross-lingual caption matching/retrieval are presented in Table 5.4 for the 10 languages in the XTD-dataset. OpenAI CLIP's ViT-L vision encoder, trained on English image-caption pairs, and a multilingual sentence transformer paraphrase-multilingual-mpnet-base-v2 were utilized for this task. The accuracy of CLIP's cosine retrieval method exhibits a significant drop when applied to languages other than English. *E.g.,* CLIP's retrieval at 5 experiences a drop of 30 points when switching from English to other Latin-alphabet languages (Spanish, French, German, and Italian). For non-Latin alphabet languages such as Korean, Chinese, Turkish, *etc.*, CLIP's performance decreases substantially, collapsing to zero, primarily due to most words resulting in unknown tokens. In contrast, the QAP and local CKA matching methods demonstrate consistent performance across all languages, including non-Latin languages, attributing to the robustness of a multilingual sentence transformer trained solely on text. On average, QAP surpasses CLIP by 12% in the caption matching task and also outperforms other baselines like relative representations and linear regression methods. For retrieval at 5, the local CKA-based method exceeds CLIP's performance by over 17%.

It is possible to use language-specific sentence encoders and we report these results for a few languages below. This is a practical application of our method as we can now turn a well-trained English CLIP model's vision encoder into a CLIP model for any low-resource language if a text-only Sentence Transformer trained on

Table 5.5: **Cross-Lingual image matching and retrieval performance comparison. Here we use multilingual captions to retrieve images from the COCO validation set.** Using QAP and local CKA-based methods we are able to do cross-lingual image matching/retrieval using CLIP's ViT-L vision encoder and a multi-lingual sentence transformer paraphrase-multilingual-mpnet-base-v2. While CLIP performs well on the Latin languages, it degrades on non-Latin languages. In comparison, our QAP and Local-CKA-based methods perform comparably in Latin languages while outperforming non-Latin languages, highlighting the efficacy of our training-free transfer approach.

Language		**Kernel CKA**		**Matching Accuracy**				**Retrieval @ 5**	
		CLIP	Ours	CLIP	Relative[110]	Linear	Ours (QAP)	CLIP	Ours (Local)
Latin	de	0.472	0.627	43.5	35.0	19.3	39.7	54.9	57.2
	en	0.567	0.646	80.9	52.5	25.6	51.3	90.4	66.7
	es	0.471	0.634	50.4	37.8	19.7	40.9	63.9	57.9
	fr	0.477	0.624	50.8	37.5	18.8	40.3	65.9	56.9
	it	0.472	0.638	41.9	37.2	19.7	38.7	52.9	57.0
Non-Latin	jp	0.337	0.598	12.9	28.3	15.2	30.2	17.8	48.6
	ko	0.154	0.620	0.9	30.4	15.3	31.3	2.2	48.4
	pl	0.261	0.642	8.1	36.6	21.0	40.0	15.7	55.9
	ru	0.077	0.632	1.7	31.8	16.3	34.8	3.5	53.9
	tr	0.301	0.624	7.8	35.8	18.7	38.9	14.6	53.1
	zh	0.133	0.641	2.4	36.5	19.2	39.9	4.8	53.7
	Avg.	–	–	27.4	36.3	18.9	**38.7**	35.1	**55.4**

that language is available.

For completeness, we report the results in Table 5.5 for the reverse setting of the cross-lingual image caption matching/retrieval task mentioned in the main paper. Given N captions in say, German, and N shuffled images the objective is to match each German caption with the correct image. In retrieval, the goal is to select the most fitting image from the retrieval set given a German caption. We notice that the matching accuracies remain the same as the direction doesn't affect the matching. However, in the case of reverse retrieval, we see that CLIP's retrieval@5 drops by over 4.5% on average when compared to our local CKA based retrieval of 2.1%.

In Table 5.6 we report the results for when we use language-specific BERT Sentence encoders for the cross-lingual caption matching/ retrieval task for 5 languages. For all these cases, the vision encoder is kept fixed as OpenAI's CLIP-VIT-L-14 trained on English image, caption pairs. We notice that the semantic alignment with the vision encoder in terms of CKA as well as matching/retrieval performance drops with language-specific encoders when compared to using a multi-lingual model like multilingual-mpnet-base-v2. We believe this could be due to the multi-lingual model being trained on a lot more data in comparison to the language-specific ones thus resulting in more meaningful embedding spaces.

Table 5.6: **Language-specific encoders for cross-lingual caption matching/retrieval for 5 languages**. Language-specific encoders have less semantic similarity with the vision encoder in terms of CKA as well as poorer matching/accuracy performances when compared to multi-lingual models like multilingual-mpnet-base-v2 which is reported in Table 4.

Language	Language model	CKA	Linear	Relative	QAP	Retrieval@5
es	hiiamsid\sentence_similarity_spanish_es	0.568	15.9	25.1	28.6	50.0
fr	dangvantuan\sentence-camembert-large	0.569	22.5	31.5	35.0	53.1
it	nickprock\sentence-bert-base-italian-uncased	0.543	16.0	22.0	26.4	47.8
jp	colorfulscoop\sbert-base-ja	0.457	9.2	12.1	14.5	33.7
tr	emrecan\bert-base-turkish-cased-mean-nli-stsb-tr	0.564	23.1	34.7	38.3	54.3

Table 5.7: **ImageNet-100 classification performance comparison.** We observe a narrow performance gap between the CLIP model and our methods. CLIP-V denotes the vision encoder of CLIP.

Method	Vision Model	Top 1	Top 5
Cosine Similarity*	CLIP	86.1	99.2
Linear Regression	CLIP-V	76.1	93.0
	ConvNeXt	84.5	95.4
	DINOv2	73.5	92.1
Relative representations [110]	CLIP-V	8.90	30.3
	ConvNeXt	7.20	15.7
	DINOv2	49.7	75.5
Local CKA	CLIP-V	68.7	91.2
	ConvNeXt	83.3	95.8
	DINOv2	67.7	88.3

5.5.5 Matching complexity

In Table 5.8, we go over the time complexity and runtimes of QAP matching and local CKA based retrieval in comparison to the other baselines for matching when number of base samples and query samples are 320, 500 respectively. For all time complexities, we assume number of base samples m to be of the order of the number of query samples n. QAP uses the seeded version of the fast QAP algorithm from the SciPy library, which has a worst time complexity of $\mathcal{O}(n^3)$ [49], while local CKA retrieval requires constructing a graph over all the query image and text pairs, $\mathcal{O}(n^2)$, using local CKA, which is also $\mathcal{O}(n^2)$ resulting in $\mathcal{O}(n^4)$. Relative involves the calculation of the relative representations for every query image and text pair, resulting in a time complexity of $\mathcal{O}(n^2)$, but it's fast due to highly optimized algorithms for matrix multiplications in PyTorch [127]. Linear has a time complexity of

Table 5.8: Run times for different methods

Method	QAP	Local CKA	Relative	Linear
Run times	40 seconds	5 mins	1 second	1 second
Complexity	$\mathcal{O}(n^3)$	$\mathcal{O}(n^4)$	$\mathcal{O}(n^2)$	$\mathcal{O}(n \times d)$

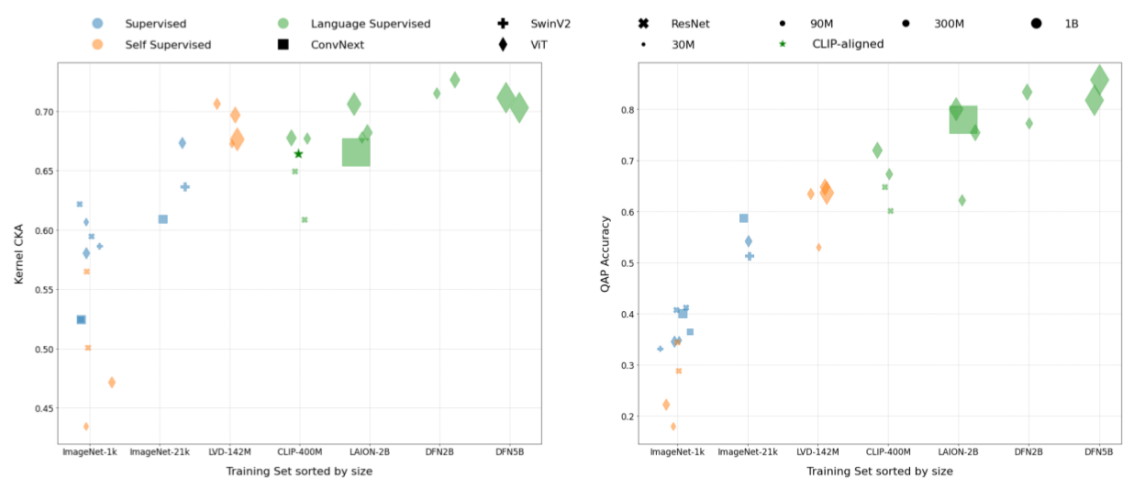


Figure 5.2: **Kernel CKA and QAP Matching accuracy are correlated with the training set size and quality of the training set.** Here the language encoder is kept constant to the best BERT-sentence encoder (i.e.All-Roberta-large-v1). There is a clear correlation between CKA and QAP Matching accuracy across all architectures, training paradigm and data regimes.

$\mathcal{O}(nd)$, where n is the number of samples and d is the number of dimensions. It is to be noted that QAP runs on the CPU, and a CUDA-optimized version could bring the runtimes further down from 40 seconds. An efficient implementation of Local Kernel CKA is also possible, where the CKA of base samples is precalculated, and the graph is constructed in an additive manner, which would bring down the time complexity to $\mathcal{O}(n^3)$. For both relative and linear matching, we make use of SciPy's modified Jonker-Volgenant algorithm [30] for linear sum assignment, which has the worst time complexity of $\mathcal{O}(n^3)$.

5.6 Analysis

5.6.1 Representational Similarity between Vision and Language Encoders

This section focuses on how training paradigms, data regimes, and encoder size/architecture influence a vision encoder's ability to represent the world similarly to a language encoder. This is assessed by comparing the semantic alignment of their representation spaces using CKA as well as QAP matching accuracy. Figure 5.2 compares the kernel CKA and caption matching accuracy of different vision encoders with a fixed text-encoder (*i.e.*,, All-Roberta-large-v1), against the training datasets on which the vision encoder was trained for all pairs in the COCO captions validation set. The findings are summarized below:

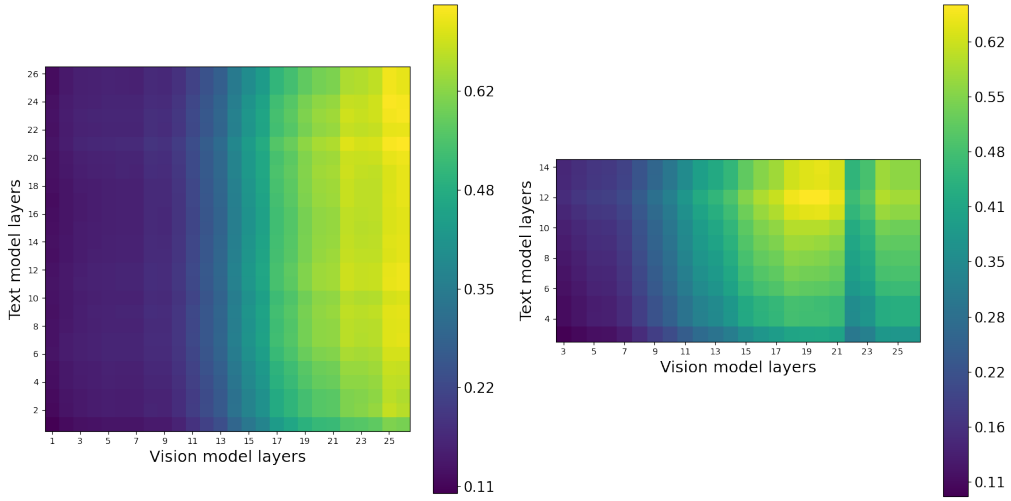
Scale and quality of dataset results in encoders with high semantic align-

ment with the language space: It is observed that SSL methods like DINOv2 can learn semantic concepts in a relative manner even without language supervision during training. The CKA and QAP matching accuracy for DINOv2 embeddings are comparable to CLIP models, despite lacking language supervision and having significantly less data (LVD-142's 142M vs Open-AI-CLIP's 400M). A general trend emerges where more training data leads to semantically richer visual embeddings, evident when comparing CKA and QAP Accuracies from ImageNet1K to DFN-5B datasets. Notably, training on a curated dataset proves more effective than on an uncurated dataset of the same size, especially for smaller models. This is illustrated by the higher CKA and QAP accuracy of ViT-Large trained on the curated DFN-2B dataset compared to ViT-Large/Giant, and ConvNext-xxLarge trained on Laion 2B. Additionally, SSL methods show less semantic consistency when trained on ImageNet1K, as indicated by the clear difference in QAP accuracies between DINO trained on ImageNet1K and DINOv2 trained on LVD-142M.

Vision Encoders Trained with Language Supervision Exhibit Greater Semantic Alignment with Language Encoders: In line with the findings of Merullo et al.[106], it is observed in our experiments that vision encoders trained with more language supervision on datasets of comparable size exhibit a higher degree of semantic alignment with language encoders compared to self-supervised methods. For example, ViT-Large trained on CLIP-400M with language supervision demonstrates superior caption-matching capabilities compared to DINOv2's ViT-Large trained on LVD-142M. Similarly, we verify that class label supervision, like that from ImageNet, leads to more semantically aligned image encoders when compared to self-supervision when similarly sized models are compared on ImageNet-1k. For example, all supervised encoders trained on ImageNet-1k have higher CKA as well as QAP matching accuracy than all the self-supervised models.

5.6.2 Layerwise CKA Analysis

Figure 5.3, Table 5.9, and Table 5.10 show the progression of CKA and QAP matching scores across layers for both text and vision models. We explore two configurations: one involves comparing layers of All-Roberta-large-V1 and DINOv2 VIT-L/14, while the other examines layers of CLIP's vision and text hidden states. For CLIP, the layer *proj* points to the final image and text embeddings that were passed through the final projection layers. In the first configuration, CKA and QAP scores gradually improve where the image model layer has a far greater effect on the similarity than the text model layer. On the other hand, the second configuration reveals that the QAP matching score in CLIP manifests prominently in the absolute last layers of both the vision/text encoders.



(a) (b)

Figure 5.3: **Layer-wise CKA heatmap illustration.** The heatmaps depict the CKA scores obtained by varying the layers from which the text and visual embeddings are taken. **On the left:** CKA scores for All-Roberta-large-v1 and DINOv2 unaligned combination. **On the right:** CKA scores for CLIP text and vision encoders. In both cases, we observe that the CKA scores are low for earlier layer embeddings of the vision model and they improve when the embeddings later layers are considered. This illustrates that both aligned and unaligned text-vision encoders behave similarly in terms of the cross-modal similarity with respect to CKA.

As shown in Table 5.9, the CLIP model obtains a significant jump in matching score after the projection head, highlighting the central role of this layer in aligning text and image modalities within a unified representation space. Here, the QAP matching accuracy does not follow a linear increase over the layers for CLIP, but rather suddenly jumps from 0.29 to 0.79 from the last layer to the projection head. This likely suggests that most of the CLIP performance comes from the projection heads ensuring a high statistical similarity. In contrast, Table 5.10 shows that DINOv2 and All-Roberta-large-v1 demonstrate a consistent improvement in the matching accuracy across successive layers, suggesting an inherent alignment process within their architectures in a hierarchical way. Here, the QAP matching accuracy linearly increases for the DINOv2 and All-Roberta-large-v1 combination when we fix the last layer of All-Roberta-large-v1 and vary the layers of DINOv2. Inversely, when we fix the last layer of DINOv2 and vary the layers of the text encoder, the QAP starts high at 0.44 and reaches 0.68 at the top layer, thus, we hypothesize that the text encoder representations do not change as much as the image representations.

Table 5.9: **QAP accuracy for different layers** of vision and text encoder of CLIP model.

		Vision					
		6th	11th	16th	21st	26th	proj
Text	6th	0.02	0.022	0.022	0.098	0.126	0.118
	11th	0.028	0.038	0.016	0.248	0.278	0.278
	14th	0.026	0.03	0.036	0.238	0.282	0.296
	proj	0.038	0.026	0.034	0.622	0.716	0.792

Table 5.10: **QAP accuracy for different layers** of DINOv2 and All-Roberta-large-v1 models.

		Vision				
		6th	11th	16th	21st	26th
Text	6th	0.008	0.020	0.150	0.314	0.448
	11th	0.010	0.022	0.146	0.360	0.498
	16th	0.008	0.016	0.194	0.334	0.500
	21st	0.002	0.004	0.148	0.420	0.538
	26th	0.008	0.016	0.198	0.450	0.672

5.6.3 Mathematical Relationship between Local CKA-based Retrieval and Relative Representations

In this section, we provide derivations that show that the relative representations method [110] can be seen as a particular case of our proposed localCKA method. Denote the set of query and base representations samples respectively as $\mathbf{Q}_A = \left[\boldsymbol{q}_1^A, \ldots, \boldsymbol{q}_N^A\right] \in \mathbb{R}^{d_A \times N}$ and $\mathbf{B}_A = \left[\boldsymbol{b}_1^A, \ldots, \boldsymbol{b}_M^A\right] \in \mathbb{R}^{d_A \times M}$, where $A \in \{I, C\}$ for images and captions, the retrieval matrix for the relative representations (RR) method is therefore given by:

$$\mathbf{R}^{\mathrm{RR}} = \mathbf{Q}_I^\top \mathbf{B}_I \mathbf{B}_C^\top \mathbf{Q}_C \in \mathbb{R}^{N \times N}.$$

From which, for instance, the i-th image query is mapped to its corresponding caption via:

$$\arg\max_j R_{ij}^{\mathrm{RR}} = \arg\max_j \left(\boldsymbol{q}_i^I\right)^\top \mathbf{B}_I \mathbf{B}_C^\top \boldsymbol{q}_j^C. \tag{5.7}$$

Whereas, our proposed localCKA method constructs the retrieval matrix $\mathbf{R}^{\mathrm{Ours}}$ having entries $R_{ij}^{\mathrm{Ours}} = \mathrm{localCKA}\left(\boldsymbol{q}_i^I, \boldsymbol{q}_j^C\right)$ with:

$$\mathrm{localCKA}\left(\boldsymbol{q}_i^I, \boldsymbol{q}_j^C\right) = \mathrm{CKA}\left(\mathbf{K}_{[\mathbf{B}_I, \boldsymbol{q}_i^I]}, \mathbf{K}_{[\mathbf{B}_C, \boldsymbol{q}_j^C]}\right). \tag{5.8}$$

Table 5.11: **Impact of adding noise to the embeddings**. Performance comparison, in terms of matching accuracy, between relative representations [110] and our global CKA-based QAP approach is shown for the image-caption matching task with 320 base samples and 500 query samples on COCO validation set. Gaussian noise with std-dev (σ) being a multiple of the embeddings std-dev is added to both image and textual embeddings. Noise level of 0 ($\sigma = 0$) denotes the performance for the original embeddings. The relative performance drop for a noise level from its reference ($\sigma = 0$) is shown in parenthesis. In comparison to relative representations, our QAP approach performance drops at a slower rate as σ increases, illustrating better noise robustness for our approach.

Method	**Noise Level (σ)**					
	0.0	0.1	0.2	0.3	0.4	0.5
Relative representations [110]	47.3	45.3 (↓4.4)	44.2 (↓6.5)	41.3 (↓12.7)	39.0 (↓17.6)	35.6 (↓24.8)
Ours (QAP)	53.9	53.7 (↓0.3)	51.8 (↓3.9)	48.7 (↓9.5)	46.9 (↓13.0)	43.3 (↓19.6)

In particular, taking the particular case of the linear kernel and defining the CKA score as the trace of the product of two kernels, i.e., $\text{CKA}(\mathbf{K}, \mathbf{L}) = \text{tr}\,(\mathbf{KL})$. We first have, for $A \in \{I, C\}$:

$$\mathbf{K}_{[\mathbf{B}_A, \boldsymbol{q}_i^A]} = [\mathbf{B}_A, \boldsymbol{q}_i^A]^\top [\mathbf{B}_A, \boldsymbol{q}_i^A] = \begin{bmatrix} \mathbf{B}_A^\top \mathbf{B}_A & \mathbf{B}_A^\top \boldsymbol{q}_i^A \\ \left(\mathbf{B}_A^\top \boldsymbol{q}_i^A\right)^\top & \|\boldsymbol{q}_i^A\|^2 \end{bmatrix}.$$

Hence, we have:

$$\begin{aligned} \text{tr}\left(\mathbf{K}_{[\mathbf{B}_I, \boldsymbol{q}_i^I]} \mathbf{K}_{[\mathbf{B}_C, \boldsymbol{q}_j^C]}\right) &= \text{tr}\left(\mathbf{B}_I^\top \mathbf{B}_I \mathbf{B}_C^\top \mathbf{B}_C\right) \\ &+ 2 \underbrace{\left(\boldsymbol{q}_i^I\right)^\top \mathbf{B}_I \mathbf{B}_C^\top \boldsymbol{q}_j^C}_{\text{relative representations term}} + \|\boldsymbol{q}_i^I\|^2 \|\boldsymbol{q}_j^C\|^2. \end{aligned}$$

Therefore, in this particular case, there is equivalence between our method and the relative representations method, since $R_{ij}^{\text{Ours}} = R_{ij}^{\text{RR}} + c$ where c is a constant scalar if the representations are normalized. As such, the relative representations method falls within our proposed localCKA method if one considers the linear kernel and takes the trace instead of the HSIC metric. Therefore, our proposed method is more general since it relies on general kernel functions and the HSIC metric, which might explain its performance.

Impact of noise addition: Table 5.11 shows the performance comparison between relative representations [110] and our global CKA-based QAP approach for the image-caption matching task with 320 base samples and 500 query samples on COCO validation set. For this experiment, 10 trials were conducted with different seeds and clustering of base samples was employed. Gaussian noise with std-dev (σ) being a multiple of the embeddings std-dev is added to both image and textual embeddings. The performance of original embeddings is also shown for reference

(noise level of 0, *i.e.*,, $\sigma = 0$). The relative performance drop for a noise level from its reference ($\sigma = 0$) is shown in parenthesis. Compared to relative representations, our QAP approach performance drops at a slower rate as σ increases. *E.g.*,, for $\sigma = 0.2$, relative representations matching accuracy drops 6.5% from its maximum of 47.3, while ours is more robust and drops only 3.9% from its maximum of 53.9 when $\sigma = 0$. These results show that our QAP approach is more robust to noise addition, in comparison to relative representations.

5.6.4 Mispredictions Visualization

In Table 5.12, we present instances of retrieval mispredictions where the original image fails to rank within the top five closest images to the given caption, as determined by local Kernel CKA method. Building upon the experimental methodology outlined in the main paper, we selected 320 base samples and conducted local Kernel CKA retrieval using an additional 500 query samples. We used All-Roberta-large-v1 for text embeddings and DINOv2 ViT-L/14 for image embeddings. The results distinctly illustrate that despite the failure to retrieve the exact original image, the alternative images identified in the top five still exhibit a considerable degree of semantic similarity to the provided caption. This underscores the robustness of the local Kernel CKA retrieval approach, revealing its capability to identify images that, while not the precise match, maintain semantic coherence with the specified caption.

5.7 Ablations

5.7.1 Ablation study on CKA, Stretching and Clustering

This section rationalizes our method choices through ablation studies on clustering, stretching, and the global CKA metric. We demonstrate the impact of these components on the performance of our methods, primarily through Table 5.13, which delineates the effectiveness of the QAP and the local CKA metric under various configurations. It shows the performance metrics in scenarios where each main component is either integrated or omitted. Notably, in instances where the CKA metric is not used, we opt for normalized correlation matrices for each graph. The empirical results presented are derived from the caption matching/retrieval task, utilizing both base and query sets extracted from the COCO validation set of size 320 and 500 respectively.

Choice of the metric: CKA is more beneficial than using just the scaled correlation matrix to represent the semantic relationships in an embedding space as matching accuracy increases from 10.1% to 48.8%. The choice of a robust metric is core to aligning vision and language latent spaces.

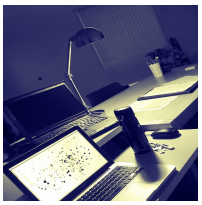
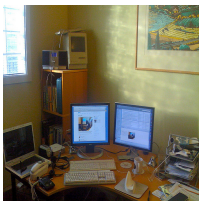
Original Image	Caption	Top-3 Retrieved Images		
	Two desktop computers sitting on top of a desk.			
	A mother and baby elephant walking in green grass in front of a bond.			
	a man is riding a surfboard at the beach			
	The Big Ben clock tower towering over the city of London.			
	A computer mouse is beside a notebook computer.			

Table 5.12: **Local Kernel CKA Retrieval Mispredictions.** In accordance with the experimental protocol detailed in the main paper, we selected 320 base samples and conducted local Kernel CKA retrieval using an additional 500 query samples. Presented above are five example prediction retrievals for instances where the original image failed to secure a position within the top-5 retrievals. We observe that although the original image was not in the retrieved top-5, the retrieved images (top-3 shown here) closely resemble the corresponding caption, thereby highlighting the efficacy of our approach.

Table 5.13: **Impact of clustering and stretching.** The matching and retrieval performance is the best when both clustering and stretching are employed. Hence, justifying this choice.

Clustering	Stretching	CKA	QAP Matching	Local CKA Matching	Local CKA Retrieval @ 5
✗	✗	✗	10.1	16.2	1.0
✗	✗	✓	48.8	48.5	60.2
✗	✓	✓	57.3	56.7	73.0
✓	✗	✓	56.2	55.1	66.4
✓	✓	✓	**65.5**	**63.3**	**77.2**

Impact of Stretching: It is clear that stretching facilitates better alignment of embeddings in our methods as stretching spreads the representations out in each modality without sacrificing the relative positions of the different embeddings within each embedding space. This is reflected in the increase of QAP accuracy from 48.8% to 57.3%.

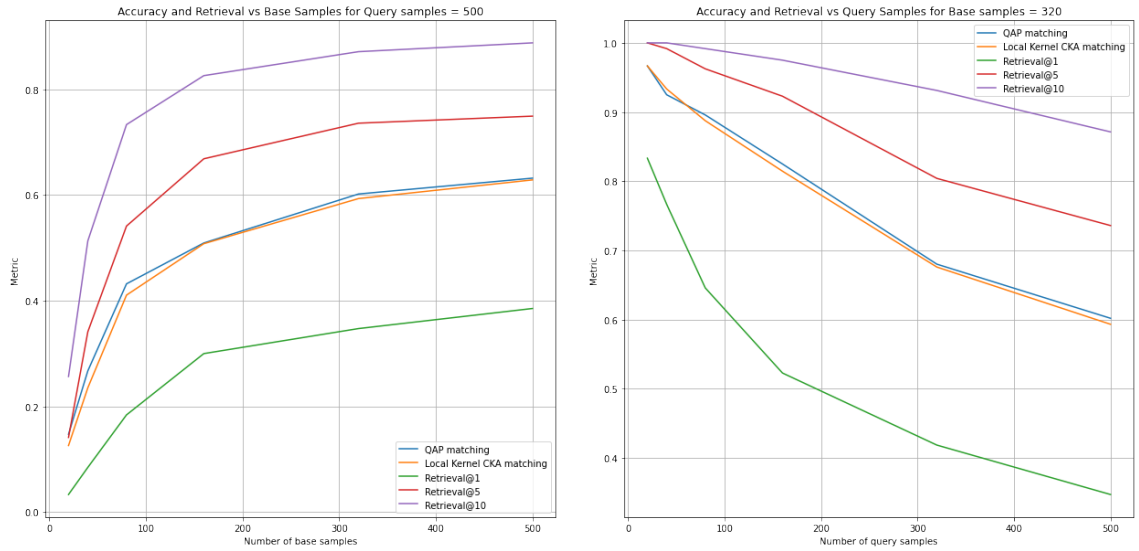
Clustering *vs.* Uniform Sampling: The choice of the base set is important in QAP matching and local CKA retrieval, as it measures any query pair alignment with the base set. A diverse base set is essential to capture a broad semantic range, and clustering within one of the embedding spaces aids in achieving this diversity. The third and fifth rows of the table demonstrate that clustering enhances the QAP performance from 57.3% to 65.5%. Consequently, these results highlight that all the components together significantly enhance the efficacy of our proposed approach.

5.7.2 Varying the Number of Samples

In Figure 5.4a, we show QAP and local CKA matching accuracies and retrieval scores for different number of base samples M, keeping the number of query samples N constant at 500. It can be observed that as M increases, accuracy/retrieval scores improve, demonstrating the importance of seed initialization for matching algorithms. Figure 5.4b shows the accuracy/retrieval scores as N the number of query samples changes keeping the number of base samples constant at M=320. We see that QAP matching accuracy as local CKA-based retrieval scores decrease with increase in N, but we still get 70% matching accuracy when $\frac{M}{N} = 1$.

5.7.3 Vision and Text Encoders

CKA is measured on combinations of a wide variety of vision and text encoders to examine the impact of: model sizes, dataset regimes, and training paradigms on vision-language alignment. This analysis also identifies the optimal pair of unaligned vision and text encoder for caption-matching tasks. Huggingface's transformers library is utilized for vision models, while the sentence transformers library



(a) **Accuracy and Retrieval Scores** of QAP Matching and Local CKA-based retrieval as the number of base samples is varied, keeping the number of query samples fixed at 500.

(b) **Accuracy and Retrieval Scores** of QAP Matching and Local CKA-based retrieval as the number of query samples is varied, keeping the number of base samples fixed at 320.

Figure 5.4: Comparison of Accuracy and Retrieval Scores for QAP Matching and Local CKA-based retrieval by varying base samples (left) and query samples (right).

is employed for text encoders. Table 5.14 details the vision models, their training data, paradigms, and model types and sizes. Similarly, Table 5.15 presents information on various text encoders. The study covers three training paradigms for vision models: supervised, self-supervised, and language-supervised, with training dataset sizes ranging from 1 million to 400 million images. Text encoders predominantly use sentence transformers, trained for semantic search using a contrastive sentence pairs loss, with dataset sizes varying from 500k to 2B.

Kernel CKA of various model combinations is presented in Table 5.19. The top-performing text encoder trained exclusively on text information is identified as All-Roberta-large-v1 paired with DINOv2, achieving a CKA of 0.706. Consequently, All-Roberta-large-v1 is selected as the text encoder for all tasks and experiments in this chapter, except for cross-lingual experiments. For these, paraphrase-multilingual-mpnet-base-v2 emerges as the most effective text encoder.

Figure 5.5 illustrates the relationship between CKA and text model size across different vision encoder types, training paradigms, and sizes. It is observed that text model size has a limited impact on achieving high CKA with the vision model. Well-trained vision models on large datasets consistently show high kernel CKA with text encoders, regardless of text model size. For instance, language-supervised models (green) and DINOv2 models, which are trained on datasets with hundreds of millions of instances (such as LVD-142's 142 million images and CLIP-400M's

Table 5.14: **Image Encoders Summary.** List of hugging face vision encoder names and information regarding their train data, paradigm, dataset size, model type, and model sizes for the comparison in Figure 5.5 and Table 5.19.

Model Name	Training Data	Training Paradigm	Model Type	Training Data Size	Model Size
facebook\dino-vits8	ImageNet-1k	DinoV1	vit-small	1.2	22
openai\clip-vit-large-patch14-336	CLIP-400M	Language Supervised	vit-large	400	307
facebook\dinov2-base	LVD-142M	DinoV2	vit-base	142	86
facebook\dinov2-small	LVD-142M	DinoV2	vit-small	142	22
facebook\dinov2-large	LVD-142M	DinoV2	vit-large	142	307
facebook\dinov2-giant	LVD-142M	DinoV2	vit-giant	142	1000
openai\clip-vit-base-patch16	CLIP-400M	Language Supervised	vit-base	400	86
facebook\dino-vitb8	ImageNet-1k	DinoV1	vit-base	1.2	86
timm\convnext_base.fb_in1k	ImageNet-1k	Supervised	convnext-base	1.2	89
timm\convnext_tiny.fb_in1k	ImageNet-1k	Supervised	convnext-tiny	1.2	29
facebook\convnext-base-224-22k	ImageNet-21k	Supervised	convnext-base	14.1	89
timm\convnext_base.fb_in22k	ImageNet-21k	Supervised	convnext-base	14.1	89
timm\vit_base_patch16_224.augreg_in21k	ImageNet-21k	Supervised	vit-base	14.1	86
timm\vit_small_patch16_224.augreg_in1k	ImageNet-1k	Supervised	vit-small	1.2	22

400 million image-caption pairs), demonstrate high CKA with language encoders of various sizes.

5.7.4 Other text encoders

Evaluating on COCO with M=320 and N=500, Table 5.16 shows that DINOv2-large achieves high QAP accuracy and retrieval performance when combined with different text encoders. This underscores the potential of pairing well-trained sentence and vision encoders for achieving high semantic similarity between image and text embeddings

5.7.5 Simple projection baseline

We trained a 2-layer MLP on frozen DINOv2-large encoder till convergence using CLIP loss and MSE loss. For fair comparison with our setting, we use 320 training and 500 query image-text samples. Results in Table 5.17 are averaged over 3 seeds. Notably, QAP matching and local-CKA retrieval excel over projection learning, which demands hyperparameter tuning. In contrast, QAP and local-CKA provide a novel, training-free mechanism to evaluate encoder representational similarity, demonstrating effective latent space communication.

5.7.6 Effect of unimodal tasks on alignment

Table 5.18 shows using ViT, DETR, DPT, and SegFormer vision encoders for local-CKA and QAP matching on COCO captions (M=320, N=500). ViT is trained on ImageNet-1k (classification), DETR on COCO 2017 (detection), DPT on 1.4M depth images (depth estimation), and SegFormer is fine-tuned on ADE20k (semantic segmentation). Results indicate that classification models exhibit higher semantic

Table 5.15: **Text Encoders Summary.** List of huggingface text encoder names and information regarding their train data, paradigm, dataset size, and model sizes for the comparison in Figure 5.5 and Table 5.19

Model Name	Model Size	Train Data	Training Paradigm	Training Data Size
all-mpnet-base-v1	109	multiple datasets	contr. sent.	1.12B sent. pairs
gtr-t5-base	110	multiple datasets	contr. sent.	2B sent. pairs
paraphrase-MiniLM-L12-v2	33	multiple datasets	contr. sent.	10M sent. pairs
gtr-t5-large	335	multiple datasets	contr. sent.	2B sent. pairs
all-mpnet-base-v2	109	multiple datasets	contr. sent.	1.12B sent. pairs
average_word_embeddings_komninos	66	Wiki2015	skipgram	2 billion words
average_word_embeddings_glove.6B.300d	120	Wiki2014, GigaWord 5	glove	6 billion tokens
all-MiniLM-L12-v1	33	multiple datasets	contr. sent.	1B sent. pairs
openai_clip-vit-large-patch14	123	CLIP-400M	contr. img-text	400M image-text pairs
all-MiniLM-L12-v2	33	multiple datasets	contr. sent.	1B sent. pairs
all-MiniLM-L6-v2	22	multiple datasets	contr. sent.	1B sent. pairs
sentence-t5-base	110	multiple datasets	contr. sent.	2B sent. pairs
msmarco-distilbert-dot-v5	66	MSMarco	contr. sent.	500k sent. pairs
paraphrase-MiniLM-L3-v2	17	multiple datasets	contr. sent.	10M sent. pairs
paraphrase-albert-small-v2	11	multiple datasets	contr. sent.	10M sent. pairs
all-MiniLM-L6-v1	22	multiple datasets	contr. sent.	1B sent. pairs
all-distilroberta-v1	82	OpenWebTextCorpus	contr. sent.	1B sent. pairs
sentence-t5-large	335	multiple datasets	contr. sent.	2B sent. pairs
All-Roberta-large-v1	355	multiple datasets	contr. sent.	1B sent. pairs
msmarco-bert-base-dot-v5	109	MSMarco	contr. sent.	500k sent. pairs
sentence-t5-xxl	4870	multiple datasets	contr. sent.	2B sent. pairs
paraphrase-TinyBERT-L6-v2	66	multiple datasets	contr. sent.	10M sent. pairs
sentence-t5-xl	1240	multiple datasets	contr. sent.	2B sent. pairs
gtr-t5-xxl	4870	multiple datasets	contr. sent.	2B sent. pairs
paraphrase-distilroberta-base-v2	82	multiple datasets	contr. sent.	10M sent. pairs
gtr-t5-xl	1240	multiple datasets	contr. sent.	2B sent. pairs

Table 5.16: Comparison of CKA, QAP acc. and local CKA retrieval for different text encoders with DINOv2-large image encoder.

Text Encoder	Kernel CKA	QAP Acc.	Ret @ 5
all-roberta-large-v1	0.690	64.93	77.27
paraphrase-distilroberta-base-v2	0.689	65.07	76.33
paraphrase-mpnet-base-v2	0.695	68.20	81.07
sentence-t5-large	0.660	57.87	69.13
sentence-t5-xxl	0.677	63.40	73.00

similarity to all-roberta-large text encoder in QAP accuracy and local-CKA scores than pixel-level tasks such as object detection, segmentation, and depth estimation.

Table 5.17: QAP acc. and Top-5 retrieval scores on COCO.

Method	QAP acc	Ret @ 5
Proj. + MSE	59.8	73.0
Proj. + CLIP	55.4	68.1
QAP	**65.9**	-
Local CKA	64.3	**76.0**

Table 5.18: Unimodal tasks' effect on image-text alignment.

Vision model	QAP acc	Ret @ 5
ViT	35.3	56.1
DETR	26.5	39.8
DPT	22.7	34.1
Segformer	16.8	33.4

5.8 Conclusion

In this chapter, we show that the representations of well-trained vision and language encoders are semantically similar as measured by the CKA metric and QAP performance on a caption matching task, providing an answer to the question **RQ4**

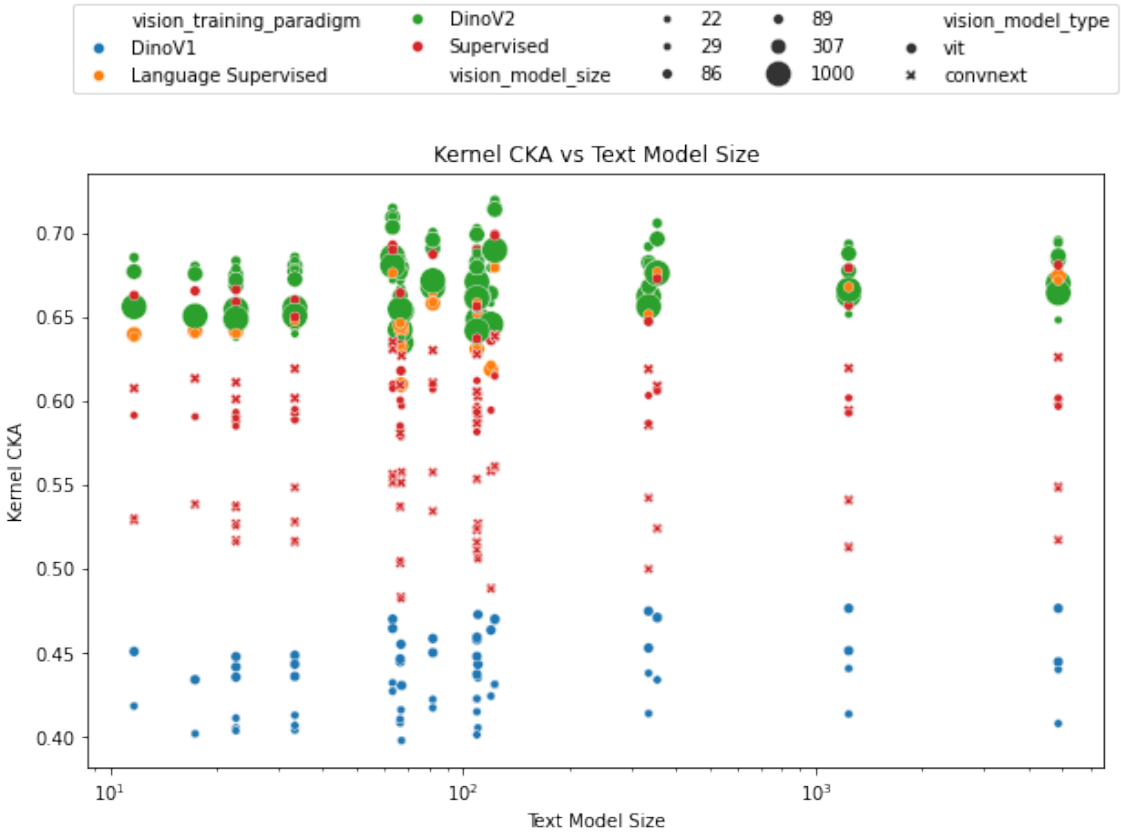


Figure 5.5: **CKA *vs.* text model size** for vision encoders of different training paradigms, model types, and model sizes. We see that text model size is not the most important for high semantic similarity with vision models.

How semantically similar are the representations of vision and text encoders given that they represent the same physical world?. Specifically, we find that well-trained vision encoders exhibit surprisingly high semantic similarity with language encoders, and this semantic similarity is comparable to that between jointly aligned encoders like CLIP. Inspired by this, we draw parallels between CKA and the QAP matching objective and use seeded graph matching to align vision and language encoders by maximizing CKA in a training-free manner. We also devise a local CKA-based metric to enable retrieval between unaligned vision and language encoders demonstrating better performance than other training-free baselines on cross-domain and cross-lingual caption matching/retrieval tasks, essentially facilitating zero-shot latent space communication between unaligned encoders. This work takes an initial step toward leveraging the semantic similarity of unaligned vision encoders to connect them in a label-efficient, training-free manner, offering evidence for our **RQ5**: *Is there a way to connect semantically similar vision and language representations in a training-free manner?*. However, as discussed in Section 5.5.5, the proposed methods rely on test-time search and graph matching algorithms with computational complexities of $\mathcal{O}(n^3)$ and $\mathcal{O}(n^4)$, relative to the number of test samples. This scal-

Table 5.19: **CKA for combinations of different vision and text encoders.** V, V_tr, V_tr_size, V_mod_size stand for Vision model name, Vision train set, Vision train set size, and Vision model size respectively. T_mod_size stands for text model size. OpenAI's CLIP text encoder shows highest CKA with facebook dinoV2base closely followed by All-Roberta-large-v1. We make use of All-Roberta-large-v1 as the language encoder for all donwstream tasks and analysis in main text because All-Roberta-large-v1 has been trained using only text data and can be considered a purely textual encoder.

V	T	CKA	V_tr	V_tr_p	V_tr_size	V_mod_size	T_mod_size
facebook_dinov2-base	openai_clip-vit-large-patch14	0.719	LVD-142M	DinoV2	142	86	123
facebook_dinov2-base	All-Roberta-large-v1	0.706	LVD-142M	DinoV2	142	86	355
timm_vit_base_patch16_224.augreg_in21k	openai_clip-vit-large-patch14	0.698	ImageNet-21k	Supervised	14.1	86	123
facebook_dinov2-large	sentence-t5-xxl	0.684	LVD-142M	DinoV2	142	307	4870
openai_clip-vit-large-patch14-336	All-Roberta-large-v1	0.677	CLIP-400M	Lang. Supervised	400	307	355
facebook_dinov2-large	sentence-t5-large	0.668	LVD-142M	DinoV2	142	307	335
facebook_dinov2-small	sentence-t5-xl	0.661	LVD-142M	DinoV2	142	22	1240
facebook_dinov2-small	all-mpnet-base-v2	0.655	LVD-142M	DinoV2	142	22	109
facebook_dinov2-small	all-MiniLM-L6-v1	0.644	LVD-142M	DinoV2	142	22	22
facebook_convnext-base-224-22k	gtr-t5-xxl	0.626	ImageNet-21k	Supervised	14.1	89	4870
timm_vit_small_patch16_224.augreg_in1k	gtr-t5-xl	0.602	ImageNet-1k	Supervised	1.2	22	1240
timm_convnext_base.fb_in22k	all-MiniLM-L6-v2	0.590	ImageNet-21k	Supervised	14.1	89	22
timm_convnext_tiny.fb_in1k	gtr-t5-xl	0.540	ImageNet-1k	Supervised	1.2	29	1240
timm_convnext_base.fb_in1k	msmarco-bert-base-dot-v5	0.512	ImageNet-1k	Supervised	1.2	89	109
facebook_dino-vitb8	msmarco-distilbert-dot-v5	0.445	ImageNet-1k	DinoV1	1.2	86	66
facebook_dino-vits8	all-mpnet-base-v2	0.423	ImageNet-1k	DinoV1	1.2	22	109
facebook_dino-vits8	paraphrase-TinyBERT-L6-v2	0.398	ImageNet-1k	DinoV1	1.2	22	66

ing makes them impractical for tasks where aligned vision-language encoders like CLIP excel, such as image-caption retrieval, where n can reach thousands or even millions. One approach to align unaligned unimodal encoders for practical tasks is to learn a transformation between their representation spaces. As noted in Section 5.6, certain unimodal encoder pairs exhibit high semantic similarity, prompting us to explore whether this similarity can be leveraged for label- and compute-efficient alignment. In the next chapter, we investigate the transformations required to align such semantically similar unimodal encoders and propose a novel framework for efficient and flexible multimodal alignment of these encoders.

Chapter 6

From Unimodal to Multimodal: Scaling up Projectors to Align Modalities

In Chapter 5, we show that vision encoders, when trained on sufficiently large datasets, produce representations highly similar to those of language encoders, regardless of the training paradigm. We also demonstrate that these representation spaces can be aligned using graph-matching techniques at test time without any training. However, the scalability of these methods is constrained by their computational complexity, which increases significantly with the size of the retrieval set, limiting practical applicability.

A more practical solution is a CLIP-like dual encoder model, which creates a joint embedding space for vision and language, allowing retrieval by computing cosine similarities between the query and the elements of the retrieval set. Such a framework can make use of powerful vision and language unimodal encoders which have been underutilized in multimodal research so far. This approach addresses several limitations of the CLIP model, such as restricted flexibility, limited generalizability to new tasks, and high computational and data requirements. Learning a transformation between frozen embedding spaces demands significantly less compute and data compared to training both encoders from scratch. Moreover, the lower computational needs and availability of strong unimodal vision and language encoders allow flexible multimodal model training for new tasks. For example, a multilingual language encoder can be employed for multilingual classification or retrieval tasks. In this chapter, we investigate the feasibility of this through **RQ6** *Can semantically similar unimodal representations be bridged using simple projection transformations?*. We explore this question using toy experiments and small-scale transfer tests, correlating the CKA between encoder pairs with measures of ease of alignment. Our

findings indicate that unimodal encoders with high semantic similarity, as measured by CKA, can be aligned using simple projection transformations. However, these transformations still fall short on certain tasks, such as zero-shot domain transfer to classification datasets, due to limited concept coverage in the pre-training datasets used in our small-scale experiments. This leads us to our final research question, **RQ7**: *How can we scale up the training of simple projection transformations to align unimodal encoders and achieve efficient, flexible, high-performing CLIP models?* To address this, we focus on scaling projector training by enhancing the concept coverage and quality of training datasets through a concept-balanced data curation approach.

Specifically, we propose a novel framework that aligns vision and language modalities using only projection layers on pretrained, frozen unimodal encoders. This involves three main steps: selecting semantically similar encoders, curating a concept-rich dataset of image-caption pairs, and training simple MLP projectors. We evaluate the quality of our alignment on 12 zero-shot classification datasets and 2 image-text retrieval datasets. Our best model, utilizing DINOv2 and All-Roberta-Large text encoder, achieves 76% accuracy on ImageNet with a 20-fold reduction in data and a 65-fold reduction in compute requirements compared to multi-modal alignment where models are trained from scratch. The proposed framework enhances the accessibility of multimodal model development, enables flexible adaptation to diverse scenarios, and provides an efficient approach to building multimodal models by leveraging advances in existing unimodal vision and language models. Code and the collected datasets were released publicly to foster open source and accessible development of multi-modal models. The work in this chapter is under review as a conference paper at the *Computer Vision and Pattern Recognition Conference* (CVPR) 2025.

The following section presents an overview of aligned encoders like CLIP and strong unimodal encoders as well as an introduction and motivation for our framework. This is followed by 6.2 detailing a literature review on efficient multi-modal pretraining, representational similarity, and automatic data curation methods. In Section 6.3, we introduce motivating visualizations and toy examples for the first part of our framework which makes use of semantic similarity methods to pick the most semantically similar unimodal encoder pairs for alignment. In Section 6.4 we go over the details of the different parts of our framework including our concept balanced data curation method for scaling up projector training. In Section 6.5 we ablate the different components of our framework while in Section 6.6 we perform evaluations on several benchmarks to assess the quality of the alignment function that we learn including 0-shot domain transfer to classification and retrieval tasks. To asses the flexibility of our framework and the quality of unimodal features in the

joint embedding space, we swap different vision and language encoders and evaluate on localization, multi-lingual and long context retrieval tasks.

6.1 Introduction

Contrastive multimodal vision-language models have recently demonstrated impressive zero-shot capabilities [132, 68, 206]. These advancements facilitate the use of language as an API for vision tasks, treating captions as adaptive classes to support a wide range of applications. However, current models face significant challenges: the typical objective function, InfoNCE, is designed to maximize mutual information between the global summary vector of an image and its text representation. This global approach, which relies on pooling functions within the CLIP vision encoder, struggles to deliver the pixel-level granularity required for tasks like segmentation [12]. In contrast, recent advances in uni-modal vision encoders, such as the DINOv2 [121], have demonstrated strong performance in both global and local vision tasks. The CLIP text encoder is limited by its English-only tokenizer and a fixed token length of 77, restricting it's long-context and multilingual retrieval capabilities. Meanwhile unimodal language encoders [140], excel in multilingual, and long-context abilities, as evidenced by improved performance on MTEB benchmarks [111]. Despite these advances in unimodal models, the current strategy for aligning vision and language models usually involves full retraining of vision and language encoders, which is both computationally expensive and inflexible.

This paper proposes a framework for vision-language alignment that efficiently leverages advanced uni-modal vision and language encoders, creating adaptable multimodal models by training only projectors between their frozen embedding spaces. Current efforts to create more efficient CLIP models often compromise on either performance or still require significant resources. For example, LiT [207] achieves comparable results to CLIP but relies on massive compute resources, while smaller-scale models like LiLT [71] may lack sufficient concepts in their training datasets, limiting their zero-shot domain transfer accuracy. Moreover, recent research suggests that LORA fine-tuning of weights of large models using small target datasets improves performance on the target domain at the cost of generalization performance [158].

To address these challenges, our approach builds on recent findings suggesting semantic similarities between well-trained unimodal vision and language embedding spaces [100, 66]. We hypothesize that these similarities enable effective alignment through simple projection transformations, and verify through a toy example in Section 6.3.2 and extensive ablation studies in Section 6.5.1. Inspired by this, our framework includes three key steps: *identifying semantically similar vision-language*

encoder pairs, curating concept-dense datasets, and training lightweight projectors for efficient alignment.

This approach has three practical benefits compared to CLIP-like training:

Strong Unimodal Features lead to Strong Multi-Modal Models Features from uni-modal vision and text encoders are more general than multi-modal trained encoders. For example, it's been shown that vision-only trained encoders perform better on vision-centric tasks when compared to multi-modal vision encoders like CLIP-vision [170]. Hence by keeping these uni-modal encoders frozen and training only projectors for alignment, we aim to keep these strong uni-modal features intact, resulting in better multi-modal representations(See Sec. 6.6.2). **Flexible adaptation to diverse scenarios:** By utilizing the frozen unimodal encoders ability to handle a specific type of data we can efficiently train multimodal models that also can handle this specialized data without the need to retrain the whole network from scratch. For example, multilingual or long context vision-language models can be achieved by aligning DINOv2 with a multilingual (Section 6.6.3) or long-context language text encoder(Section 6.6.4). **Accessible development and Model Reuse:** Relying on already established encoders, projection heads with a dense dataset require significantly less computational resources compared to full model training. In purely practical sense, this approach not only decreases the environmental impact of developing multimodal models but also makes their creation more accessible to the broader research community (Section 6.6.5).

Finally, we evaluate our approach on zero-shot transfer to 12 different classification datasets and 2 image-text retrieval datasets. Our best projector between unimodal models, utilizing DINOv2 and All-Roberta-Large-v1, achieves 76% accuracy on ImageNet, surpassing CLIP's performance while using approximately 20 times less data and 65 times less compute for alignment. We also demonstrate our framework's versatility across tasks like zero-shot domain transfer, multilingual classification, zero-shot semantic segmentation, and image-paragraph retrieval.

Our main contributions lie not in a specific model, but in demonstrating a new framework for vision-language alignment. In summary, we demonstrate that CLIP-like performance can be achieved by training only projection layers, using a curated, concept-rich dataset to enable efficient projector training with significantly less data and compute.

6.2 Related Works

6.2.1 Multimodal Pretraining:

The CLIP models from OpenAI [132] and ALIGN [68] pioneered using web-scale image-caption data to align image and text modalities via an InfoNCE [120] loss, optimizing mutual information between embeddings. LAION [153, 152] replicated this approach in the open domain, open-sourcing pre-training datasets. While these models excel in zero-shot tasks, they demand substantial computational resources, around 20k GPU hours. Taking advantage of the recent improvements in the representation quality of unimodal encoders such as DINOv2 [121] (vision) and Sentence Transformer [139] (language) models, [207] reduce the training cost by locking the image encoder and training only the text encoder to achieve competitive performance. Similarly, [71] further aligned frozen uni-modal encoders using projection layers, BitFit [205], and trainable adapters, but their approach is sub-optimal compared to CLIP, likely due to smaller datasets used and random encoder pair selection. In contrast, we strive to identify the best encoder pairs for alignment first and then scale up projector-only training to improve the multimodal alignment.

6.2.2 Representational Similarity:

Recent studies show that the semantic similarity between vision and language model embeddings is high for several model pairs. [100] reports that this similarity, measured by Centered Kernel Alignment [75], increases with more training data for vision models. Similarly, [66] finds that better-performing language models have higher semantic similarity to the DINOv2 [121] vision model. These similarities have been leveraged for 0-shot and multi-lingual retrieval tasks using strong uni-modal encoders without additional training [100, 110], though scalability is an issue. Additionally, [106] demonstrates that a simple linear mapping allows a frozen language model to interpret visual input, provided the visual encoder aligns with language concepts (e.g., CLIP). These findings suggest that a simple projection transformation separates the embedding spaces of well-trained vision and language models, motivating our work on developing CLIP models using projection layers between semantically similar encoder pairs.

6.2.3 Automatic Data Curation:

Our dataset curation pipeline draws on various approaches in Vision-Language dataset construction [132, 52, 196]. [132] used image metadata to gather high-quality image-caption pairs, while [153] replicated the CLIP dataset by filtering with pre-

trained vision encoders. Recent methods like [52] employ CLIP-based filtering and ad hoc filtering techniques, and [196] mimics CLIP's data collection via metadata retrieval. Similarly, [121] uses a pretrained vision encoder to curate web images most similar to images in curated datasets. Our approach is similar, constructing concept image prototypes from few-shot labeled examples and retrieving relevant web images from the LAION-400M pool using CLIP caption embeddings, avoiding the computational cost of generating vision embeddings for the entire dataset.

6.3 Selecting which encoders to align

Previous studies [66, 100] have shown that well-trained vision and language encoders exhibit high semantic similarity, as measured by Centered Kernel Alignment (CKA) [75] (see Section 6.3.1 for CKA details). A layerwise analysis in [100] reveals that most of this similarity is concentrated in the final projection layer. Additionally, model stitching methods [83, 9, 106] demonstrate that different network regions can be stitched together using linear layers. Inspired by this, in Section 6.3.2, we investigate whether semantically similar encoder embedding spaces can be aligned through a simple projection transformation, using toy examples in the first instance to validate the underlying concept.

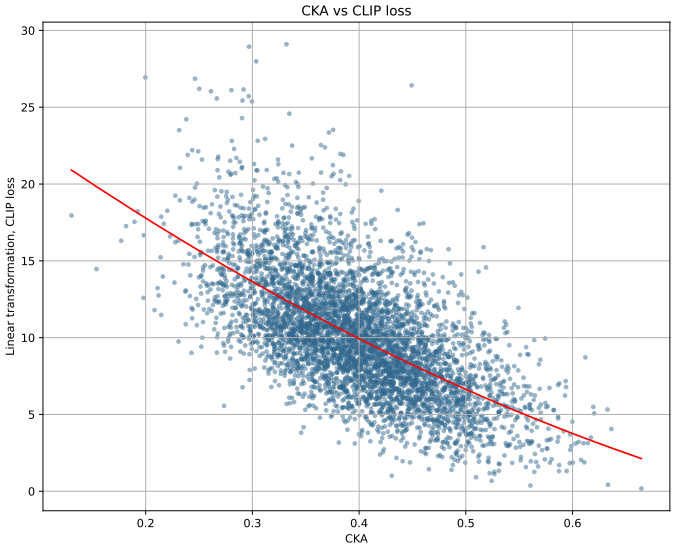


Figure 6.1: **CLIP Loss minima are negatively correlated to CKA.** We plot CKA vs CLIP Loss for instances of A and B that are generated using random non-linear transformations of latent vector Z denoting a representation of the real world.

6.3.1 CKA Preliminary

Centered Kernel Alignment (CKA) has shown its relevance in understanding and comparing the information encoded by different layers of a neural network.

CKA can be defined as follows: Given two sets of vectors X and Y, CKA measures the similarity of these vectors in their respective high-dimensional feature spaces. The kernel matrices K and L are derived from the data sets X and Y, respectively, and represent the inner products between the vectors in these spaces. The entries of K and L are:

$$K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j), \quad L_{ij} = l(\mathbf{y}_i, \mathbf{y}_j)$$

where k and l are kernel functions applied to the vectors $\mathbf{x}_i, \mathbf{x}_j \in X$ and $\mathbf{y}_i, \mathbf{y}_j \in Y$, respectively. Common choices for these kernel functions include linear kernels, where $k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^\top \mathbf{x}_j$, or Gaussian kernels, where $k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2)$ for some $\gamma > 0$.

The CKA coefficient, $\mathrm{CKA}(K, L)$, is defined as:

$$\mathrm{CKA}(K, L) = \frac{\mathrm{HSIC}(K, L)}{\sqrt{\mathrm{HSIC}(K, K) \cdot \mathrm{HSIC}(L, L)}}$$

where HSIC stands for Hilbert-Schmidt Independence Criterion [57, 97], which measures the dependence between the sets of vectors. This measure is invariant to orthogonal transformations and isotropic scaling of the data, making it robust for comparing different models.

6.3.2 CKA vs Ease of Alignment

Centered Kernel Alignment (CKA) [75] measures the similarity of induced graphs of concepts in each encoder space and can act as a guide for selecting encoder pairs that are amenable to alignment. Here we aim to understand the relationship between CKA and the level of difficulty of aligning embedding spaces using toy examples. We define the ease of alignment as the minima of the training loss after convergence. We examine how CKA correlates with the minimum CLIP loss when transforming one vector set to match another using a Linear layer. Since CLIP loss lacks a closed-form solution, we applied SGD for 500 iterations per instance, recording the final loss value as the minimum. We fixed the temperature at 0.07 and the learning rate at 0.01, choosing 500 iterations because the loss value showed minimal change beyond this point.

```
# Init Z with random values scaled to [-1, 1]
Z = 2 * rand(n, d) - 1

# Define non-linear transforms T1 and T2
T1, T2 = NLTransform(d, d), NLTransform(d, d)

# Sample random weights w1 and w2
w1, w2 = rand(1), rand(1)

# Compute A and B using transforms
A = T1(Z) + w1 * rand(n, d)
B = T2(Z) + w2 * rand(n, d)
```

Figure 6.2: Code for initializing A and B

In our experiment, we generated 10^3 instances of two vector sets, A and B, each containing 32 vectors of 16 dimensions. Following the approach in [100, 66], we modeled the world using a latent distribution Z, with Image and Text representations (A and B) as random independent non-linear transformations from Z with additive noise. For each sampled pair of A and B matrices, we calculated the CKA and the minimum CLIP loss. The non-linear transform was defined as a randomly initialized 2-layer MLP with ReLU non-linearity and hidden dimensions significantly larger than the input dimensions, ensuring it could universally approximate the non-linear transformation [63]. Figure 6.2 was used to generate each instance.

Figure 6.1 illustrates the results of this experiment, showing a clear negative correlation between CKA and minima of the CLIP loss. As CKA increases, indicating greater similarity between the similarity structures of A and B, the minima of CLIP loss consistently decreases. Despite arising from a simplified experiment, this strong inverse relationship provides empirical support for the use of CKA as a predictor of alignment potential between different embedding spaces.

6.3.3 Ease of Alignment with real embeddings

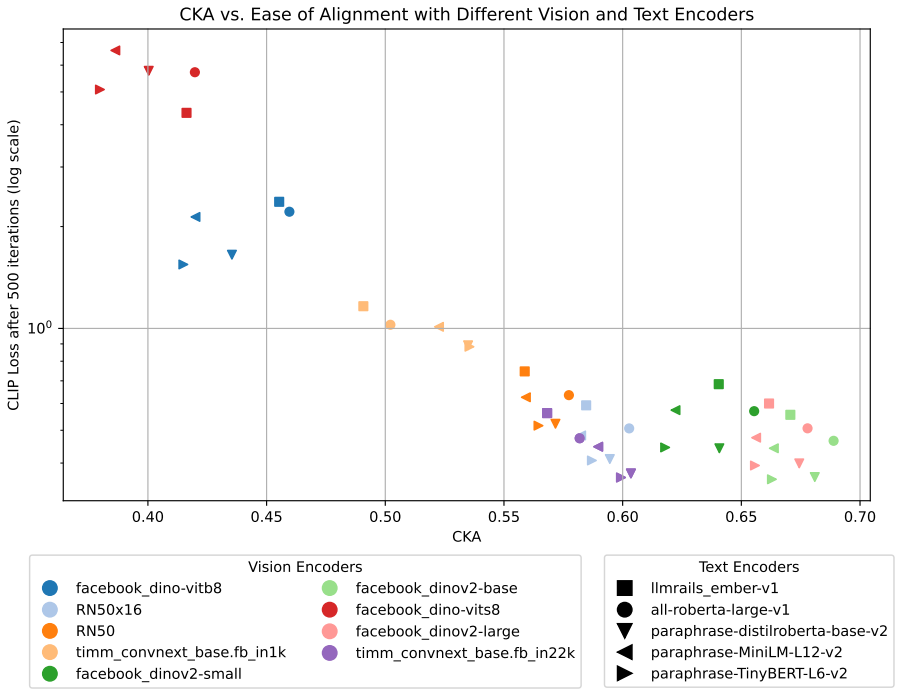


Figure 6.3: **CLIP Loss minima vs CKA for different encoder pairs on a toy image, caption pair dataset**. We plot the CLIP loss after 500 iterations vs CKA for different image, text encoders and find that a negative correlation exists between CKA and ease of alignment.

We go a step further and investigate whether a similar inverse relationship exists between the minima of CLIP Loss and CKA when the embeddings are from real

data and real encoders. We consider five different sentence encoders and nine different vision encoders and sample 5000 different image,caption pairs from the COCO validation set and pass them through all possible encoder combinations to produce 45 different sets of A and B. We then calculate CKA and compute the minima of CLIP Loss after 500 iterations for these A, B and plot them in Figure 6.3 with CKA on the x axis and minima of CLIP loss on the y axis in a log scale. We see that even for real-world embeddings of images and captions there is a strong inverse relationship between CKA and the minima of the CLIP loss providing further evidence that encoders with high CKA could have similar similarity structures making them easy to align using simple projections.

6.3.4 CKA and Graph structure Visualizations

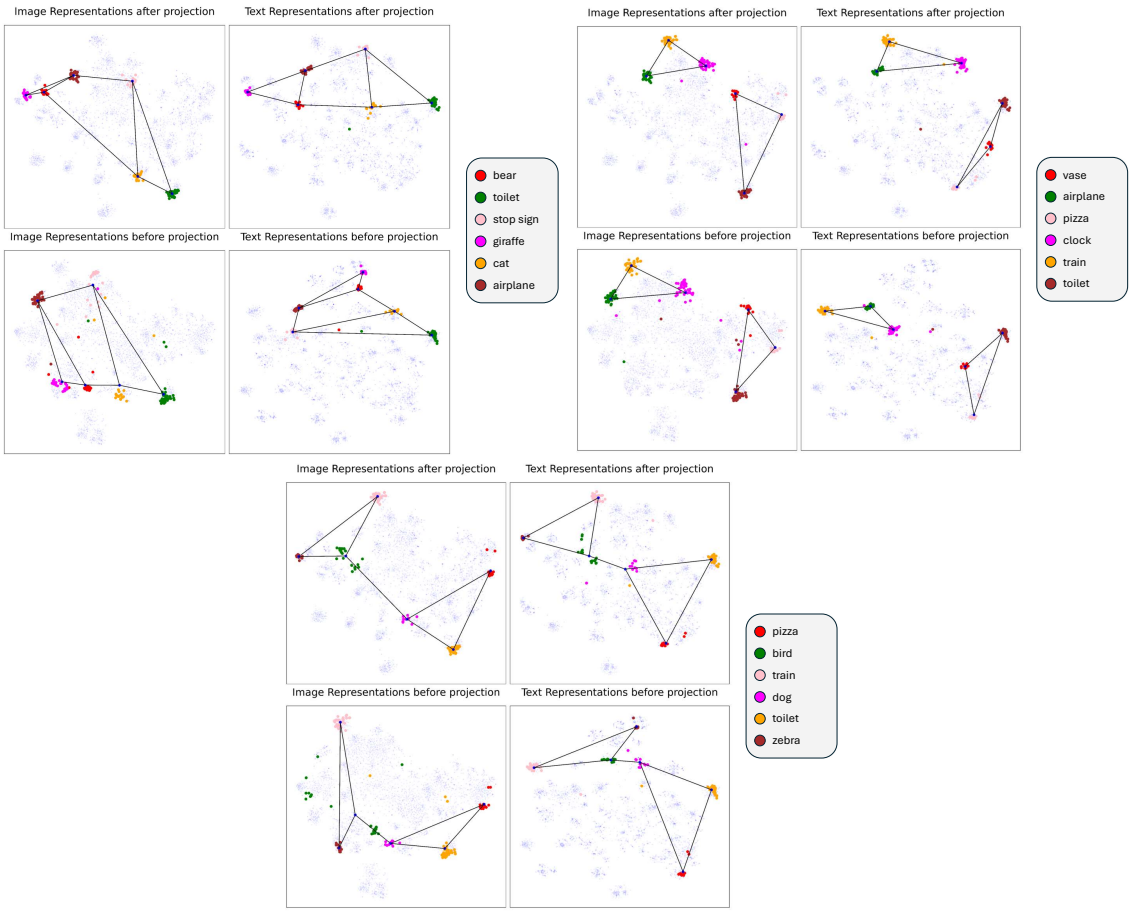


Figure 6.4: TSNE visualizations of encoder outputs for six COCO detection classes. Left: DINOv2 (vision), Right: All-Roberta-Large-v1 (text).

To visually demonstrate how CKA represents similarities in graph structures across different encoder spaces, we conducted an experiment using the MSCOCO validation set. We examined encoder outputs for DINOv2 and All-Roberta-Large-v1, before and after projection, focusing on relationships between formed clusters

in both domains. For each cluster, we identify COCO detection class and COCO image-caption pairs where the image contained only the respective class among its detection annotations. We then extracted encoder outputs for these samples from both vision and text encoders, before and after applying our projection layers, and applied the TSNE algorithm to visualize their structure in a lower-dimensional space. For each visualization, we pick 6 classes to highlight the shape similarities between graphs of encoder spaces.

Figure 6.4 shows the resulting TSNE visualizations for the six selected classes across four conditions: vision pre-projection, vision post-projection, text pre-projection, and text post-projection. The visualizations reveal striking similarities in cluster shapes and relative positions across the different encoder spaces, particularly before projection. This visual similarity aligns with our quantitative CKA results, providing an intuitive illustration of how CKA captures structural similarities between different embedding spaces.

6.4 Framework

Our framework consists of three main components: (1) Encoder Pair Selection, (2) Dataset Curation, and (3) Lightweight Projector Training.

6.4.1 Encoder Pair Selection

Inspired by Section 6.3 we use CKA for selecting the most semantically similar encoder pairs for multimodal alignment. We opted for a linear kernel in the CKA computation after observing that the trends in results were largely consistent between linear and RBF kernels, while the linear kernel offers superior computational efficiency. We measure the CKA between encoder spaces by constructing sets of vision embeddings and text embeddings on the COCO validation set of 5000 image, caption pairs. The COCO validation set is chosen as the reference set for its high semantic alignment between the image content and the caption description. We ablate the use of CKA for encoder pair selection in 6.5.1 and find a positive correlation between CKA and transfer performance to downstream datasets.

6.4.2 Dataset Curation

By only training the projection layers (11M parameters) to align embedding spaces, our approach requires significantly less data compared to training a CLIP model from scratch. However, to ensure high-quality alignment and effective transfer to diverse downstream tasks, it is essential to use a small but well-curated dataset that has the following features. 1. high concept coverage which aids in covering all regions of

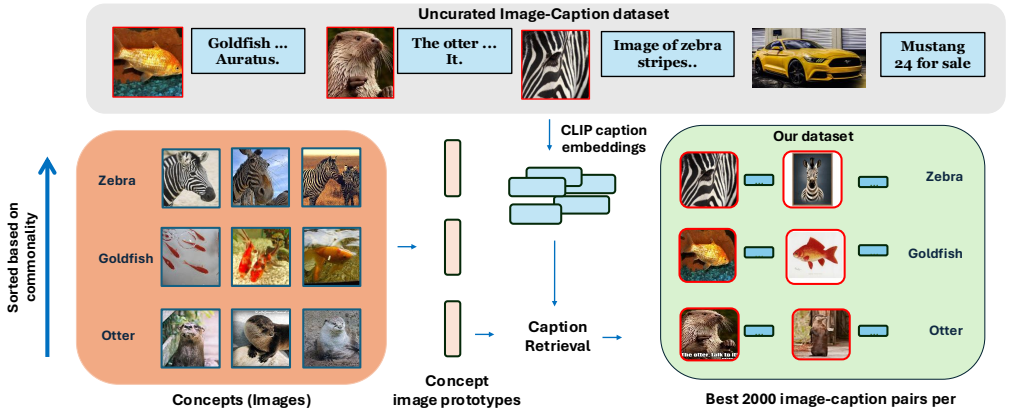


Figure 6.5: **Overview of our concept-balanced dataset curation process.** Images for each concept are acquired from curated datasets and mapped to CLIP embeddings and averaged to construct Image Prototypes for each concept. Captions of the uncurated dataset are mapped to CLIP's joint embedding space and 2000 samples are picked per concept on the basis of the closest caption embeddings to each concept image prototype.

the uni-modal embedding spaces 2. high semantic alignment between image-caption pairs which aids in learning an effective mapping between vision and the embedding spaces. With these requirements in mind, our dataset curation process is structured into two key steps:

Concept Coverage Collection: To ensure high concept coverage, we collect $\sim$ 3000 unique concepts from class names of ImageNet, and several other curated datasets (see A.0.2). Concept image prototypes are then constructed by averaging few-shot image embeddings for each concept using CLIP VIT-Large's vision encoder. To create a class-balanced dataset, we first collect image-caption pairs from LAION400M, a large, uncurated source dataset. We then embed all captions using CLIP ViT-Large's text encoder and compute the caption-image prototype similarity for each concept. To ensure diversity, we retrieve 2,000 samples per concept, starting with the less common concepts. As a proxy to establish the commonality of a concept in the pool, we use the average cosine similarity of the top 25,000 captions closest to each concept prototype. This process results in LAION-CLASS-Collected, a high-quality dataset of 6M samples with broad concept coverage. The detailed algorithm is illustrated in Fig 6.5.

Our primary goal is to compile a concept-rich dataset that enables quick learning and validates the efficacy of projectors for modality alignment, rather than developing a specific curation method. This paper demonstrates the potential of such multimodal models, emphasizing their practicality and efficiency when supported by a dataset with ample concept coverage and robust semantic alignment. The development of an exhaustive dataset that spans all domains of unimodal spaces, ensuring optimal semantic alignment between images and captions, is reserved for

future work.

Retrieval Datasets: The LAION-CLASS-Collected dataset offers high concept diversity, but LAION itself is uncurated, with many captions poorly aligned with their images [45, 116, 20]. While concept coverage is crucial for a dense coverage of the unimodal embedding space, image quality, text diversity, and image-caption alignment are key for effective zero-shot image-text retrieval. In contrast, datasets like CC3M [155], CC12M [18], and SBU [122] feature higher-quality images and better image-caption alignment than LAION. By combining these, we create a 20M MIX-CLASS-Collected dataset that balances concept coverage with image-text similarity, resulting in both dense coverage of the uni-modal embedding spaces as well as high semantic alignment between cross-modal embeddings. We examine the impact of each data source on task performances in Sec 6.5.3.

Data Curation Implementation Details

We streamline our class collection process by precomputing CLIP text embeddings for LAION-400M and CLIP image prototype embeddings for various concepts, allowing us to run different collection methods without needing to recompute embeddings. The embedding process takes just 12 hours on two nodes with 4 A6000 GPUs each. Class-level collection is performed using GPU-accelerated PyTorch code on a single GPU, completing in under an hour. While image-to-image-prototype collection, as in [121], could yield higher-quality results, it demands significantly more GPU resources due to the need to create CLIP embeddings for all LAION-400M images. We find that caption-image-concept similarity performs well for image classification accuracy. To support efficient multi-modal model training, we release the LAION-CLASS-Collected parquets for research use.

6.4.3 Projector Architecture

We train lightweight projectors using contrastive loss between adapted image and text embeddings while keeping the unimodal encoders frozen. Figure 6.6 shows our projector architecture/configuration. We use a lightweight Token Projector [112] with linear and non-linear branches in a residual configuration for both local tokens and the CLS token of each encoder. The projector's weights are shared for local tokens and separate for the CLS token to enable adaptation of both spatial and global information of the vision encoder while limiting the parameter count. Adapted local tokens are averaged and added to the adapted CLS token to form a global embedding, capturing both global and local encoder information. For text encoders, Token Projectors are applied to the tokens, followed by a 2-layer MLP as a global Text Projector, as the text embeddings need further adaptation to become

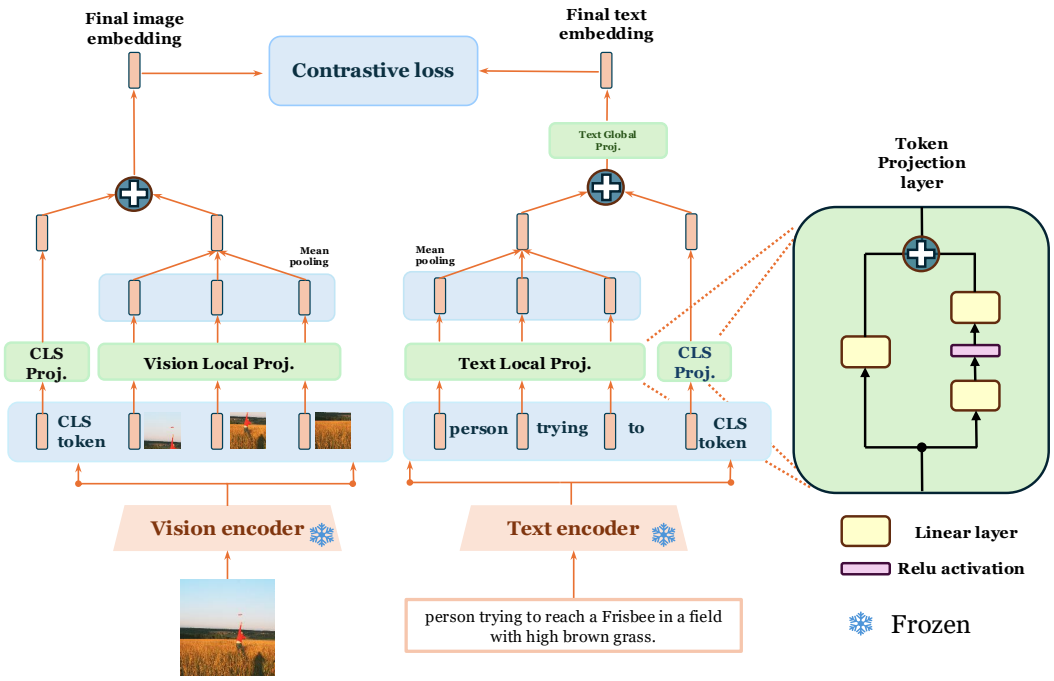


Figure 6.6: **Lightweight Projector Architecture.** We train only Projection Layers to align modalities. Separate projectors are applied on both the local tokens and the CLS token for each encoder and then combined in a residual manner.

more aligned with the vision embeddings. All projector choices are thoroughly ablated in Section 6.5.2. Training information and hyperparameters are detailed below.

Projector training details

We use the standard CLIP loss with a learnable temperature parameter to train the projectors while keeping the vision and text encoders frozen. For our largest experiments on the 20M MIX-CLASS-Collected dataset, we use an effective batch size of 16k and train for 30 epochs. Training is done with a cosine learning rate scheduler, ramping up to 1e-3 in the first epoch. The training process takes 50 hours on a node with 8 A100 GPUs.

6.5 Ablation Experiments

We present a set of ablations to validate different components of our pipeline empirically: CKA for encoder selection 6.5.1, the projector architecture and configuration 6.5.2, the alignment datasets, and the impact of class-collected data 6.5.3. We evaluate on downstream tasks like 0-shot domain transfer to Imagenet classification and COCO / Flickr30k image-text retrieval scores.

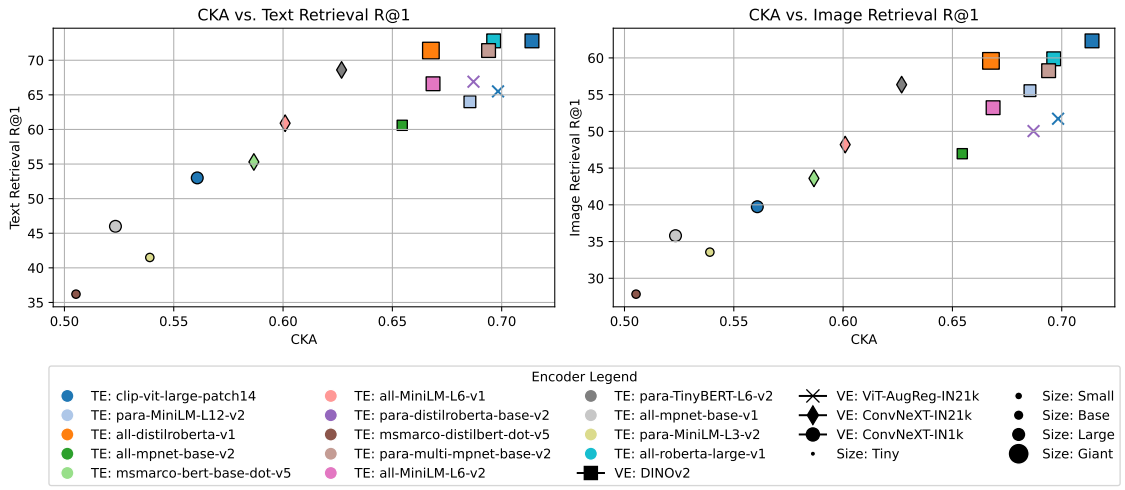


Figure 6.7: **Retrieval performance vs. CKA for different encoder pairs.** Text/Image retrieval accuracies on Flickr30k are compared to CKA, calculated on the COCO val set. Models trained on COCO train set. A clear correlation exists between CKA and alignment quality (Pearson correlation = 0.92, p = 2.1e-7), as reflected in retrieval accuracies.

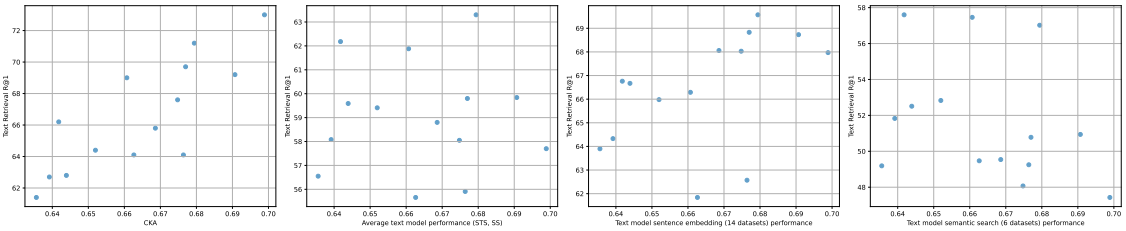


Figure 6.8: **Unimodal performance does not track alignment performance.** Text/Image retrieval accuracies on Flickr30k are compared different text encoder tasks performance. Downstream retrieval performance is more closely correlated with CKA than unimodal text encoder downstream task performances on sentence embedding and semantic search tasks. Models trained on COCO train set.

6.5.1 Effectiveness of CKA for encoder pair selection

We train our projector configurations on various combinations of unimodal encoders using the COCO dataset and evaluate image/text retrieval accuracies on the Flickr30k test set, plotting these against CKA scores in Figure 6.7. The CKA, calculated on the COCO image-caption pairs, shows a strong correlation with retrieval accuracy, indicating that higher semantic similarity, as measured by CKA, predicts better alignment in terms of 0-shot transfer to image/text retrieval on the Flickr dataset. Our findings suggest that CKA can effectively predict which encoder pairs will align well with projector training. The DINOv2-Large and CLIP-ViT-Large-text combination achieves the highest retrieval score, but certain unimodal pairs, like DINOv2-Large and All-Roberta-Large-v1 (CKA = 0.69), perform nearly as well.

This indicates that these unimodal encoders are highly effective for vision-language alignment, leading us to choose the DINOv2-Large and All-Roberta-Large-v1 pair for larger-scale experiments.

Unimodal Performance Does Not Reflect Alignment Quality: A naive approach to choosing the best encoder pair is to chose the unimodal encoders with highest performance in their respective modalities, but it's not straightforward which benchmarks can be more predictive of ease of alignment. To demonstrate this, we consider the same ablation as above, but with DINOv2 and 14 different text encoders from the SentenceTransformers [139] library. We consider 2 types of text model benchmarks. 1. Sentence Embedding task or Semantic Textual Similarity (STS) is the task of evaluating how similar two texts are in terms of meaning. These models take a source sentence and a list of sentences and return a list of similarity scores. The task is evaluated using Spearman's Rank Correlation. We average over 14 datasets reported in [139, 140]. 2. Semantic Search (SS) is the task of retrieving relevant documents or passages based on the semantic content of a query. Rather than relying solely on keyword matching, semantic search models generate embeddings for both the query and the documents, allowing for retrieval based on contextual and conceptual similarity and is evaluated using Normalized Discounted Cumulative Gain (nDCG), which measure the relevance of retrieved documents in ranked lists. We average over 6 datasets reported in [139, 140].

In Fig 6.8, we see that there is a clear correlation (pearson corr.=0.81, p=4e-4) between downstream Flickr30k performance and CKA on the COCO val set, suggesting that CKA is a better predictor of ease of alignment. The average unimodal performance (pearson corr.=0.47, p=0.08), as well as the semantic search (SS) performance (pearson corr.=0.13, p=0.65), are not predictive of the ease of alignment. Meanwhile, Sentence Task Similarity (STS) tasks are more predictive of downstream alignment (pearson corr.=0.72, p=0.003) but still worse than CKA and it's not intuitive which unimodal performance is to be considered.

6.5.2 Impact of Projector Architectures

We ablate our projector combinations (6.1) for the DINOv2 and All-Roberta-Large-v1 encoders by training the projectors to convergence on the LAION-Class-Collected dataset and evaluating the performance on ImageNet 0-shot domain transfer. An MLP applied solely to the local vision tokens achieved 68.81% accuracy, while a Token projection [112] performed slightly better. Therefore, we used the Token projector for all tokens, both visual and textual. Adding projectors to the text side, targeting both text tokens and a global projector on the averaged local tokens (rows 3, 4, and 5), resulted in performance improvements. These projectors help transform

Table 6.1: **Projector Ablations**. We ablate projector combinations for DINOv2 and All-Roberta-Large-v1 encoders, trained on the LAION-Class-Collected dataset and evaluated on ImageNet 0-shot transfer. A Token projector slightly improved over an MLP on vision tokens, while adding text and global projectors boosted performance. The best result (76.12%) was achieved using both CLS and patch projectors, leveraging DINOv2's dual objectives for effective feature learning.

V Proj. Local	V Proj. CLS	T Proj. Local	T Proj. Global	INet 0-shot
mlp	identity	identity	identity	68.81
token	identity	identity	identity	68.84
token	identity	identity	mlp	70.90
token	identity	patch	identity	71.85
token	identity	token	mlp	72.15
identity	token	token	mlp	75.53
token	token	token	mlp	**76.12**

the unimodal text encoder's language-only representations to be more similar to the visual representations. Introducing projectors to the CLS token (row 6) of the visual encoder led to a significant performance increase from 72.15% to 75.13%. Using both CLS and patch projectors in tandem yielded the best performance at 76.12%. This improvement is attributed to DINOv2's dual training objectives: the image-level DINO [17] objective on the CLS token and the patch-level iBOT [214] objective on the patch tokens learning effective global and local features.

6.5.3 Impact of Class-Collected Data / Retrieval Data

In Table 6.2, we ablate the different components of our alignment data. Specifically, we compare the high concept coverage LAION-CLASS-Collected dataset with the high semantic alignment retrieval datasets: CC3M, CC12M, and SBU. Our experiments show that aligning DINOv2 and All-Roberta-Large-v1 on the high concept coverage dataset results in a high ImageNet zero-shot domain transfer accuracy of 76.1 %, though the retrieval accuracies are lower, at 52.7%/42.2% due to the noisy semantic alignment in LAION dataset. In contrast, training with the higher image-caption quality retrieval datasets results in high image and text retrieval scores on the Flickr30k val set (85.3% and 72.4%, respectively). However, the limited concept coverage of these datasets leads to a lower ImageNet accuracy of 54.1%. Combining both types of datasets yields both high ImageNet accuracy and high image/text retrieval accuracies verifying that both dense coverage of unimodal spaces as well as high cross-modal semantic similarity is required to train effective projectors. To ensure that the extra data is adequately utilized, we train for an additional 15 epochs resulting in our best-performing model, achieving an ImageNet accuracy of 76.30%

and Flickr retrieval scores of 87.54%/74.17% (last row).

Table 6.2: **Ablation of Alignment Training Data.**We compare the LAION-CLASS-Collected dataset (high concept coverage) with CC3M, CC12M, and SBU (higher image-caption quality). LAION achieves strong ImageNet accuracy (76.1%) but lower retrieval scores, while the retrieval datasets yield better retrieval performance but lower ImageNet accuracy (54.1%). Combining both datasets with extended training achieves the best results: 76.30% ImageNet accuracy and 87.54%/74.17% image/text retrieval scores

Data Source	N	ImageNet	I2T	T2I
LAION-CLASS-Collected	6M	76.12	52.70	42.48
CC3M, CC12M, SBU	14M	54.17	85.30	72.44
Both	20M	75.04	81.32	71.38
Both longer training	20M	**76.30**	**87.54**	**74.17**

6.6 Results

We evaluate the alignment between vision and text encoders across commonly used VLM benchmarks, including zero-shot domain transfer, image retrieval 6.6.1, localization 6.6.2, multilingual classification/retrieval 6.6.3, and dense caption image-text retrieval 6.6.4. Our goal here is to evaluate the effectiveness of the learned alignment, showcase the flexibility of the framework as well as show that strong task-specific capabilities of uni-modal embeddings are retained in the joint embedding space. We demonstrate that aligning unimodal vision-language encoders can match or exceed the performance of large CLIP models, despite using smaller datasets and less compute. Additionally, our alignment framework is flexible, enabling the use of specialized encoders for specific tasks, such as aligning multilingual text encoders for multilingual or low-resource image classification/retrieval, or long-context text encoders for dense image/caption retrieval. Furthermore, aligning DINOv2 with a text encoder improves image localization beyond CLIP's vision encoder due to DINOv2's superior localization features.

Baselines: We primarily compare our models against CLIP models trained on the OpenAI 400M WIT dataset [132] and the LAION-400M dataset [152]. Notably, our models are trained on a smaller but higher-quality, concept-coverage balanced dataset of 20M-MIX-CLASS-Collected. For multilingual experiments, we also compare against open-clip models trained on the LAION5B dataset[152], which includes image-caption pairs in over 100 languages, as well as multilingual CLIP model [19] that use translated text pairs to align text encoders to CLIP's text encoder space.

Model	N	ImageNet	ImageNetv2	Caltech	Pets	Cars	Flowers	Food	Aircrafts	SUN	CUB	UCF101
LAION-CLIP VIT-L	400M	72.7	65.4	92.5	91.5	**89.6**	73.0	90.0	24.6	70.9	**71.4**	71.6
OpenAI-CLIP VIT-L	400M	75.3	**69.8**	92.6	**93.5**	77.3	**78.7**	**92.9**	**36.1**	67.7	61.4	**75.0**
LiT L16L	112M	75.7	66.6	89.1	83.3	24.3	76.3	81.1	15.2	62.5	58.7	60.0
LilT$_{DA}$-base	0.5M	15.9	12.9	37.6	7.2	1.6	1.1	13.3	1.7	25.6	2.3	19.1
LilT$_{LwA}$-base	0.5M	14.4	12.1	42.3	4.8	1.3	2.1	12.3	1.6	26.5	1.4	26.6
DINOv2-MpNet (Ours)	20M	74.8	68.0	91.8	91.7	71.0	75.8	87.5	23.0	71.9	63.2	71.0
DINOv2-ARL(Ours)	20M	**76.3**	69.2	**92.8**	92.1	73.9	78.4	89.1	28.1	**72.6**	66.1	73.2

Table 6.3: **0-shot domain transfer to classification datasets.** We compare the performance of our DINOv2-ARL projector model, trained on a 20M dataset, against CLIP models from OpenAI and LAION across various datasets. Despite the smaller training size, our model achieves a 76.3% accuracy on ImageNet, outperforming comparably sized CLIP models.

6.6.1 0-shot classification and Retrieval

In this section we aim to evaluate the effectiveness of simple projection transformations in learning an alignment between semantically similar embedding spaces. Tables 6.3 and 6.4 report our model's performance on zero-shot domain transfer to image classification datasets and image-text retrieval on the Flickr30k and COCO datasets, respectively. Detailed descriptions of the evaluation datasets can be found in the A.0.1, highlighting dataset domains, sizes, and prompt descriptions. We see that despite being trained on a 20M dataset our DINOv2-ARL projector model achieves an ImageNet accuracy of 76.3 % which is 1 % and 3.6 % better than comparably sized CLIP models from OpenAI [133] and LAION [153] respectively. Our DINOv2-ARL model demonstrates competitive performance across various datasets compared to LAION and OpenAI CLIP models . The relative performance of these models varies depending on the specific dataset. For example, on the Stanford Cars dataset, LAION-400m [153] CLIP outperforms OpenAI CLIP by a significant margin of over 12%. Conversely, for the Aircrafts dataset, both OpenAI CLIP and our DINOv2-ARL model show superior performance compared to LAION-400m CLIP. We believe this to be due to the differences in concept coverage for these particular datasets between the LAION400m, OpenAI WIT, and our MIX-CLASS-Collected datasets.

In text retrieval, our model outperforms or matches the next best CLIP model, LAION400M-CLIP VIT-L, with scores of 87.5% vs 87.6% on Flickr and 59.7% vs 60.1% on COCO. For image retrieval, our models show a significant advantage, achieving scores of 74.1% vs 70.2% on Flickr and 45.1% vs 43.0% on COCO. This improvement is likely due to the superior quality of the unimodal features produced by the DINOv2 and All-Roberta-Large-v1 encoders, compared to those of the multi-modal vision and text embeddings in the CLIP models. These results demonstrate that simple projector transformations between uni-modal encoders can achieve competitive performance similar to models trained from scratch, providing further evidence that simple projection transformations separate semantically

similar embedding spaces.

Comparison to LiLT

We also report the zero-shot domain classification and retrieval performance of LiLT models [71] and LiT [207] which are the closest parameter-efficient alignment methods to that of our work. The vision encoder in LiLT is initialized with the DeiT base model [172], and the text encoder is from SimCSE [54]. The LilT_{DA}-base model is trained by duplicating and appending the last transformer layer, while only unlocking the last encoder and projector layers. The LilT_{LwA}-base model introduces trainable layerwise adapters for both the vision and text encoders. LiLT public checkpoints are trained on 500k image-caption pairs from the COCO dataset. However, LiLT's performance lags far behind CLIP models and our DINOv2-ARL projector model, primarily due to suboptimal encoder pairs and limited concept coverage in the COCO training set for alignment.

Model	Flickr		COCO	
	I2T	T2I	I2T	T2I
LAION-CLIP VIT-L	**87.6**	70.2	59.7	43.0
OpenAI-CLIP VIT-L	85.2	64.9	56.3	36.5
LiT L16L	73.0	53.4	48.5	31.2
LilT_{DA}-base	47.6	34.46	41.4	29.1
LilT_{LwA}-base	56.8	41.7	47.0	33.7
DINOv2-MpNet (Ours)	84.6	71.2	58.0	42.6
DINOv2-ARL (Ours)	87.5	**74.1**	**60.1**	**45.1**

Table 6.4: **Image, Text Retrieval on COCO/Flickr30k.** Our model shows comparable text retrieval scores and significantly better image retrieval results.

6.6.2 0-shot Localization

One key advantage of leveraging frozen unimodal vision and text encoders is the enhancement provided by unimodal features. Specifically, the DINOv2 vision encoder's robust localization capabilities enhance the joint embedding space of the DINOv2-ARL model when trained solely with projectors. We assess this through zero-shot segmentation performance, similar to the [12, 112], as shown in Table 6.5. Our approach involves computing cosine similarities between each patch and all the ground truth classes and subsequently upscaling to the target size. Each patch is then classified into a corresponding class. Consistent with previous studies, the intersection over union (IoU) is computed solely for the foreground classes. In the zero-shot segmentation process of CLIP models, we employ a technique similar to [213] to alleviate the opposite visualization problem in CLIP models [86]. The patch embeddings from the penultimate layer are passed through the value layer and output MLP of the final self-attention block, followed by projection into the joint

embedding space using the vision projector. Meanwhile, our DINOv2-ARL model considers patch embeddings projected into the joint embedding space by the patch projector and augments them with the projected CLS token in a residual manner.

Our DINOv2-ARL model demonstrates superior performance compared to jointly trained dual encoder models like OpenAI's CLIP, achieving over 8% improvement on Pascal VOC and over 10% on Pascal Context. Notably, models utilizing a fine-grained alignment loss like SPARC [12] show improvements over CLIP. However, our DINOv2-ARL model outperforms SPARC by 4% on VOC and 3% on Context datasets. This underscores that the strong localization abilities of DINOv2 patch embeddings are retained even without training with a fine-grained alignment loss. We hypothesize that the localization performance could also benefit from a more precise localization alignment. Exploring fine-grained losses like SPARC with projector-only models presents an exciting direction for enhancing localization capabilities in VLMs. Additionally, the lower data demands of projector training may allow for the effective use of high-quality, smaller-scale grounding datasets to achieve precise alignment between word tokens and image patches in a supervised manner.

Model	Pascal VOC	Pascal Context
OpenAI-CLIP-VIT-L*	23.46	14.25
SPARC	27.36	21.65
DINOv2-ARL	**31.37**	**24.61**

Table 6.5: **0-shot semantic segmentation mean IOU.** The table shows significant improvements by DINOv2-ARL, even without fine-grained alignment loss. * uses MaskCLIP trick.

6.6.3 Multi-Lingual Results

model	EN	DE	ES	FR	IT	JP	KO	PL	RU	TR	ZH	average
nllb-clip-base@v1	47.2	43.3	44.1	45.0	44.7	37.9	39.4	45.5	40.6	41.2	41.1	42.3
M-CLIP/XLM-Roberta-Large-Vit-B-32	48.5	46.9	46.4	46.1	45.8	35.0	36.9	48.0	43.2	45.7	45.4	43.9
M-CLIP/XLM-Roberta-Large-Vit-L-14	56.3	52.2	52.7	51.8	53.6	41.5	42.5	54.1	48.4	52.7	53.5	50.3
xlm-roberta-base-ViT-B-32@laion5b	63.2	54.5	54.6	55.7	55.7	47.1	43.8	55.5	50.3	48.2	50.8	51.6
nllb-clip-large@v1	59.9	56.5	56.7	56.0	55.5	**49.3**	**51.7**	57.4	50.4	56.0	52.3	54.2
M-CLIP/XLM-Roberta-Large-Vit-B-16Plus	63.2	**61.4**	**59.8**	59.3	**61.0**	48.3	49.8	64.0	54.8	**59.6**	**58.8**	**57.7**
ViT-L-14@laion400m_e31	64.5	26.7	31.4	38.3	26.6	1.4	0.4	4.8	1.7	4.1	1.0	13.6
openai/clip-vit-large-patch14	59.4	19.9	26.6	28.5	19.2	4.1	0.3	3.9	1.3	2.6	0.7	10.7
DINOv2-MpNet (Ours)	**70.7**	60.6	59.0	**60.6**	60.7	45.6	49.8	58.3	52.7	55.8	57.9	56.1

Table 6.6: **Multilingual image-caption retrieval** performance on XTD dataset. DINOv2-MpNet outperforms many baselines despite English-only training. Upper: multilingual-trained models; Lower: English-only trained models.

Similar to the previous section, here we assess whether multi-lingual capabilities of a language encoder is retained when aligned to a vision encoder using projec-

model	EN	AR	ES	FR	DE	JP	ZH	RU	average
nllb-clip-base@v1	25.4	20.4	23.9	23.9	23.3	21.7	20.3	23.0	22.4
nllb-clip-large@v1	39.1	30.1	36.5	36.0	36.2	32.0	29.0	33.9	33.4
M-CLIP/XLM-Roberta-Large-Vit-B-32	46.2	33.4	43.7	43.3	43.3	31.6	29.1	38.8	37.6
M-CLIP/XLM-Roberta-Large-Vit-B-16Plus	48.0	35.1	46.6	45.4	46.1	32.9	31.3	40.3	39.7
xlm-roberta-base-ViT-B-32@laion5b	63.0	29.0	53.4	53.8	55.8	37.3	26.8	40.3	42.3
M-CLIP/XLM-Roberta-Large-Vit-L-14	54.7	**40.0**	51.9	51.6	51.9	37.2	**35.2**	47.4	45.0
ViT-L-14@laion400m_e32	72.3	6.4	44.7	49.9	48.2	2.7	2.3	4.5	22.7
openai/clip-vit-large-patch14	**75.6**	6.7	46.2	49.6	46.7	6.6	2.2	3.5	23.1
DINOv2-MpNet (Ours)	73.4	38.0	56.8	58.3	61.6	**43.2**	33.3	49.3	48.6

Table 6.7: **Multi-lingual classification.** Classification performance comparison of DINOv2-MpNet and various CLIP models and multilingual baselines on multilingual ImageNet. Our DINOv2-MpNet model trained only on English data outperforms even models trained on multi-lingual data. The upper half of the table lists models trained on multiple languages, while the lower half lists models trained only on English data. The models are evaluated on translations of the labels and the prompts made using nllb-200-distilled-600M translation model. [29]

tors. We demonstrate this by aligning DINOv2-Large with paraphrase-multilingual-MpNetv2 (referred to as MpNet), chosen for its high CKA compatibility, using only English image-caption pairs and evaluating model performance on multi-lingual image retrieval on the XTD dataset [4] and classification on the ImageNet dataset. For multi-lingual classification, we translate our VDT prompts [102] to the languages being considered using the nllb-700M model [29] and then use the same prompts for all the models being considered.

For both multi-lingual classification and retrieval tasks, our comparisons are structured into two categories as delineated in Table 6.7 and Table 6.6. The lower sections of each of these tables list models trained exclusively with English captions, more specifically the CLIP-VIT-L models from OpenAI and LAION trained on 400 million image caption pairs of WIT dataset [132] and LAION400M [153] dataset respectively. The upper sections of these tables feature models trained with translated/multi-lingual captions, including those employing contrastive training with multi-lingual image-caption pairs such as CLIP-models based on the LAION5B [152] multi-lingual dataset, which contains image-caption pairs in over 100 languages. We also compare against, M-CLIP [19] models that are trained using English and translated captions to align a multi-lingual text encoder with CLIP's original text encoder through contrastive learning, thereby enhancing performance on multi-lingual tasks. Additionally we also compare against the NLLB-CLIP [180] models developed through LiT [207] techniques, coupling a frozen CLIP visual encoder with an unfrozen multi-lingual text encoder using translated captions from the smaller LAION-COCO dataset. We compare against only model sizes of up to ViT-Large for fair comparison.

Retrieval results: Our model DINOv2-MpNet trained only on English im-

age,caption pairs outperforms all other CLIP models trained only on English image caption pairs, by a large margin of over 43 % on average retrieval performance over 10 languages. We also outperform the next best performing English CLIP model trained on LAION400m English caption retrieval by over 6 percent. On Latin script languages the CLIP models have decent performance while it falls significantly for non Latin languages like JP, KO, PL, RU, TR, and ZH. This is mainly because these models were trained using an English only tokenizer which results in unknown token for most characters of these languages. However our DINOv2-MpNet projector model maintains competitive performance on all languages both Latin script and non Latin script even when compared against models specifically trained using multi-lingual data (Upper half of the table). Amongst the multi-lingual trained CLIP models we perform better than laion5b trained xlm-roberta-base-VitB32 by 4.5 percent. It is to be noted here that we only use 20 million Image caption pairs for alignment while LAION5B has over 5B image-caption pairs/ It is to be noted that our DINOv2-Mpnet is also competitive with M-CLIP model XLM-Roberta-Large-Vit-B-16Plus(56.1 vs 57.7) which has been trained using translated English sentences of over 175 million data points to over 100 languages, and 3M translated image, caption pairs from CC3m.

Classification results: We see a similar trend when we compare our DINOv2-MpNet projector model against CLIP baselines(lower section), and multi-lingual baselines (upper section) on multi-lingual imagenet classification in Table. Our model showcases competitive performance to that of OpenAI-clip model while beating LAION400m trained ViT-Large on english Imagenet, while performing significantly better on all other languages considered (over 24 percent better on 8 language average). When compared with models trained with multi-lingual data, our model outperforms both nllb-clip models as well as M-CLIP models, beating the next best performing model M-CLIP/XLM-Roberta-Large-Vit-L-14 by over 3 percent despite not training using any multi-lingual text data. We believe that training using translated image-caption pairs of our dataset would further improve the performance of our method, and we leave this as a future work. The main advantage of training using our methods is that we can get highly porformant CLIP-like models using much lesser amount of image-caption pairs, (more than 20x lesser) resulting in quick adaptation to low resource languages given that a multi-lingual text encoder exists for that language.

In summary, DINOv2-MpNet's robust multilingual performance, achieved without any multilingual training data, demonstrates that MpNet's capabilities are preserved in the joint embedding space through effective projector training of unimodal models.

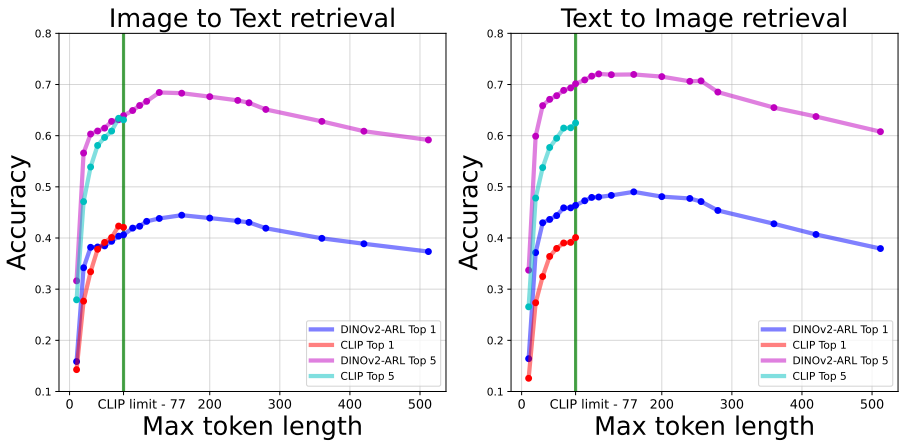


Figure 6.9: Retrieval performance comparison between DINOv2-ARL encoder pair and OpenAI CLIP as the maximum token length increases. The vertical green line indicates the standard CLIP token limit of 77.

6.6.4 Densely Captioned Images (DCI) Dataset and Long-Text Retrieval

We assess whether the ARL model maintains its long-context capabilities in the joint embedding space by conducting image and long caption retrieval on the Densely Captioned Images (DCI) dataset [177], which features caption pairs averaging over 1,000 words. Unlike DCI's benchmarks that use summarized captions (see 6.6.4), we focus on full image-text and text-image retrieval tasks without summarization or subcropping, enabling a comprehensive evaluation of our framework's long-text retrieval capabilities.

To demonstrate the retention of long-context ability, we conducted an experiment varying the maximum token length allowed by the tokenizer. As shown in Figure 6.9, our DINOv2-ARL encoder pair achieves comparable performance to OpenAI CLIP at the standard limit of 77 tokens. However, our approach's strength becomes evident as we extend beyond this limit, with consistent improvement in retrieval accuracy up to approximately 200-300 tokens. Given that DINOv2-ARL was trained with short-context image-caption pairs, these results underscore the model's ability to retain long-context capabilities in the aligned joint embedding space.

sDCI benchmark results

We also evaluate our method on the original benchmark suite of Densely Captioned Images (DCI) dataset [177] in this section. To accommodate current models' token limits, the authors also provide sDCI, a summarized version with CLIP-compatible 77-token captions generated by LLMs.

sDCI introduces several benchmarks:

- All SCM (Subcrop-Caption Matching): Matches captions to corresponding

Model	All SCM	All Neg	All Pick5-SCM	All Pick5-Neg	Base Neg	All Hard-Negs
CLIP Baseline	**40.06**%	60.79%	**11.21**%	**24.06**%	67.56%	41.34%
DINOv2-ARL (Ours)	29.33%	**64.36**%	9.35%	21.39%	**81.94**%	**61.10**%

Table 6.8: Performance comparison on DCI dataset benchmarks

image subcrops.

- All Neg: Distinguishes between positive captions and LLM-generated negatives.
- All Pick5-SCM: Similar to All SCM, but uses multiple captions per subcrop.
- All Pick5-Neg: Distinguishes between multiple positive captions and a negative.
- Base Neg: Focuses on caption-negative distinction for full images only.
- All Hard-Negs: Uses the most challenging LLM-generated negatives.

We tested our DINOv2-ARL model on the sDCI dataset benchmarks. Table 6.8 presents our results alongside the CLip baseline. Our method demonstrates competitive performance compared to the CLIP baseline across several DCI benchmarks.

In the Subcrop-Caption Matching tasks (All SCM and All Pick5-SCM), our model performs slightly below the CLIP baseline. This suggests that there is room for improvement in our approach when it comes to distinguishing between the different parts that compose an image.

However, our model shows notable improvements in the negative detection tasks. We outperform CLIP on All Neg (64.36% vs. 60.79%), Base Neg (81.94% vs. 67.56%), and All Hard-Negs (61.10% vs. 41.34%). These results demonstrate the potential of our method in aligning vision and language models for a fine-grained understanding of image content, especially in scenarios requiring robust discrimination between relevant and irrelevant captions. Future work could focus on improving the model's performance on sub-crop caption matching tasks while maintaining its strong capabilities in negative detection.

6.6.5 Alignment Compute

We report the Alignment Training compute requirements for different models in 6.9. We see that aligning pre-trained vision, language encoders to get a competitive CLIP like model requires only 50 hours of training with 8 A100 GPUS which is almost a 65 fold reduction in the amount of training compute. This makes the development of multi-modal models accessible to the wider research community

Model	Data	SS	Trainable / Total	Compute	IN 0-shot
OpenAI CLIP	400M	12.8B	427M / 427M	21,845	72.7%
LAION400M CLIP	400M	12.8B	427M / 427M	25,400	75.3%
DINOv2-ARL	20M	0.6B	11.5M / 670M	400	76.3%

Table 6.9: **Compute requirements, Dataset size, and Number of trainable parameters are orders of magnitude lower when using projectors to align semantically similar encoders.** By using projectors to align semantically similar encoders, compute requirements drop 65-fold, dataset size shrinks by 20 times, and only 1% of total parameters are trainable while outperforming other CLIP models. Compute measured in GPU hours on an A100 (80 GB) GPU.

as well as reducing the environmental impact of training highly performant multi-modal models by reusing strong publicly available uni-modal models. Since we only need to train 11.5M of the total 670M parameters (about 1 %) we can train with a much smaller and denser dataset reducing the data requirements to 20M which is 20 fold decrease in dataset requirement compared to CLIP models from LAION and OpenAI making our framework useful for training performant multi-modal models in various domains like mutli-modal systems for low-resource languages, 3D model search systems, fMRI to Image model mapping systems and many more. Despite the reduced compute and data requirements for alignment our model outperforms both CLIP models compared on domain transfer to Imagenet as well as image, text retrieval.

6.6.6 Multi-lingual 0-shot Semantic Segmentation

The lower compute and paired data requirements of the framework lead to application flexibility simply by swapping the unimodal encoders. (see Sec. 6.2-6.4 in the main paper). An additional advantage of this flexibility is showcased in Fig. 6.10 and Tab. 6.10, where we use our aligned DINOv2-MpNet to perform multi-lingual semantic segmentation. Our segmentation scores stay consistent with different languages while CLIP often fails on non-english languages.

Language	CLIP	*DINOv2-MpNet*
EN	23.46	29.07
ES	18.86	28.69
ZH	8.46	28.06
FR	15.12	28.48
DE	21.30	27.91
RU	5.72	26.85

Table 6.10: Comparison of CLIP and *DINOv2-MpNet* performance across languages.

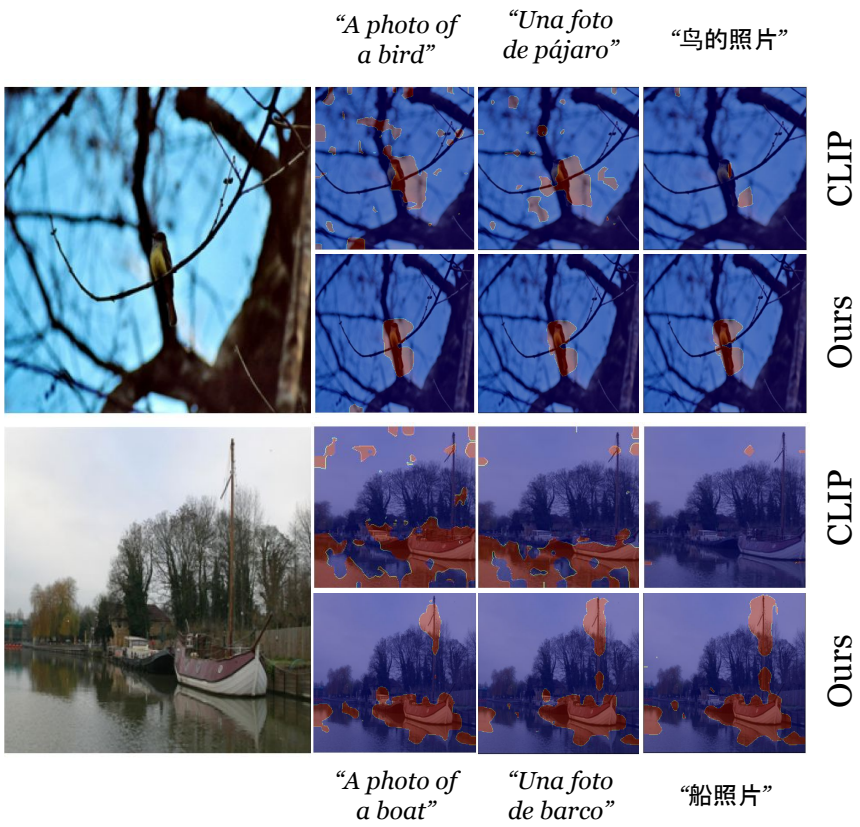


Figure 6.10: Compared to CLIP, our approach of aligning DINOv2-MpNet achieves improved segmentation maps focusing on the relevant objects in the multilingual setting.

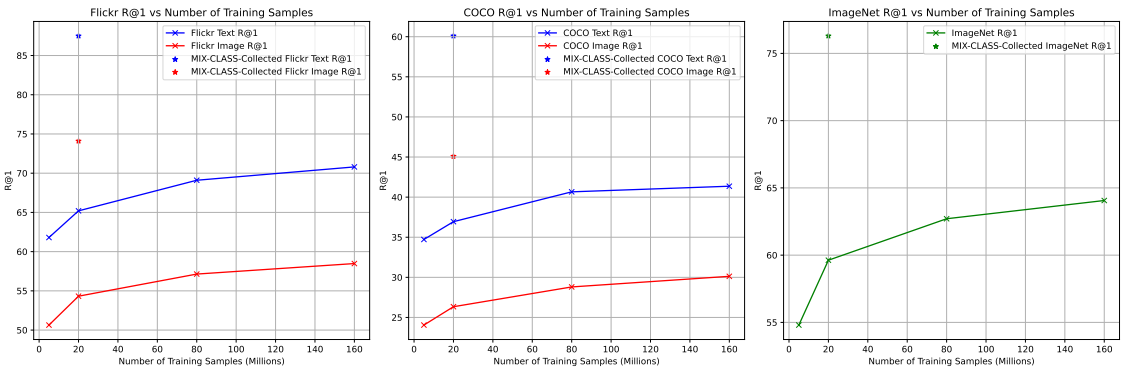


Figure 6.11: **Performance scales with higher amounts of randomly sampled LAION data** The performance scales with higher amounts of randomly sample data from LAION400M, but very slowly, highlighting the need for a densely covered and high quality dataset when training projectors only to align modalities.

6.6.7 Dataset Scale Ablation

Figure 6.11 illustrates that while performance scales with an increasing number of randomly sampled data points from the LAION400M dataset, the rate of improvement diminishes, highlighting the critical need for densely covered and high-quality datasets when training projectors to align modalities. Additionally, the comparative performance of MIX-CLASS-Collected data reveals that datasets curated with more focused criteria can lead to better performance gains than simply increasing the volume of data. This underscores the importance of prioritizing dataset quality over quantity, especially given the observed diminishing returns when using larger data sizes for projector-based alignment.

6.7 Conclusion

In this chapter, we show that semantically similar unimodal representations are separated by simple projection transformations through toy experiments and small-scale alignment transfer experiments. Section 6.3 provides preliminary evidence that representation spaces with higher CKA are easier to align, while Section 6.5.1 demonstrates that this correlation holds in small-scale alignment experiments, where models are trained on one dataset (COCO) and tested for retrieval on another (Flickr30k). These results address our research question **RQ6**: *Can semantically similar unimodal representations be bridged using simple projection transformations?* However, we found that these learnt representations underperform when transferred to zero-shot classification tasks, mainly due to limited concept coverage in the alignment dataset. To improve the generalizability of the learnt projections, the alignment dataset must comprehensively cover the domain of the unimodal encoder spaces. Therefore, we incorporated a concept-balanced data curation strategy in our framework (Section 6.4) to ensure diverse concept coverage.

Our evaluations on zero-shot domain transfer and retrieval (Section 6.6.1) using projectors trained on the curated dataset show that the learnt transformations effectively preserve the unimodal features of the constituent encoders in the joint embedding space. This results in a dual-encoder CLIP model that is both compute- and data-efficient, while also being flexible and generalizable across diverse scenarios, effectively addressing our final research question **RQ7**: *How can we scale up training of simple projection transformations to align unimodal encoders and achieve efficient, flexible CLIP models?*

The retention of strong unimodal features in our joint embedding space allows us to seamlessly swap the language encoders with multilingual or long-context encoders, thereby demonstrating flexibility and generalizability to tasks such as multilingual

classification/retrieval (Section 6.6.3) and long-context retrieval (Section 6.6.4). In the final section, we show that aligning strong unimodal encoders requires orders of magnitude less compute and data compared to training models from scratch, thereby democratizing multimodal alignment and making it more environmentally friendly. In summary, we address the challenges of the CLIP model identified in Chapter 4 by studying representational similarities of vision and language encoders (Chapter 5) and developing an efficient alignment framework in this chapter

There is significant scope for future work in this line of research such as exploring fine-grained alignment techniques, optimizing projection architectures, and expanding to other modalities beyond vision and language. The reduced dataset requirements on alignment data makes our framework useful for training performant multimodal models in various domains where strong unimodal encoders exist like multimodal systems for low-resource languages, 3D model search systems, fMRI to Image model mapping systems and many more. By democratizing multimodal AI research, our framework has the potential to accelerate innovation and reshape approaches to multimodal AI development.

Chapter 7

Conclusion and Future Work

In this final chapter, we consolidate the key findings of our research, emphasizing our contributions and their significance. We revisit the hypotheses and research questions from Chapter 1, evaluating how effectively they have been addressed. This summary provides important insights, discusses practical implications, and proposes future research directions. Our goal is to offer a clear and concise conclusion that underscores the value of this work.

7.1 Hypotheses and Answers to the Research Questions

This section revisits the hypotheses introduced in Chapter 1, examining them in relation to the research findings presented in the subsequent chapters. Each research question linked to these hypotheses is explored to clarify the specific contributions of this thesis.

Hypothesis 1 (H1): Semantic information about the world is suited for improving generalizability and sample efficiency of vision models, because they are a more robust representation of the world that can guide learning of visual representations when the visual data is limited.

(RQ1) : *How can we incorporate semantic information to improve the performance of few-shot learning? More specifically, can we use base to novel concept relationships from domain expert annotated semantic information to facilitate learning novel concepts with few examples?*
In Chapter 3, we propose a simple semantic querying method that uses semantically similar base dataset instances to refine novel support instances, producing robust prototypes for few-shot learning. We demonstrate that incorporating semantic infor-

mation into computer vision models enables efficient learning of new concepts using visual features from similar base concepts. Specifically, we design a cross-attention mechanism that attends to semantically similar base features to construct improved prototypes for novel concepts. Our method enhances few-shot learning performance across three benchmark datasets (see Section 3.4, Tables 3.1, 3.2, 3.4), validating our hypothesis that domain expert-annotated semantic information provides a robust representation of the world, and base to novel concept relationships from these sources can be leveraged to improve the generalizability and sample efficiency in computer vision models (answering **RQ1**).

Hypothesis 2 (H2): LLMs have a good world model and text generated by LLMs can be a good proxy for extracting semantic information about the world that can be used to guide the generalizability and sample efficiency of CLIP models.

RQ2 *How can LLMs be used to generate semantic information useful for computer vision models in a scalable manner?*

In Chapter 4 we show that text information that describes the classes visually or visually descriptive text (VDT) (in contrast to non VDT) is effective in improving the 0-shot domain transfer performance of CLIP [132] to fine-grained datasets like CUB [184] (See Table 4.1) that were not prevalent in their pre-training. VDT sentences effectively map the novel concepts into concepts that the CLIP's text encoder already understands. We design a 2-step prompting strategy to extract VDT sentences from LLMs like GPT4, and GPT3.5 in a scalable manner and show its effectiveness by improving the 0-shot domain transfer of CLIP on an eclectic suite of 12 datasets (See Table 4.3) effectively addressing **RQ2**.

.

RQ3 *How can LLM generated semantic information be used to improve 0-shot and few-shot domain transfer performance of vision language foundation models like CLIP?*

In Chapter 4, we utilize generated semantic information to enhance the zero-shot and few-shot domain transfer performance of vision-language models (VLMs). We use scalable VDT generated by LLMs like GPT-3.5 and GPT-4 to construct more informative class prototypes for zero-shot classification, achieving significantly better performance than CLIP on 12 datasets. Additionally, we design an attention-based VDT adapter to select the most relevant visually descriptive information for each class using the few labeled examples available in the few-shot setting. This

approach outperforms both prompt tuning and state-of-the-art adaptation methods such as CoOp [216], CoCoOp [215], and CLIP-Adapter [53], effectively addressing **RQ3** (see Table 4.5). We also study the impact of different LLMs on the quality of generated VDT and find that smaller LLMs are prone to hallucinations, which degrades zero-shot transfer performance. However, our attention-based adapter effectively extracts useful information even from these less reliable sources (see Table 4.7).

Hypothesis 3 (H3): Given that vision and language are trying to model the same physical reality, the representations from vision and language encoders should demonstrate high similarity in terms of semantics.

RQ4 *How similar are vision and language representations given that their encoders are trying to model the same physical reality?*

In Chapter 5, we design experiments and metrics to measure the semantic similarity between unimodal representations of vision and language encoders. We show that well-trained vision and language encoders exhibit highly similar semantic structures, as measured by CKA. Moreover, we demonstrate that the similarity between vision encoder representations and those of a language encoder scales with the quantity and quality of training data, regardless of the training paradigm (e.g., self-supervised, supervised, language-supervised) used (see Figure 5.2). These findings validate our hypothesis that vision and language encoders have high semantic similarity because they model the same physical reality. Additionally, we find that certain unimodal vision and language encoder pairs achieve semantic similarity levels comparable to jointly trained vision-language models like CLIP, providing an answer to **RQ4**.

RQ5 *Is there a way to connect semantically similar vision and language representations in a training-free manner?*

Based on the understanding that well trained vision and language encoders have very similar representation structures, in the latter half of Chapter 5, we propose two novel methods to align uni-modal encoders in a training free manner by framing alignment as a graph matching problem. We show that with as few as 320 image, caption pairs, it's possible to perform retrieval and classification between two unaligned unimodal encoders that exhibit high semantic similarity (See Tables 5.2, 5.3, 5.4, 5.5) providing an answer to **RQ5**.

Hypothesis 4 (H4): Given that well-trained vision and language en-

coders have semantically similar representation spaces, simple projection transformations might be sufficient to bridge them.

RQ6 *Can semantically similar unimodal representations be bridged using simple projection transformations?*

In Chapter 6, we demonstrate that semantically similar unimodal representations can be bridged by simple projection transformations, as shown through toy experiments (Section 6.3.2) and small-scale alignment transfer experiments (Section 6.5.1). Section 6.3 provides preliminary evidence that representation spaces with higher CKA are easier to align, indicating that the achievable CLIP loss minimum depends on the initial semantic similarity between image and language encoder spaces. Additionally, Section 6.5.1 confirms this correlation in small-scale alignment transfer experiments, where models are aligned on one dataset (COCO) and tested for retrieval on another (Flickr30k). These findings demonstrate that semantically similar unimodal encoders can be effectively bridged using simple projection transformations, answering **RQ6**.

RQ7 *How can we scale up the training of simple projection transformations to align unimodal encoders and achieve performant, flexible and compute/data efficient CLIP models?*

In our small-scale transfer experiments, we found that the learnt representations underperform in zero-shot classification tasks, primarily due to limited concept coverage in the alignment dataset. To improve the generalizability of the learnt projections, the alignment dataset must comprehensively cover the domain of the unimodal encoder spaces. Therefore, we incorporated a concept-balanced data curation strategy in our framework (Section 6.4) to ensure diverse concept coverage. Our evaluations on zero-shot domain transfer and retrieval (Section 6.6.1), using projectors trained on the curated dataset, show that the learnt transformations effectively preserve the unimodal features of the constituent encoders in the joint embedding space. This results in a dual-encoder CLIP model that is both compute- and data-efficient, while also being flexible and generalizable across diverse scenarios. Building upon these observations, we designed a framework to develop a multimodal dual-encoder model from two strong unimodal encoders by only training lightweight projection transformations. CLIP models trained using our proposed framework outperform CLIP baselines on both image-text retrieval and zero-shot classification tasks, using 20 times less data and 40 times less compute. Additionally, due to its low compute and data requirements, and the retention of strong unimodal features in the joint-

embedding space, our framework is highly flexible as we use it to train performant CLIP models for multi-lingual image classification/retrieval (Tables 6.7 6.6), 0-shot image localization (Table 6.5) and long context image-text retrieval (Figure 6.9). These experimental results validate that we can scale up projector training between unimodal encoders to achieve performant, flexible and compute/data efficient CLIP models addressing **RQ7**.

7.2 Research Contributions

The contributions of this research are collected and summarized in the following list:

- Chapter 3: Semantics from language for One Shot Learning
    1. An approach leveraging semantic information from domain experts, semantic graphs (e.g., WordNet), and language models to enhance few-shot learning by improving support instance representations.
    2. A novel method BaseTransformers that uses part-based composition and aggregation that matches semantically meaningful parts between novel instances and well-supported base dataset samples, enabling the construction of robust prototypes for novel categories.
    3. State-of-the-art performance achieved on three benchmark few-shot datasets, with experimental results demonstrating the effectiveness of BaseTransformers across different backbone architectures in the inductive one-shot setting.
- Chapter 4: Language semantics for 0-shot and few-shot adaptation of foundational VLMs
    1. A demonstration that including visually descriptive textual (VDT) information in prompts improves CLIP's 0-shot domain transfer performance, particularly in fine-grained classification tasks.
    2. A scalable approach using GPT-4 to generate VDT sentences, resulting in consistent 0-shot performance improvements over CLIP's default prompt across multiple datasets, without requiring domain-specific annotations.
    3. A simple adapter network that leverages VDT information to enhance few-shot transfer performance, achieving better results than methods like CLIP-Adapter [53] and CoCoOp [215] in the Base-to-New setting.
    4. Public release of the generated VDT information for 12 datasets, supporting further research in prompt engineering and adapter design for low-shot domain adaptation in vision-language models.

- Chapter 5: Semantic Representational Similarity between Vision and Language Encoders and 0-shot alignment of Unimodal Vision and Language Encoders

    1. An analysis of semantic similarity between well-trained vision and language encoders, demonstrating that even without explicit alignment, these models capture semantically comparable representations of the physical world, as evidenced by high Centered Kernel Alignment (CKA) scores between unaligned encoders.
    2. A matching method that maximizes semantic similarity between vision and language encoders by finding the optimal permutation of captions to maximize CKA. This is achieved through a novel formulation of CKA maximization as a quadratic assignment problem, with enhanced transformations and normalizations for improved matching performance.
    3. A local CKA metric for retrieval tasks, enabling zero-shot retrieval between two unaligned embedding spaces, demonstrating superior performance on COCO caption-image retrieval compared to existing relative representation methods.
    4. Comprehensive benchmarking on COCO, NoCaps, and ImageNet-100 datasets for cross-domain caption-image retrieval and classification tasks, validating the effectiveness of our zero-shot approach for communication between latent spaces without explicit alignment.
    5. A practical application of cross-lingual image retrieval, using sentence transformers trained in multiple languages and a CLIP vision encoder trained exclusively in English, showcasing the adaptability of our method across languages and domains.

- Chapter 6: Efficient Alignment of Unimodal Encoders

    1. An analysis of semantic similarity and alignment potential, showing that high Centered Kernel Alignment (CKA) scores between vision and language encoders indicate greater ease of alignment. Our experiments reveal a strong inverse relationship between CKA and the minimum CLIP loss, suggesting that encoders with higher CKA values have more compatible embedding structures and can be aligned more effectively using simple projections.
    2. A framework for creating CLIP-like models by training only lightweight projection layers on top of semantically similar uni-modal vision and language encoders, achieving comparable performance to CLIP models from

OpenAI and LAION with 20 times reduction in data and 65 times reduction in compute requirements.

3. A novel approach to dataset curation, collecting a dense and concept-rich set of image-caption pairs from uncurated sources, supporting effective projection training and accessible multi-modal model development. This curated dataset is made publicly available to facilitate further research.
4. Diverse application of the framework across several tasks showcasing flexibility and, demonstrating strong performance in zero-shot domain transfer, multilingual classification, zero-shot semantic segmentation, and image-paragraph retrieval, showcasing the flexibility of our framework
5. An accessible and environmentally friendly method for multi-modal model creation, reducing the computational burden by focusing on lightweight projector training, thus making high-performance vision-language alignment more accessible to a broader research community.

7.3 Recommendations for Future Research

There are a number of opportunities for future research related to the ideas explored in this thesis, and these are outlined in the following.

- Semantic Querying for 0-shot and few-shot Domain Transfer
  - In BaseTransformers (Chapter 3) we make use of semantically similar data from the base dataset to improve few-shot learning of new concepts. With CLIP, there has been a paradigm shift in few-shot and 0-shot approaches, but a similar idea of using semantically relevant images and text from the Pre-Training dataset could potentially help CLIP's fine-grained classification performance. This information could be queried from the pre-training data and combined into the prompt structure and query/support images of the few-shot dataset to improve the domain transfer to fine-grained datasets where CLIP underperforms, anologous to the Retrieval augmented generation approaches common in LLMs for performance on tasks distant from their pre-training data.
  - In VDT-Adapter (Chapter 4), we show that CLIP models can be transferred to fine-grained datasets by making use of semantics from large-language models. The LLM generated visually descriptive text is selected and aggregated using an attention mechanism that is learned on the few-shot support set during training. However, the performance on several fine-grained datasets like CUB, and FGVC Aircrafts is still subpar at 74

% and 35 % respectively. This could be because of a lack of fine-grained alignment between the VDT sentences and parts of the image. Introducing an adapter that learns fine-grained alignment between different regions of the image and the different sentences, by making use of supervisory signals from an open-world detector has the potential to make CLIP models more adaptable to fine-grained datasets in a few-shot manner. This would also lead to more interpretability as the weights of the adapter could be probed to see which parts of the image are important for making the classification.

- Analysis of Representational Similarity to other modalities, decoders.
  - In Chapter 5, we show that well-trained unimodal vision and language encoders have high semantic similarity in their representations despite being not trained together. We believe this to be because the vision and language modalities are essentially trying to model the same physical world and their performance can be a noisy measure of how close they are to this latent space. However, it's not apparent if this is also valid for encoders of other modalities like audio and 3d data. Studying representational similarity between other modalities would provide further evidence for or against the platonic representation hypothesis [66] as well as result in insights that could lead to the design of efficient architectures for connecting them in a label / compute efficient manner. This would have high impact in modalities where paired data is hard to come by, For example, simulation, real-world paired images for robotics.
  - In Chapter 5, we limit our analysis to language encoders, as extracting a good representation from a language decoder remains challenging. Concurrent work [66] conducts similar semantic analysis on language decoders by aggregating semantic similarity scores across layers and selecting the maximum, akin to BrainScore [149, 150]. However, obtaining a reliable representation of text from a language decoder is still an open research problem. Investigating this through the lens of semantic similarity with strong vision encoders could provide insights into which transformer layers and attention heads in a language decoder contain the most information and help develop techniques to derive a global representation from LLMs. This is significant, as the largest language models today are decoders, which are orders of magnitude larger and more information-rich than encoder models specialized for sentence embeddings. Developing a robust global representation from LLMs could enhance generalization capabilities and promote model reuse.

  - Investigating semantic similarity metrics between a unimodal encoder and a unimodal decoder across different modalities is an under-explored area with the potential to provide valuable insights for designing effective projectors and connection modules for integrating multiple modalities into LLM decoders. This could be especially beneficial in scenarios with limited paired data, such as 3D point cloud and text data.

- Parameter and Data Efficient Multi-Modal Alignment

  - In Chapter 6, we present an effective framework for aligning unimodal encoders efficiently, though our exploration is limited to a small set of vision and language encoders. Since then, considerable advancements have been made in language encoders, with larger models achieving higher performance on MTEB tasks [111]. It would be valuable to investigate whether these improvements facilitate easier alignment. Additionally, applying our framework to modality pairs with limited paired data, such as point cloud and text, could be an interesting direction for multimodal alignment. Another potential avenue is aligning vision encoders specialized for semantic segmentation with language models to develop robust open-world segmentation systems.

  - In Chapter 6, we show that DINOv2 and ARL models demonstrate superior spatial localization capabilities with the standard CLIP loss, due to the inherent localization strength of DINOv2 features. Recent CLIP-based studies have investigated enhancing spatial localization using fine-grained losses [12]. By integrating our framework with a fine-grained image-to-text token alignment loss, we can further improve fine-grained localization, enhancing CLIP models' performance on compositionality [204] and image-text localization benchmarks [69].

  - Combining the strengths of DINOv2 and CLIP into a single model could significantly benefit VLLMs by creating a unified vision encoder capable of handling all VLM tasks. DINOv2 excels at general visual tasks, while CLIP is particularly effective for OCR, enabling better text extraction from images. Recent research has explored multi-encoder approaches [170, 171], but a unified model that leverages both capabilities could streamline VLLM development, improving multimodal understanding and performance on benchmarks that require both vision and language comprehension.

7.4 Closing Remarks

In this thesis, we sought to address key challenges in visionmodels by leveraging semantic information and studying the representational relationships between vision and language encoders. Our findings demonstrate that incorporating semantics, whether sourced from domain experts or generated by large language models, significantly enhances the generalization and efficiency of vision models, particularly in low-data and open-world scenarios.

We also showed that well-trained unimodal encoders for vision and language exhibit high semantic similarity, paving the way for efficient alignment through lightweight projection transformations. This work contributes to making the development of multimodal models more accessible and environmentally friendly by reducing compute and data requirements while enhancing model flexibility and performance.

Our exploration of representational similarity and effective alignment has opened new directions for creating more robust, scalable, and adaptable models capable of understanding and processing the complexities of our richly multimodal world. By systematically addressing compute, data efficiency, flexibility, and generalization, this thesis takes significant strides toward developing vision-language systems that more closely resemble human multimodal perception, contributing to the broader pursuit of human-like artificial intelligence.

We hope that these insights inspire further advancements in the development of more efficient, flexible, and powerful vision-language models, ultimately bridging the gap between human-like perception and artificial intelligence.

Appendix A

Appendix

Table A.1: Comparing our VDT with that of descriptors from [104] for 2 random classes of datasets DTD and Eurosat

	Ours	**DCLIP[104]**
Stratified (DTD)	'The surface feels moderately smooth, with slight roughness due to the layered structure.'	'a series of layers'
	'There is no distinct pattern, but the layers create a natural, linear visual effect.'	'each layer is of a different material'
	'The structure is characterized by multiple layers stacked upon each other.'	'the layers are parallel to each other'
	'The texture has a two-dimensional feel, with the layers adding a sense of depth.'	'the layers may be of different thicknesses'
	'The density varies, with some layers appearing closely packed while others are more sparse.'	'the layers may be of different colors'
	'The regularity of the texture is defined by the consistent layering.'	'the layers may have different textures'
	'The texture is opaque, with no transparency between the layers.'	
	'There are no significant surface defects, but minor irregularities may occur between layers.'	

Continued on next page

Table A.1 – continued from previous page

	Ours	**DCLIP[104]**
Lined (DTD)	'The texture feels moderately smooth to the touch, not too rough nor too sleek.'	'a series of parallel lines'
	'It exhibits a lined pattern, reminiscent of ruled notebook paper.'	'can be straight or curved'
	'The structure of the texture is stratified, with lines arranged one after the other.'	'may be of different colors'
	'The texture has a two-dimensional quality, with no noticeable depth or relief.'	'may be of different widths'
	'The lines are densely packed, leaving little space between them.'	'may be of different thicknesses'
	'The texture displays a high degree of regularity, with the lines evenly spaced and parallel.'	
	'The texture is opaque, with no transparency or translucency.'	
	'There are no noticeable surface defects, the lines are clean and uninterrupted.'	
Industrial (Eurosat)	'Industrial buildings have texture that is smooth, regular.'	'evidence of human activity'
	'Industrial buildings have shape that is rectangular, irregular.'	
	'Industrial buildings have size (relative) that is large.'	
	'Industrial buildings have pattern that is regular, dense.'	
	'Industrial buildings have spectral reflectance that is high in visible spectrum.'	
	'Industrial buildings have a shadow that is present (due to high-rise buildings).'	

Continued on next page

Table A.1 – continued from previous page

	Ours	**DCLIP[104]**
	'Industrial buildings have adjacent land features that is commercial, residential, roads.' 'Industrial buildings have change over time that is stable.' 'Industrial buildings have density that is high.' 'Industrial buildings have proximity to water bodies that is variable.' 'Industrial buildings have road accessibility that is high.'	
Forest (Eurosat)	'Forest has texture that is rough.' 'Forest has shape that is irregular.' 'Forest has size (relative) that is large.' 'Forest has pattern that is no pattern.' 'Forest has spectral reflectance that is high in near-infrared.' 'Forest has shadow that is present (due to trees).' 'Forest has adjacent land features that is land, mountains, rivers.' 'Forest has change over time that is mostly stable.' 'Forest has density that is high.' 'Forest has proximity to water bodies that is variable.' 'Forest has road accessibility that is low.'	'a large area of trees' 'green leaves'

A.0.1 0-Shot Classification and Retrieval Evaluation Datasets

To evaluate the performance of our DINOv2-ARL projector model and compare it with baseline CLIP models, we utilized a diverse set of datasets for zero-shot classification and retrieval tasks. These datasets span various domains and challenge the models' ability to generalize across different visual concepts.

For zero-shot classification, we employed the following datasets:

- ImageNet [33]: A large-scale dataset with 1000 object categories, widely used as a benchmark for image classification tasks. It contains over 1.2 million training images and 50,000 validation images, with each image labeled with one of 1000 object classes.

- ImageNetV2 [138]: A newer version of ImageNet designed to test the robustness of models trained on the original ImageNet. It features 10,000 new test images collected using the same procedure as the original, but addressing certain biases in the original dataset.

- Caltech101 [47]: A dataset containing pictures of objects belonging to 101 categories, plus a background category. It includes about 40 to 800 images per category, with most categories having about 50 images. The dataset is known for its high intra-class variability.

- Oxford-IIIT Pet [126]: A 37-category pet dataset with roughly 200 images for each class, featuring different breeds of cats and dogs. It includes pixel-level trimap segmentations and breed-level labels for each image.

- Stanford Cars [76]: A dataset of 196 car classes, totaling 16,185 images. Classes are at the level of Make, Model, Year (e.g., 2012 Tesla Model S). It includes 8,144 training images and 8,041 testing images, with bounding box annotations.

- Oxford Flowers102 [117]: A 102 category dataset consisting of 102 flower categories common to the UK. It contains 40 to 258 images per class and provides segmentation data for each image. The dataset is particularly challenging due to the fine-grained nature of the categories.

- Food101 [14]: A large dataset of 101 food categories, with 101,000 images. It features 1000 images per food class, with 250 test images and 750 training images per class. The training images are not manually cleaned, adding a level of noise to the dataset.

- FGVC Aircraft [98]: A fine-grained visual classification dataset with 10,200 images of aircraft, spanning 100 aircraft models. Each model is associated with a specific variant, manufacturer, family, and collection. The dataset includes 6,667 training images and 3,333 test images.

- SUN397 [143]: A scene recognition dataset with 397 categories and 108,754 images, covering a large variety of environmental scenes under various lighting conditions. It provides at least 100 images per class and has been used extensively for scene recognition tasks.

- Caltech-UCSD Birds-200-2011 (CUB) [184]: A dataset for fine-grained image classification with 200 bird species, containing 11,788 images. Each image has detailed annotations including 15 part locations, 312 binary attributes, and 1 bounding box. It's widely used for fine-grained visual categorization research.

- UCF101 [163]: An action recognition dataset with 101 action categories, consisting of realistic action videos collected from YouTube. It contains 13,320 videos from 101 action categories, with videos exhibiting large variations in camera motion, object appearance and pose, illumination conditions, and more.

For zero-shot image-text retrieval, we used:

- Flickr30k [129]: A dataset containing 31,783 images collected from Flickr, each paired with 5 crowd-sourced captions. It focuses on describing the objects and actions in everyday scenes. The dataset is split into 29,783 training images, 1000 validation images, and 1000 test images.

- COCO [89]: A large-scale dataset for object detection, segmentation, and captioning, which we use for its image-caption pairs in the retrieval task. It features over 330,000 images, each with 5 captions. The dataset includes 80 object categories and instance segmentation masks, making it versatile for various computer vision tasks.

These datasets comprehensively evaluate a model's ability to perform zero-shot classification across various domains and its capacity for cross-modal retrieval. By using this diverse set of benchmarks, we can assess the generalization capabilities of our approach compared to existing CLIP models. We use Visually Descriptive Class-Wise prompts from [102] to enable the unimodal-text encoder in our DINOv2-ARL projector model to better identify the zero-shot classes of the downstream datasets.

A.0.2 Concept Coverage Collection datasets

We use a few shot examples from 14 curated computer vision datasets to construct our Concept Image prototypes to curate the images from our uncurated data pool. The 14 curated datasets are described as follows.

- BirdSnap [11]: A fine-grained dataset consisting of 49,829 images of 500 North American bird species. The images are annotated with species labels, and the dataset is primarily used for species classification and fine-grained recognition tasks.

- Caltech101 [47]: A dataset containing pictures of objects belonging to 101 categories, plus a background category. It includes about 40 to 800 images per category, with most categories having about 50 images. The dataset is known for its high intra-class variability.

- EuroSAT [62]: A satellite image dataset with 10 categories related to land use classification (e.g., forests, rivers, residential areas). It contains 27,000 labeled images, with 2700 images per class, widely used in remote sensing and geospatial tasks.

- FGVC Aircraft [98]: A fine-grained classification dataset with 10,000 images of 100 aircraft model variants from 70 manufacturers. It is used for distinguishing between visually similar objects in fine-grained recognition tasks.

- Flowers102 [117]: A dataset containing 102 flower categories, commonly used for fine-grained classification tasks. It has a total of 8,189 images, with 40 to 258 images per category, and is organized into a training, validation, and test set.

- Food101 [14]: A dataset containing 101,000 images of 101 food categories. Each category has 750 training images and 250 test images, commonly used for food classification and recognition tasks.

- GTSRB [164]: The German Traffic Sign Recognition Benchmark dataset, containing over 50,000 images of 43 different traffic sign classes. It is designed for multi-class classification tasks in the context of traffic sign recognition.

- ImageNet [33]: A large-scale dataset with 1,000 object categories, widely used as a benchmark for image classification tasks. It contains over 1.2 million training images and 50,000 validation images, with each image labeled with one of 1,000 object classes.

- Oxford Pets [126]: A dataset of 7,349 images, containing 37 categories of pets (both cats and dogs). Each image is annotated with species and breed information, commonly used for image classification and segmentation tasks.
- RESISC45 [24]: A dataset of remote sensing images used for scene classification, containing 31,500 images across 45 scene classes. Each class has 700 images with variations in resolution, scale, and orientation.
- Stanford Cars [76]: A dataset with 16,185 images of 196 car models, annotated by make, model, and year. The dataset is designed for fine-grained classification and recognition tasks of vehicles.
- Pascal VOC 2007 [44]: A dataset for object detection, segmentation, and classification, containing 9,963 images of 20 object categories. It is widely used for benchmarking models in computer vision tasks.
- SUN397 [143]: A large-scale scene understanding dataset with 397 categories and 108,754 images. It covers a wide range of environments, from natural to man-made scenes, commonly used for scene classification tasks.
- UCF101 [163]: A video dataset consisting of 13,320 videos across 101 human action categories. It is widely used for action recognition tasks in video analysis and computer vision research.

Bibliography

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Mohamed Afham, Salman Khan, Muhammad Haris Khan, Muzammal Naseer, and Fahad Shahbaz Khan. Rich semantics improve few-shot learning. *British Machine Vision Conference*, 2021.

[3] Arman Afrasiyabi, Jean-François Lalonde, and Christian Gagné. Associative alignment for few-shot image classification. In *European Conference on Computer Vision*, pages 18–35. Springer, 2020.

[4] Pranav Aggarwal and Ajinkya Kale. Towards zero-shot cross-lingual image retrieval. *arXiv preprint arXiv:2012.05107*, 2020.

[5] Harsh Agrawal, Karan Desai, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, Stefan Lee, and Peter Anderson. Nocaps: Novel object captioning at scale. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8948–8957, 2019.

[6] Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. Label-embedding for attribute-based classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 819–826, 2013.

[7] Richard Antonello, Javier S Turek, Vy Vo, and Alexander Huth. Low-dimensional structure in the space of language representations is reflected in brain responses. *Advances in Neural Information Processing Systems*, 2021.

[8] Sungyong Baik, Myungsub Choi, Janghoon Choi, Heewon Kim, and Kyoung Mu Lee. Meta-learning with adaptive hyperparameters. *Advances in Neural Information Processing Systems*, 33:20755–20765, 2020.

[9] Yamini Bansal, Preetum Nakkiran, and Boaz Barak. Revisiting model stitching to compare neural representations. *Advances in Neural Information Processing Systems*, 2021.

[10] Serguei Barannikov, Ilya Trofimov, Nikita Balabin, and Evgeny Burnaev. Representation topology divergence: A method for comparing neural network representations. *arXiv preprint arXiv:2201.00058*, 2021.

[11] Thomas Berg, Jiongxin Liu, Seung Woo Lee, Michelle L Alexander, David W Jacobs, and Peter N Belhumeur. Birdsnap: Large-scale fine-grained visual categorization of birds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2011–2018, 2014.

[12] Ioana Bica, Anastasija Ilic, Matthias Bauer, Goker Erdogan, Matko Bošnjak, Christos Kaplanis, Alexey A Gritsenko, Matthias Minderer, Charles Blundell, Razvan Pascanu, et al. Improving fine-grained understanding in image-text pre-training. In *Forty-first International Conference on Machine Learning*, 2024.

[13] Lisa Bonheme and Marek Grzes. How do variational autoencoders learn? insights from representational similarity. *arXiv preprint arXiv:2205.08399*, 2022.

[14] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *Computer vision- 13th European Conference on Computer Vision 2014, zurich, Switzerland, September 6-12, 2014, proceedings, part VI 13*, pages 446–461. Springer, 2014.

[15] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[16] Luana Bulat, Stephen Clark, and Ekaterina Shutova. Speaking, seeing, understanding: Correlating semantic models with conceptual representation in the brain. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1081–1091, 2017.

[17] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021.

[18] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3558–3568, 2021.

[19] Guanhua Chen, Lu Hou, Yun Chen, Wenliang Dai, Lifeng Shang, Xin Jiang, Qun Liu, Jia Pan, and Wenping Wang. mclip: Multilingual clip via cross-lingual transfer. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13028–13043, 2023.

[20] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. Sharegpt4v: Improving large multi-modal models with better captions. In *European Conference on Computer Vision*, pages 370–387. Springer, 2024.

[21] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pages 1597–1607. PMLR, 2020.

[22] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *International Conference on Learning Representations*, 2019.

[23] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[24] Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017.

[25] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

[26] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Conference on Computer Vision and Pattern Recognition*, 2014.

[27] Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*, 2017.

[28] Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Algorithms for learning kernels based on centered alignment. *The Journal of Machine Learning Research*, 13(1):795–828, 2012.

[29] Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*, 2022.

[30] David F Crouse. On implementing 2d rectangular assignment algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, 52(4):1679–1696, 2016.

[31] Adrián Csiszárik, Péter Kőrösi-Szabó, Akos Matszangosz, Gergely Papp, and Dániel Varga. Similarity and matching of neural network representations. *Advances in Neural Information Processing Systems*, 34:5656–5668, 2021.

[32] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee, 2005.

[33] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2009.

[34] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2009.

[35] Fatma Deniz, Anwar O Nunez-Elizalde, Alexander G Huth, and Jack L Gallant. The representation of semantic information across human cerebral cortex during listening versus reading is invariant to stimulus modality. *Journal of Neuroscience*, 39(39):7722–7736, 2019.

[36] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[37] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[38] Carl Doersch, Ankush Gupta, and Andrew Zisserman. Crosstransformers: spatially-aware few-shot transfer. *Advances in Neural Information Processing Systems*, 33:21981–21993, 2020.

[39] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer,

Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[40] Jon Driver and Toemme Noesselt. Multisensory interplay reveals crossmodal influences on 'sensory-specific'brain regions, neural responses, and judgments. *Neuron*, 57(1):11–23, 2008.

[41] Kshitij Dwivedi and Gemma Roig. Representation similarity analysis for efficient task taxonomy & transfer learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12387–12396, 2019.

[42] Gerald M Edelman. *Neural Darwinism: The theory of neuronal group selection.* Basic books, 1987.

[43] Gerald M Edelman and Joseph A Gally. Reentry: a key mechanism for integration of brain function. *Frontiers in integrative neuroscience*, page 63, 2013.

[44] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111:98–136, 2015.

[45] Lijie Fan, Dilip Krishnan, Phillip Isola, Dina Katabi, and Yonglong Tian. Improving clip training with language rewrites. *Advances in Neural Information Processing Systems*, 36, 2024.

[46] Nanyi Fei, Zhiwu Lu, Tao Xiang, and Songfang Huang. Melr: Meta-learning via modeling episode-level relationships for few-shot learning. In *International Conference on Learning Representations*, 2020.

[47] Li Fei-Fei, R. Fergus, and P. Perona. 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, 2004.

[48] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017.

[49] Donniell E Fishkind, Sancar Adali, Heather G Patsolic, Lingyao Meng, Digvijay Singh, Vince Lyzinski, and Carey E Priebe. Seeded graph matching. *Pattern recognition*, 87:203–215, 2019.

[50] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic

embedding model. *Conference on Neural Information Processing Systems*, 2013.

[51] Andreas Fürst, Elisabeth Rumetshofer, Johannes Lehner, Viet T Tran, Fei Tang, Hubert Ramsauer, David Kreil, Michael Kopp, Günter Klambauer, Angela Bitto, et al. Cloob: Modern hopfield networks with infoloob outperform clip. *Advances in neural information processing systems*, 35:20450–20468, 2022.

[52] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, et al. Datacomp: In search of the next generation of multimodal datasets. *Advances in Neural Information Processing Systems*, 36, 2024.

[53] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *International Journal of Computer Vision*, 132(2):581–595, 2024.

[54] Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*, 2021.

[55] Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, Patrick Pérez, and Matthieu Cord. Boosting few-shot visual learning with self-supervision. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8059–8068, 2019.

[56] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

[57] Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *International conference on algorithmic learning theory*, pages 63–77. Springer, 2005.

[58] Michael Gutmann and Aapo Hyvärinen. 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, 2010.

[59] Fusheng Hao, Fengxiang He, Jun Cheng, Lei Wang, Jianzhong Cao, and Dacheng Tao. Collect and select: Semantic alignment metric learning for few-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8460–8469, 2019.

[60] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Conference on Computer Vision and Pattern Recognition*, 2020.

[61] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition*, 2016.

[62] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. 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, 2019.

[63] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[64] Ruibing Hou, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen. Cross attention network for few-shot classification. *Advances in Neural Information Processing Systems*, 32, 2019.

[65] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR, 2019.

[66] Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. The platonic representation hypothesis. *arXiv preprint arXiv:2405.07987*, 2024.

[67] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, July 2021.

[68] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR, 2021.

[69] Amita Kamath, Jack Hessel, and Kai-Wei Chang. What's "up" with vision-language models? investigating their struggle with spatial reasoning. In *Empirical Methods in Natural Language Processing*, 2023.

[70] Dahyun Kang, Heeseung Kwon, Juhong Min, and Minsu Cho. Relational embedding for few-shot classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8822–8833, 2021.

[71] Zaid Khan and Yun Fu. Contrastive alignment of vision to language through parameter-efficient transfer learning. In *The Eleventh International Conference on Learning Representations*, 2023.

[72] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *International Conference on Learning Representations*, 2014.

[73] Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, et al. Openassistant conversations–democratizing large language model alignment. *arXiv preprint arXiv:2304.07327*, 2023.

[74] Simon Kornblith, Ting Chen, Honglak Lee, and Mohammad Norouzi. Why do better loss functions lead to less transferable features? *Advances in Neural Information Processing Systems*, 34:28648–28662, 2021.

[75] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pages 3519–3529. PMLR, 2019.

[76] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013.

[77] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[78] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[79] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE transactions on pattern analysis and machine intelligence*, 36(3):453–465, 2013.

[80] Patrick Langechuan Liu. Illustrated difference between mlp and transformers for tensor reshaping, 2021. Towards Data Science Blog Post. https://towardsdatascience.com/illustrated-difference-between-mlp-and-transformers-for-tensor-reshaping-52569edaf89 (accessed 2 April 2025).

[81] Claudia Leacock and Martin Chodorow. Combining local context and wordnet similarity for word sense identification. *WordNet: An electronic lexical database*, 49(2):265–283, 1998.

[82] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 10657–10665, 2019.

[83] Karel Lenc and Andrea Vedaldi. Understanding image representations by measuring their equivariance and equivalence. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 991–999, 2015.

[84] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics.

[85] Wenbin Li, Lei Wang, Jinglin Xu, Jing Huo, Yang Gao, and Jiebo Luo. Revisiting local descriptor based image-to-class measure for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7260–7268, 2019.

[86] Yi Li, Hualiang Wang, Yiqun Duan, and Xiaomeng Li. Clip surgery for better explainability with enhancement in open-vocabulary tasks. *arXiv preprint arXiv:2304.05653*, 2023.

[87] Yixuan Li, Jason Yosinski, Jeff Clune, Hod Lipson, and John Hopcroft. Convergent learning: Do different neural networks learn the same representations? *arXiv preprint arXiv:1511.07543*, 2015.

[88] Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-sgd: Learning to learn quickly for few-shot learning. *arXiv preprint arXiv:1707.09835*, 2017.

[89] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[90] Chen Liu, Yanwei Fu, Chengming Xu, Siqian Yang, Jilin Li, Chengjie Wang, and Li Zhang. Learning a few-shot embedding model with contrastive learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8635–8643, 2021.

[91] Shubao Liu, Yuan Xie, Wang Yuan, and Lizhuang Ma. Cross-modality graph neural network for few-shot learning. *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2021.

[92] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[93] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022.

[94] David G Lowe. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1150–1157. Ieee, 1999.

[95] Qiang Lyu and Weiqiang Wang. Compositional prototypical networks for few-shot classification. In *AAAI Conference on Artificial Intelligence*, 2023.

[96] Jiawei Ma, Hanchen Xie, Guangxing Han, Shih-Fu Chang, Aram Galstyan, and Wael Abd-Almageed. Partner-assisted learning for few-shot image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10573–10582, 2021.

[97] Wan-Duo Kurt Ma, JP Lewis, and W Bastiaan Kleijn. The hsic bottleneck: Deep learning without back-propagation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5085–5092, 2020.

[98] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

[99] Puneet Mangla, Nupur Kumari, Abhishek Sinha, Mayank Singh, Balaji Krishnamurthy, and Vineeth N Balasubramanian. Charting the right manifold: Manifold mixup for few-shot learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2218–2227, 2020.

[100] Mayug Maniparambil, Raiymbek Akshulakov, Yasser Abdelaziz Dahou Djilali, Mohamed El Amine Seddik, Sanath Narayan, Karttikeya Mangalam, and Noel E. O'Connor. Do vision and language encoders represent the world similarly? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14334–14343, June 2024.

[101] Mayug Maniparambil, Kevin McGuinness, and Noel O'Connor. BaseTransformers: Attention over base data-points for One Shot Learning. In *British Machine Vision Conference 2022, BMVC 2022*, 2022.

[102] Mayug Maniparambil, Chris Vorster, Derek Molloy, Noel Murphy, Kevin McGuinness, and Noel E. O'Connor. Enhancing CLIP with GPT-4: Harnessing visual descriptions as prompts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 262–271, October 2023.

[103] Harry McGurk and John MacDonald. Hearing lips and seeing voices. *Nature*, 264(5588):746–748, 1976.

[104] Sachit Menon and Carl Vondrick. Visual classification via description from large language models. *International Conference on Learning Representations*, 2023.

[105] Lotfi B Merabet and Alvaro Pascual-Leone. Neural reorganization following sensory loss: the opportunity of change. *Nature Reviews Neuroscience*, 11(1):44–52, 2010.

[106] Jack Merullo, Louis Castricato, Carsten Eickhoff, and Ellie Pavlick. Linearly mapping from image to text space. In *The Eleventh International Conference on Learning Representations*, 2023.

[107] Tomas Mikolov, Quoc V Le, and Ilya Sutskever. Exploiting similarities among languages for machine translation (2013). *arXiv preprint arXiv:1309.4168*, 2022.

[108] George A Miller. WordNet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[109] Ari Morcos, Maithra Raghu, and Samy Bengio. Insights on representational similarity in neural networks with canonical correlation. *Advances in Neural Information Processing Systems*, 31, 2018.

[110] Luca Moschella, Valentino Maiorca, Marco Fumero, Antonio Norelli, Francesco Locatello, and Emanuele Rodolà. Relative representations enable zero-shot latent space communication. In *The Eleventh International Conference on Learning Representations*, 2022.

[111] Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*, 2022.

[112] Jishnu Mukhoti, Tsung-Yu Lin, Omid Poursaeed, Rui Wang, Ashish Shah, Philip H.S. Torr, and Ser-Nam Lim. Open vocabulary semantic segmentation with patch aligned contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19413–19423, June 2023.

[113] Tsendsuren Munkhdalai and Hong Yu. Meta networks. In *International Conference on Machine Learning*, pages 2554–2563. PMLR, 2017.

[114] Muhammad Ferjad Naeem, Muhammad Gul Zain Ali Khan, Yongqin Xian, Muhammad Zeshan Afzal, Didier Stricker, Luc Van Gool, and Federico Tombari. I2MVFormer: Large language model generated multi-view document supervision for zero-shot image classification. In *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 15169–15179, 2023.

[115] Muhammad Ferjad Naeem, Yongqin Xian, Luc V Gool, and Federico Tombari. I2dformer: Learning image to document attention for zero-shot image classification. *Advances in Neural Information Processing Systems*, 35:12283–12294, 2022.

[116] Thao Nguyen, Samir Yitzhak Gadre, Gabriel Ilharco, Sewoong Oh, and Ludwig Schmidt. Improving multimodal datasets with image captioning. *Advances in Neural Information Processing Systems*, 36, 2024.

[117] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, 2008.

[118] Antonio Norelli, Marco Fumero, Valentino Maiorca, Luca Moschella, Emanuele Rodola, and Francesco Locatello. Asif: Coupled data turns unimodal models to multimodal without training. *arXiv preprint arXiv:2210.01738*, 2022.

[119] Itsaso Olasagasti, Sophie Bouton, and Anne-Lise Giraud. Prediction across sensory modalities: A neurocomputational model of the McGurk effect. *Cortex*, 68:61–75, 2015.

[120] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[121] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[122] Vicente Ordonez, Girish Kulkarni, and Tamara Berg. Im2text: Describing images using 1 million captioned photographs. *Advances in Neural Information Processing Systems*, 24, 2011.

[123] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. *Advances in Neural Information Processing Systems*, 31, 2018.

[124] Mei-Hong Pan, Hong-Yi Xin, Chun-Qiu Xia, and Hong-Bin Shen. Few-shot classification with task-adaptive semantic feature learning. *Pattern Recognition*, 141:109594, 2023.

[125] Eunbyung Park and Junier B Oliva. Meta-curvature. *Advances in Neural Information Processing Systems*, 32, 2019.

[126] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *Conference on Computer Vision and Pattern Recognition*, 2012.

[127] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[128] Zhimao Peng, Zechao Li, Junge Zhang, Yan Li, Guo-Jun Qi, and Jinhui Tang. Few-shot image recognition with knowledge transfer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 441–449, 2019.

[129] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649, 2015.

[130] Sarah Pratt, Ian Covert, Rosanne Liu, and Ali Farhadi. What does a platypus look like? generating customized prompts for zero-shot image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15691–15701, 2023.

[131] Maurice Ptito, Ron Kupers, Steve Lomber, and Pietro Pietrini. Sensory deprivation and brain plasticity. *Neural plasticity*, 2012:810370, 2012.

[132] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[133] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019. `https://openai.com/blog/language-models-are-unsupervised-multitask-learners`. Accessed: 2025-04-03.

[134] Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. *Advances in Neural Information Processing Systems*, 30, 2017.

[135] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *Advances in Neural Information Processing Systems*, 34:12116–12128, 2021.

[136] Aravind Rajeswaran, Chelsea Finn, Sham M Kakade, and Sergey Levine. Meta-learning with implicit gradients. *Advances in Neural Information Processing Systems*, 32, 2019.

[137] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *International Conference on Learning Representations*, 2017.

[138] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, 2019.

[139] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.

[140] Nils Reimers and Iryna Gurevych. Pretrained models — sentence transformers documentation, 2024. `https://www.sbert.net/docs/sentence_transformer/pretrained_models.html`. Accessed: 2024-09-24.

[141] Marcus Rohrbach, Michael Stark, and Bernt Schiele. Evaluating knowledge transfer and zero-shot learning in a large-scale setting. In *Conference on Computer Vision and Pattern Recognition*, pages 1641–1648. IEEE, 2011.

[142] Marcus Rohrbach, Michael Stark, György Szarvas, Iryna Gurevych, and Bernt Schiele. What helps where–and why? semantic relatedness for knowledge transfer. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 910–917. IEEE, 2010.

[143] Vanessa Rouach, Yuliana Pushevsky, Alla Mayboroda, Alina Osherov, and Michal Guindy. Sun-397 the osteosee system measurements, based on parametric electrical impedance tomography, correlate with dual x-ray absorptiometry results for the diagnosis of osteoporosis. *Journal of the Endocrine Society*, 4(Supplement_1):SUN–397, 2020.

[144] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

[145] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.

[146] Andrei A. Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. In *International Conference on Learning Representations*, 2019.

[147] Abhineet Saxena. Convolutional neural networks (cnns): An illustrated explanation. XRDS: Crossroads, The ACM Magazine for Students, 2016. https://blog.xrds.acm.org/2016/06/convolutional-neural-networks-cnns-illustrated-explanation/. Accessed: 2023-12-07.

[148] Edgar Schonfeld, Sayna Ebrahimi, Samarth Sinha, Trevor Darrell, and Zeynep Akata. Generalized zero-and few-shot learning via aligned variational autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8247–8255, 2019.

[149] Martin Schrimpf, Jonas Kubilius, Ha Hong, Najib J. Majaj, Rishi Rajalingham, Elias B. Issa, Kohitij Kar, Pouya Bashivan, Jonathan Prescott-Roy, Franziska Geiger, Kailyn Schmidt, Daniel L. K. Yamins, and James J. DiCarlo. Brain-score: Which artificial neural network for object recognition is most brain-like? *bioRxiv preprint*, 2018.

[150] Martin Schrimpf, Jonas Kubilius, Michael J Lee, N Apurva Ratan Murty, Robert Ajemian, and James J DiCarlo. Integrative benchmarking to advance neurally mechanistic models of human intelligence. *Neuron*, 2020.

[151] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[152] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022.

[153] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of CLIP-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021.

[154] Eli Schwartz, Leonid Karlinsky, Rogerio Feris, Raja Giryes, and Alex Bronstein. Baby steps towards few-shot learning with multiple semantics. *Pattern Recognition Letters*, 160:142–147, 2022.

[155] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, 2018.

[156] Xiahan Shi, Leonard Salewski, Martin Schiegg, Zeynep Akata, and Max Welling. Relational generalized few-shot learning. *British Machine Vision Conference*, 2020.

[157] Shinsuke Shimojo and Ladan Shams. Sensory modalities are not separate modalities: plasticity and interactions. *Current opinion in neurobiology*, 11(4):505–509, 2001.

[158] Reece Shuttleworth, Jacob Andreas, Antonio Torralba, and Pratyusha Sharma. Lora vs full fine-tuning: An illusion of equivalence. *arXiv preprint arXiv:2410.21228*, 2024.

[159] Paulo Ramiler Silva, Tiago Farias, Fernando Cascio, Levi dos Santos, Vinícius Peixoto, Eric Crespo, Carla Ayres, Marcos Ayres, Victor Marinho, Victor Hugo Bastos, et al. Neuroplasticity in visual impairments. *Neurology international*, 10(4):7326, 2018.

[160] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[161] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in Neural Information Processing Systems*, 30, 2017.

[162] Richard Socher, Milind Ganjoo, Hamsa Sridhar, Osbert Bastani, Christopher D Manning, and Andrew Y Ng. Zero-shot learning through cross-modal transfer. In *NeurIPS*, 2013.

[163] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[164] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural networks*, 32:323–332, 2012.

[165] Barry E Stein, Paul J Laurienti, Terrence R Stanford, and Mark T Wallace. Neural mechanisms for integrating information from multiple senses. In *2000 IEEE International Conference on Multimedia and Expo. ICME2000. Proceedings. Latest Advances in the Fast Changing World of Multimedia (Cat. No. 00TH8532)*, volume 1, pages 567–570. IEEE, 2000.

[166] Jong-Chyi Su, Subhransu Maji, and Bharath Hariharan. When does self-supervision improve few-shot learning? In *European conference on computer vision*, pages 645–666. Springer, 2020.

[167] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learn-

ing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1199–1208, 2018.

[168] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[169] Manuel Teichert, Jürgen Bolz, et al. How senses work together: cross-modal interactions between primary sensory cortices. *Neural plasticity*, 2018, 2018.

[170] Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *arXiv preprint arXiv:2406.16860*, 2024.

[171] Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. *arXiv preprint arXiv:2401.06209*, 2024.

[172] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021.

[173] Eleni Triantafillou, Richard Zemel, and Raquel Urtasun. Few-shot learning through an information retrieval lens. *Advances in neural information processing systems*, 30, 2017.

[174] Anton Tsitsulin, Marina Munkhoeva, Davide Mottin, Panagiotis Karras, Alex Bronstein, Ivan Oseledets, and Emmanuel Müller. The shape of data: Intrinsic distance for data distributions. *arXiv preprint arXiv:1905.11141*, 2019.

[175] Vishaal Udandarao, Ankush Gupta, and Samuel Albanie. Sus-x: Training-free name-only transfer of vision-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2725–2736, 2023.

[176] Vishaal Udandarao, Ameya Prabhu, Adhiraj Ghosh, Yash Sharma, Philip Torr, Adel Bibi, Samuel Albanie, and Matthias Bethge. No" zero-shot" without exponential data: Pretraining concept frequency determines multimodal model performance. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

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

IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 26700–26709, 2024.

[178] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

[179] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in Neural Information Processing Systems*, 29, 2016.

[180] Alexander Visheratin. Nllb-clip–train performant multilingual image retrieval model on a budget. *arXiv preprint arXiv:2309.01859*, 2023.

[181] Joshua T Vogelstein, John M Conroy, Vince Lyzinski, Louis J Podrazik, Steven G Kratzer, Eric T Harley, Donniell E Fishkind, R Jacob Vogelstein, and Carey E Priebe. Fast approximate quadratic programming for graph matching. *PLOS One*, 10(4):e0121002, 2015.

[182] Patrice Voss, Olivier Collignon, Maryse Lassonde, and Franco Lepore. Adaptation to sensory loss. *Wiley Interdisciplinary Reviews: Cognitive Science*, 1(3):308–328, 2010.

[183] Ivan Vulić, Sebastian Ruder, and Anders Søgaard. Are all good word vector spaces isomorphic? *arXiv preprint arXiv:2004.04070*, 2020.

[184] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. Caltech-UCSD birds-200-2011 (CUB-200-2011). *Technical Report CNS-TR-2011-001, California Institute of Technology*, 2011.

[185] Mark T Wallace, M Alex Meredith, and Barry E Stein. Integration of multiple sensory modalities in cat cortex. *Experimental Brain Research*, 91:484–488, 1992.

[186] Yan Wang, Wei-Lun Chao, Kilian Q Weinberger, and Laurens van der Maaten. Simpleshot: Revisiting nearest-neighbor classification for few-shot learning. *arXiv preprint arXiv:1911.04623*, 2019.

[187] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

[188] Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. Convnext v2: Co-designing and scaling convnets with masked autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16133–16142, 2023.

[189] John M Wu, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. Similarity analysis of contextual word representation models. *arXiv preprint arXiv:2005.01172*, 2020.

[190] Ziyang Wu, Yuwei Li, Lihua Guo, and Kui Jia. Parn: Position-aware relation networks for few-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6659–6667, 2019.

[191] Yongqin Xian, Zeynep Akata, Gaurav Sharma, Quynh Nguyen, Matthias Hein, and Bernt Schiele. Latent embeddings for zero-shot classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 69–77, 2016.

[192] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 41(9):2251–2265, 2018.

[193] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *Conference on Computer Vision and Pattern Recognition*, 2010.

[194] Chen Xing, Negar Rostamzadeh, Boris Oreshkin, and Pedro O O Pinheiro. Adaptive cross-modal few-shot learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[195] Chengming Xu, Yanwei Fu, Chen Liu, Chengjie Wang, Jilin Li, Feiyue Huang, Li Zhang, and Xiangyang Xue. Learning dynamic alignment via meta-filter for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5182–5191, 2021.

[196] Hu Xu, Saining Xie, Xiaoqing Tan, Po-Yao Huang, Russell Howes, Vasu Sharma, Shang-Wen Li, Gargi Ghosh, Luke Zettlemoyer, and Christoph Feichtenhofer. Demystifying CLIP data. In *The Twelfth International Conference on Learning Representations*, 2024.

[197] Jin Xu, Jean-Francois Ton, Hyunjik Kim, Adam Kosiorek, and Yee Whye Teh. Metafun: Meta-learning with iterative functional updates. In *International Conference on Machine Learning*, pages 10617–10627. PMLR, 2020.

[198] Fengyuan Yang, Ruiping Wang, and Xilin Chen. Sega: Semantic guided attention on visual prototype for few-shot learning. *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1586–1596, 2021.

[199] Shuo Yang, Lu Liu, and Min Xu. Free lunch for few-shot learning: Distribution calibration. In *International Conference on Learning Representations*, 2021.

[200] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Few-shot learning via embedding adaptation with set-to-set functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8808–8817, 2020.

[201] Sung Whan Yoon, Jun Seo, and Jaekyun Moon. Tapnet: Neural network augmented with task-adaptive projection for few-shot learning. In *International Conference on Machine Learning*, pages 7115–7123. PMLR, 2019.

[202] Jifan Yu, Xiaozhi Wang, Shangqing Tu, Shulin Cao, Daniel Zhang-Li, Xin Lv, Hao Peng, Zijun Yao, Xiaohan Zhang, Hanming Li, et al. Kola: Carefully benchmarking world knowledge of large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

[203] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? *arXiv preprint arXiv:2210.01936*, 2022.

[204] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? In *International Conference on Learning Representations*, 2023.

[205] Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *arXiv preprint arXiv:2106.10199*, 2021.

[206] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11975–11986, 2023.

[207] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18123–18133, 2022.

[208] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12203–12213, 2020.

[209] Manli Zhang, Jianhong Zhang, Zhiwu Lu, Tao Xiang, Mingyu Ding, and Songfang Huang. Iept: Instance-level and episode-level pretext tasks for few-shot learning. In *International Conference on Learning Representations*, 2020.

[210] Renrui Zhang, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free adaption of clip

for few-shot classification. In *European Conference on Computer Vision*, pages 493–510. Springer, 2022.

[211] Xueting Zhang, Debin Meng, Henry Gouk, and Timothy M Hospedales. Shallow bayesian meta learning for real-world few-shot recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 651–660, 2021.

[212] Heng Zhao, Kim-Hui Yap, and Alex Chichung Kot. Fusion learning using semantics and graph convolutional network for visual food recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1711–1720, 2021.

[213] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In *European Conference on Computer Vision*, pages 696–712. Springer, 2022.

[214] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. Image BERT pre-training with online tokenizer. In *International Conference on Learning Representations*, 2022.

[215] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16816–16825, 2022.

[216] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022.

[217] Luisa Zintgraf, Kyriacos Shiarli, Vitaly Kurin, Katja Hofmann, and Shimon Whiteson. Fast context adaptation via meta-learning. In *International Conference on Machine Learning*, pages 7693–7702. PMLR, 2019.

[218] George Kingsley Zipf. *The psycho-biology of language: An introduction to dynamic philology*. Routledge, 2013.