Do Vision and Language Encoders Represent the World Similarly?

Raiymbek Akshulakov

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2024-105
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2024/EECS-2024-105.html

May 15, 2024

Do Vision and Language Encoders Represent the World Similarly?

by Raiymbek Akshulakov

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

Committee:

Jmalik

Professor Jitendra Malik
Research Advisor

May 14, 2024

(Date)

Professor Trevor Darrell
Second Reader

May 14, 2024

(Date)

Abstract

Do Vision and Language Encoders Represent the World Similarly?

by

Raiymbek Akshulakov

Master of Science in Computer Science

University of California, Berkeley

Aligned text-image encoders such as CLIP have become the de-facto model for vision-language tasks. Furthermore, modality-specific encoders achieve impressive performances in their respective domains. This raises a central question: does an alignment exist between uni-modal vision and language encoders since they fundamentally represent the same physical world? Analyzing the latent spaces structure of vision and language models on image-caption benchmarks using the Centered Kernel Alignment (CKA), we find that the representation spaces of unaligned and aligned encoders are semantically similar. In the absence of statistical similarity in aligned encoders like CLIP, we show that a possible matching of unaligned encoders exists without any training. We frame this as a seeded graph-matching problem exploiting the semantic similarity between graphs and propose two methods - a Fast Quadratic Assignment Problem optimization, and a novel localized CKA metric-based matching/retrieval. We demonstrate the effectiveness of this on several downstream tasks including cross-lingual, cross-domain caption matching and image classification. Code available at github.com/mayug/0-shot-llm-vision.

Contents

List of Figures

List of Tables

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my advisor Jitendra Malik for his generous support throughout my academic journey. I am also incredibly thankful to Karttikeya Mangalam for giving me the invaluable opportunity to conduct research under his mentorship. Working with you has significantly advanced my technical skills and deepened my understanding of our field, inspiring me to pursue my research with passion and dedication. The research presented in this report is part of a larger collaborative effort with Mayug Maniparambil, Yasser Dahou, Mohamed Seddik, Sanath Narayan, Karttikeya Mangalam, and Noel E. O'Connor. I would like to extend my sincere thanks to all of them for their valuable contributions and insights. I am also immensely grateful to my fellow students, Vincent Wang and Daniel Flaherty from Malik Group, who have provided support and made significant contributions to my research efforts.

I want to express my gratitude to my friends who have been there for me throughout my life. To my close friends, Alikhan, Saddam, Abubakir, Khamzat, Mohammed, Abdullah, Leonardo, Aishwar, Mahyar, William, Zhanibek, Nikita, Malika, Rishi, Yersultan and many others thank you for your encouragement, and laughter, and for always being there for me. Your friendship means the world to me. A special thank you goes to my roommate Danial and my psychologist Kamilla for keeping me sane during challenging times. Last, and most importantly, I would like to thank my family who have supported me throughout my life. Their love, encouragement, and belief in me have been the foundation of my success.

Chapter 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 [20, 40]. 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 [19, 36, 10]. 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 [31, 34, 35]. The work of [31] 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 [9] 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) [41, 12, 22], which is known for measuring representation similarity both within and between networks. As shown in Figure 4.1, we measure the CKA between a variety of unaligned vision and language encoders [16, 47, 28, 37, 8], on the image-caption pairs of the COCO [27] dataset and observe that some have comparable scores to that of aligned encoders like CLIP [40], 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

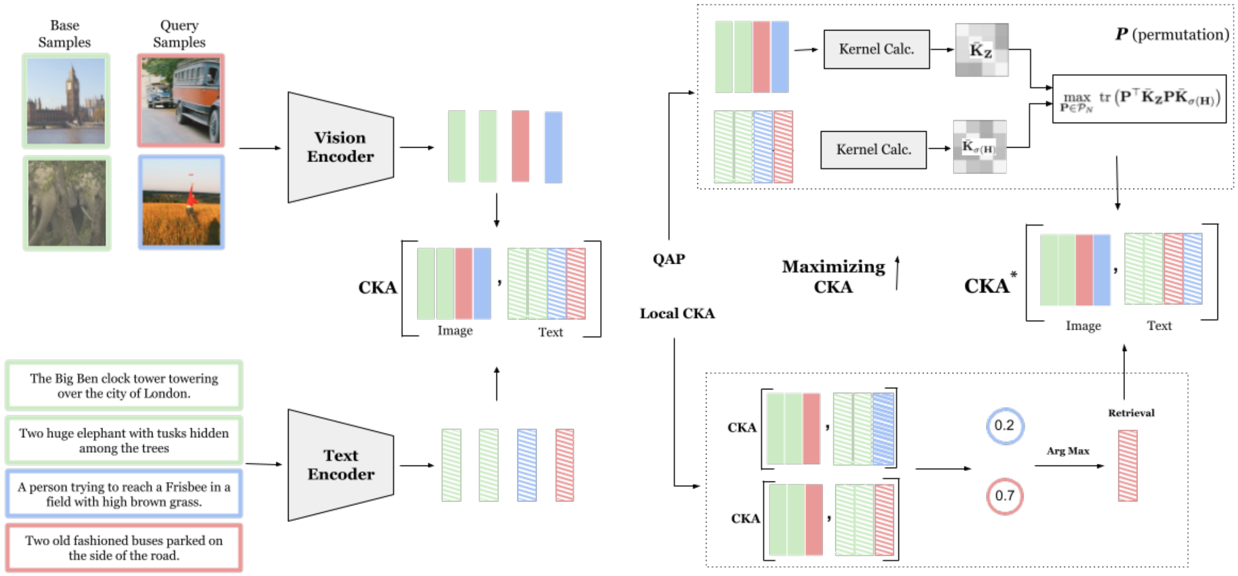

Base Samples
Query Samples
Vision Encoder
CKA
Image
Text
The Big Ben clock tower towering over the city of London.
Two huge elephant with tusks hidden among the trees
A person trying to reach a Frisbee in a field with high brown grass.
Two old fashioned buses parked on the side of the road.
Text Encoder
Kernel Calc.
$\bar{K}_Z$
Kernel Calc.
$\bar{K}_{\sigma(H)}$
P (permutation)
$\max_{P \in \mathcal{P}_N} \text{tr}\left(P^\top \bar{K}_Z P \bar{K}_{\sigma(H)}\right)$
QAP
Local CKA
Maximizing CKA
CKA*
Image
Text
CKA
CKA
0.2
0.7
Arg Max
Retrieval

Figure 1.1: **Methodology**. 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 $\boldsymbol{P}$. For retrieval, we append query embeddings to base embeddings and retrieve the best caption that maximizes the local CKA for a query image.

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. 1.1). 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 [34] on the COCO caption image retrieval.

- The method is benchmarked on COCO, NoCaps [2] cross-domain caption and image retrieval as well ImageNet-100 [15] 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.

Chapter 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 [33, 26, 22, 6] show that these networks exhibit representation similarity by learning similar latent spaces when trained on similar tasks and data [44, 5, 46, 11, 24, 32, 3]. Specifically, [22] 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) [42] being invariant to an invertible linear transformation that often leads to difficulty in measuring meaningful similarities between representations. [48] 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 [6] utilizes CKA along with Procrustes similarity for understanding the ability of variational autoencoders (VAEs) [21] 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 [34, 35] 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 [4, 14] analyze networks and empirically verify the “good networks learn similar representations” hypothesis by utilizing model stitching [24], which introduces trainable stitching layers to enable swapping parts of different networks. LiMBeR [31] can be seen as stitching the output of an image encoder to the input of a language model in the form of soft prompts [25]. 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 [34, 35] 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.

2.1 Preliminaries

Centered Kernel Alignment (CKA) has shown its relevance in understanding and comparing the information encoded by different layers of a neural network [22]. 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:

$$\mathrm{CKA}(\mathbf{K}, \mathbf{L}) = \frac{\mathrm{HSIC}(\mathbf{K}, \mathbf{L})}{\sqrt{\mathrm{HSIC}(\mathbf{K}, \mathbf{K})\,\mathrm{HSIC}(\mathbf{L}, \mathbf{L})}}, \tag{2.1}$$

where $\mathrm{HSIC}(\cdot, \cdot)$ is the Hilbert-Schmidt Independence Criterion [18, 30] defined as:

$$\mathrm{HSIC}(\mathbf{K}, \mathbf{L}) = \frac{1}{(N-1)^2}\,\mathrm{tr}\left(\mathbf{KCLC}\right), \tag{2.2}$$

with $\mathbf{C} = \mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^\top$ the centring matrix. We refer the reader to [22] for broader properties and studies of the CKA metric on neural network representations.

Chapter 3

Proposed Method

Consider a set of N image-caption pairs, $\mathcal{S} = \{(\boldsymbol{x}_i, \boldsymbol{c}_i)\}_{i=1}^{N}$, 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 3.1, the maximum CKA score is obtained on the ground-truth ordering of the representations $\text{CKA}_{\text{max}} = \text{CKA}(\mathbf{K}_{\mathbf{Z}}, \mathbf{K}_{\mathbf{H}})$, where $\mathbf{K}_{\mathbf{Z}}$ and $\mathbf{K}_{\mathbf{H}}$ are the kernels for the image and text representations, defined respectively as $\mathbf{K}_{\mathbf{Z}} = k(\mathbf{Z}^{\top}, \mathbf{Z})$ and $\mathbf{K}_{\mathbf{H}} = k(\mathbf{H}^{\top}, \mathbf{H})$. 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 lower CKA score as shown in Table 3.1. Therefore, our goal is to find a permutation σ^* that maximizes the CKA. Formally:

$$\sigma^* = \arg\max_{\sigma} \text{CKA}(\mathbf{K}_{\sigma(\mathbf{Z})}, \mathbf{K}_{\mathbf{H}}). \tag{3.1}$$

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. 1.1): 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

Table 3.1: **CKA reduces with shuffling.** We measure the CKA score between DINOv2 [37] and All-Roberta-large-v1 [28] on the 5k COCO [27] 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

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*.

3.1 QAP Matching

For some random permutation σ, the optimization problem in Equation 3.1 can be reformulated as a quadratic optimization problem [45] which reads as:

$$\max_{\mathbf{P}\in\mathcal{P}_N} \operatorname{tr}\left(\mathbf{P}^\top \bar{\mathbf{K}}_{\sigma(\mathbf{Z})}\mathbf{P}\bar{\mathbf{K}}_{\mathbf{H}}\right), \tag{3.2}$$

where $\mathcal{P}_N$ is the set of all permutation matrices of size N and $\bar{\mathbf{K}} = \mathrm{HSIC}(\mathbf{K},\mathbf{K})^{-\frac{1}{2}}\mathbf{KC}$ 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 3.2 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 3.2 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 [17]. 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:

$$\max_{\mathbf{P}\in\mathcal{P}_N} \operatorname{tr}\left((\mathbf{I}_M \oplus \mathbf{P})^\top \bar{\mathbf{K}}_{\mathbf{Z}}(\mathbf{I}_M \oplus \mathbf{P})\bar{\mathbf{K}}_{\mathbf{H}}\right), \tag{3.3}$$

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 $\mathbf{P}$.

3.2 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:

$$\text{localCKA}(\boldsymbol{z}^q, \boldsymbol{h}^q) = \text{CKA}(\mathbf{K}_{[\mathbf{Z}, \boldsymbol{z}^q]}, \mathbf{K}_{[\mathbf{H}, \boldsymbol{h}^q]}), \tag{3.4}$$

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 [23] for matching tasks.

3.3 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 5.1.

Chapter 4

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.

4.1 Vision and Language Encoders

The experimental setup covers vision encoders of different architectures, such as ViTs [16] and ConvNeXt [29], 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 [43] 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 [39, 7], 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 [28] demonstrates the best CKA/MS across all vision models, establishing it as the primary language encoder for subsequent tasks, unless specified otherwise.

4.2 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 [34]: 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 [40]: 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.

4.3 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 [27] 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 [2] 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.

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 [1] enhances COCO2014 with 1,000 human-annotated multi-lingual captions in ten languages for cross-lingual image retrieval and tagging, serving as a zero-shot

Table 4.1: **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 [40]. We use the Large version of all vision encoders. Table A.5 shows the reverse setting.

Method	**Vision Model**	**NoCaps** [2]		**COCO** [27]	
		Matching accuracy	Top-5 retrieval	Matching accuracy	Top-5 retrieval
Cosine Similarity*	CLIP [40]	99.5	99.6	97.1	96.1
Linear regression	CLIP-V [40]	29.3	44.7	42.7	59.1
	ConvNeXt [47]	19.0	28.5	31.3	46.1
	DINOv2 [37]	38.1	50.3	45.1	65.4
Relative representations [34]	CLIP-V [40]	61.3	37.6	61.6	41.3
	ConvNeXt [47]	25.5	17.8	38.6	34.1
	DINOv2 [37]	46.0	46.4	47.7	52.3
Ours: QAP	CLIP-V [40]	**67.3**	-	**72.3**	-
	ConvNeXt [47]	46.7	-	66.1	-
	DINOv2 [37]	57.7	-	66.0	-
Ours: Local CKA	CLIP-V [40]	65.1	60.5	71.9	69.9
	ConvNeXt [47]	43.7	44.4	64.8	65.5
	DINOv2 [37]	58.7	**61.8**	64.3	**70.5**

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.

4.4 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 6.1 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 [34], 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 4.1. 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.

ImageNet-100 Classification: In Table 4.2, 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.

Cross-lingual Caption Retrieval: The results of cross-lingual caption matching/retrieval are presented in Table 4.3 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

Table 4.2: **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 [34]	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

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 push the performance further by using language-specific sentence encoders and we report these results for a few languages in Sec 6.9 of supplementary. 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 that language is available.

4.5 Matching complexity

In Table 4.4, 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 $\mathcal{O}(n^3)$ [17], 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

Table 4.3: **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. See Table A.6 and Table A.7 in appendix for additional results.

Language		**Kernel CKA**		**Matching Accuracy**				**Retrieval @ 5**	
		CLIP	Ours	CLIP	Relative[34]	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**

Table 4.4: Run times for different methods

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

highly optimized algorithms for matrix multiplications in PyTorch [38]. Linear has a time complexity of $\mathcal{O}(nd)$, 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 [13] for linear sum assignment, which has the worst time complexity of $\mathcal{O}(n^3)$.

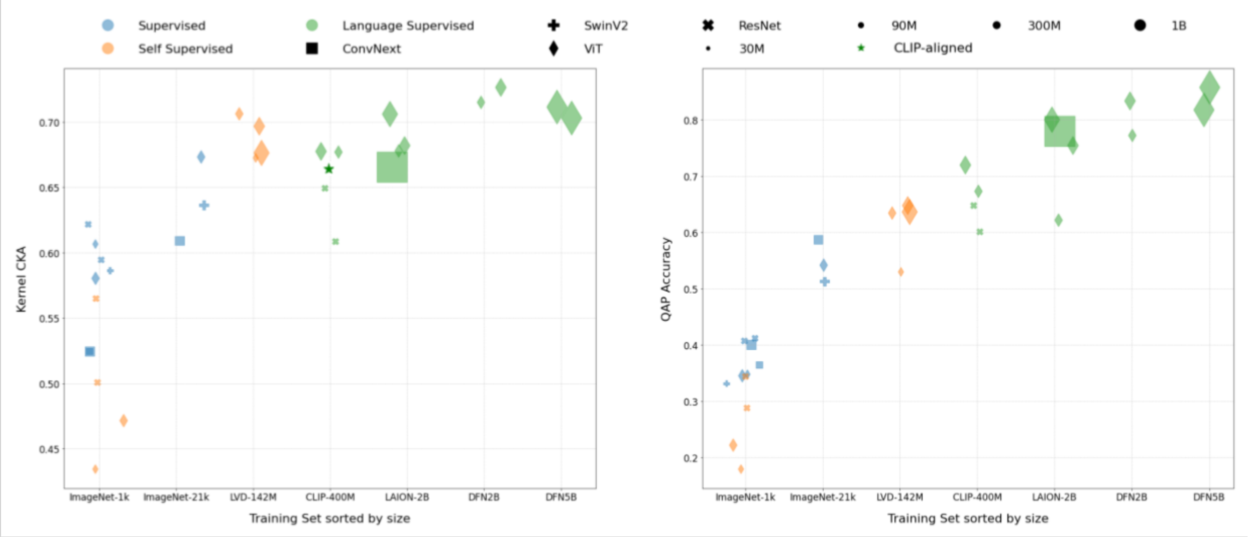


Figure 4.1: **Kernel CKA and QAP Matching accuracy are correlated with the training set size and quality of the training set.** Here the language encoder is kept constant to the best BERT-sentence encoder (i.e.All-Roberta-large-v1). There is a clear correlation between CKA and QAP Matching accuracy across all architectures, training paradigm and data regimes.

Chapter 5

Analysis and Conclusion

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 4.1 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 alignment 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.[31], 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.

5.1 Ablations

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 5.1, 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.

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.

5.2 Conclusion

In this work, we ask the question, *'Do vision encoders and language encoders represent the world similarly?'* and study this using CKA and a caption-matching task. We find that well-trained vision encoders on sufficiently large datasets exhibit surprisingly high semantic similarity with language encoders comparable to aligned encoders, irrespective of the training paradigm. 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. We also devise a local CKA-based metric to enable retrieval between unaligned vision and

Table 5.1: **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**

language encoders demonstrating a better performance than that of relative representations on cross-domain and cross-lingual caption matching/retrieval tasks, facilitating zero-shot latent space communication between unaligned encoders.

Bibliography

[1] Pranav Aggarwal and Ajinkya Kale. "Towards zero-shot Cross-lingual Image retrieval". In: *arXiv preprint arXiv:2012.05107* (2020).

[2] Harsh Agrawal et al. "Nocaps: Novel object captioning at scale". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 8948–8957.

[3] Richard Antonello et al. "Low-dimensional structure in the space of language representations is reflected in brain responses". In: *Advances in neural information processing systems* (2021).

[4] Yamini Bansal, Preetum Nakkiran, and Boaz Barak. "Revisiting model stitching to compare neural representations". In: *Advances in neural information processing systems* (2021).

[5] Serguei Barannikov et al. "Representation topology divergence: A method for comparing neural network representations". In: *arXiv preprint arXiv:2201.00058* (2021).

[6] Lisa Bonheme and Marek Grzes. "How do variational autoencoders learn? insights from representational similarity". In: *arXiv preprint arXiv:2205.08399* (2022).

[7] Tom Brown et al. "Language models are few-shot learners". In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.

[8] Mathilde Caron et al. "Emerging properties in self-supervised vision transformers". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 9650–9660.

[9] Soravit Changpinyo et al. "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*. 2021, pp. 3558–3568.

[10] Ting Chen et al. "A simple framework for contrastive learning of visual representations". In: *International conference on machine learning*. PMLR. 2020, pp. 1597–1607.

[11] Alexis Conneau et al. "Word translation without parallel data". In: *arXiv preprint arXiv:1710.04087* (2017).

[12] Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. "Algorithms for learning kernels based on centered alignment". In: *The Journal of Machine Learning Research* 13.1 (2012), pp. 795–828.

[13] David F Crouse. “On implementing 2D rectangular assignment algorithms”. In: *IEEE Transactions on Aerospace and Electronic Systems* 52.4 (2016), pp. 1679–1696.

[14] Adrián Csiszárik et al. “Similarity and matching of neural network representations”. In: *arXiv preprint arXiv:2110.14633* (2021).

[15] J. Deng et al. “Imagenet: A large-scale hierarchical image database”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2009.

[16] Alexey Dosovitskiy et al. “An image is worth 16x16 words: Transformers for image recognition at scale”. In: *arXiv preprint arXiv:2010.11929* (2020).

[17] Donniell E Fishkind et al. “Seeded graph matching”. In: *Pattern recognition* 87 (2019), pp. 203–215.

[18] Arthur Gretton et al. “Measuring statistical dependence with Hilbert-Schmidt norms”. In: *International conference on algorithmic learning theory*. Springer. 2005, pp. 63–77.

[19] 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*. JMLR Workshop and Conference Proceedings. 2010, pp. 297–304.

[20] Chao Jia et al. “Scaling up visual and vision-language representation learning with noisy text supervision”. In: *International conference on machine learning*. PMLR. 2021, pp. 4904–4916.

[21] Diederik P Kingma and Max Welling. “Auto-encoding variational bayes”. In: *ICLR* (2014).

[22] Simon Kornblith et al. “Similarity of neural network representations revisited”. In: *International conference on machine learning*. PMLR. 2019, pp. 3519–3529.

[23] Harold W Kuhn. “The Hungarian method for the assignment problem”. In: *Naval research logistics quarterly* 2.1-2 (1955), pp. 83–97.

[24] 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*. 2015, pp. 991–999.

[25] Brian Lester, Rami Al-Rfou, and Noah Constant. “The power of scale for parameter-efficient prompt tuning”. In: *arXiv preprint arXiv:2104.08691* (2021).

[26] Yixuan Li et al. “Convergent learning: Do different neural networks learn the same representations?” In: *arXiv preprint arXiv:1511.07543* (2015).

[27] Tsung-Yi Lin et al. “Microsoft coco: Common objects in context”. In: *ECCV*. 2014.

[28] Yinhan Liu et al. “Roberta: A robustly optimized bert pretraining approach”. In: *arXiv preprint arXiv:1907.11692* (2019).

[29] Zhuang Liu et al. “A convnet for the 2020s”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 11976–11986.

[30] 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*. Vol. 34. 04. 2020, pp. 5085–5092.

[31] Jack Merullo et al. "Linearly mapping from image to text space". In: *arXiv preprint arXiv:2209.15162* (2022).

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

[33] Ari Morcos, Maithra Raghu, and Samy Bengio. "Insights on representational similarity in neural networks with canonical correlation". In: *Advances in neural information processing systems* 31 (2018).

[34] Luca Moschella et al. "Relative representations enable zero-shot latent space communication". In: *The Eleventh International Conference on Learning Representations*. 2022.

[35] Antonio Norelli et al. "Asif: Coupled data turns unimodal models to multimodal without training". In: *arXiv preprint arXiv:2210.01738* (2022).

[36] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. "Representation learning with contrastive predictive coding". In: *arXiv preprint arXiv:1807.03748* (2018).

[37] Maxime Oquab et al. "Dinov2: Learning robust visual features without supervision". In: *arXiv preprint arXiv:2304.07193* (2023).

[38] Adam Paszke et al. "Pytorch: An imperative style, high-performance deep learning library". In: *Advances in neural information processing systems* 32 (2019).

[39] Alec Radford et al. "Language models are unsupervised multitask learners". In: *OpenAI blog* 1.8 (2019), p. 9.

[40] Alec Radford et al. "Learning transferable visual models from natural language supervision". In: *International conference on machine learning*. PMLR. 2021, pp. 8748–8763.

[41] Maithra Raghu et al. "Do vision transformers see like convolutional neural networks?" In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 12116–12128.

[42] Maithra Raghu et al. "Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability". In: *Advances in neural information processing systems* 30 (2017).

[43] 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, Nov. 2019. URL: `https://arxiv.org/abs/1908.10084`.

[44] Anton Tsitsulin et al. "The shape of data: Intrinsic distance for data distributions". In: *arXiv preprint arXiv:1905.11141* (2019).

[45] Joshua T Vogelstein et al. "Fast approximate quadratic programming for graph matching". In: *PLOS one* 10.4 (2015), e0121002.

[46] Ivan Vulić, Sebastian Ruder, and Anders Søgaard. "Are all good word vector spaces isomorphic?" In: *arXiv preprint arXiv:2004.04070* (2020).

[47] Sanghyun Woo et al. "Convnext v2: Co-designing and scaling convnets with masked autoencoders". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 2023, pp. 16133–16142.

[48] John M Wu et al. "Similarity analysis of contextual word representation models". In: *arXiv preprint arXiv:2005.01172* (2020).

Chapter 6

Appendix

6.1 Varying the Number of Samples

In Figure A.1, 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 A.2 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 an increase in N, but we still get 70% matching accuracy when $\frac{M}{N} = 1$.

6.2 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 is employed for text encoders. Table A.1 details the vision models, their training data, paradigms, and model types and sizes. Similarly, Table A.2 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 A.13. 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 the main paper, except for

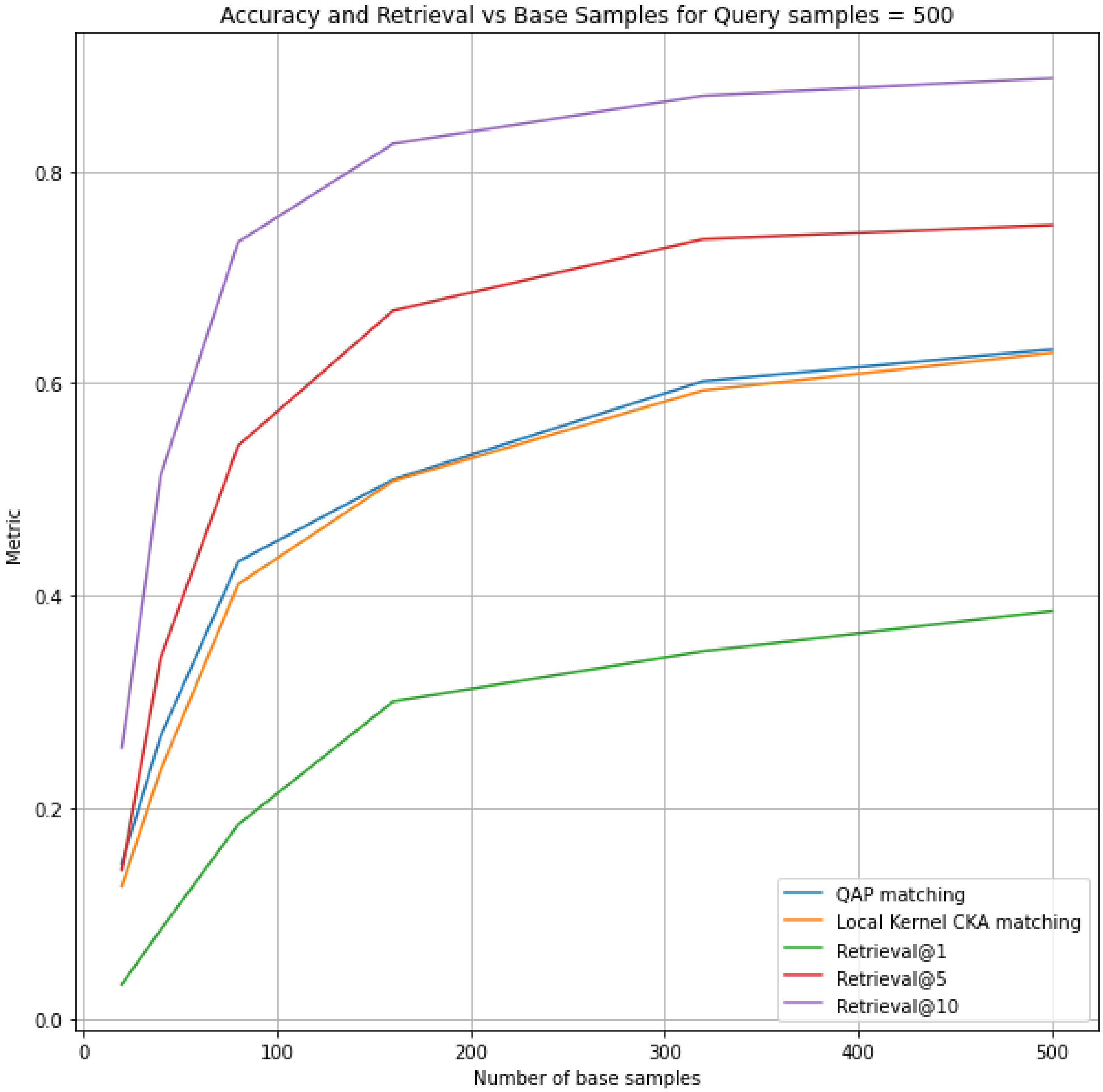


Figure A.1: **Accuracy and Retrieval Scores** of QAP Matching and Local CKA-based retrieval as the number of base samples is varied, keeping the number of query samples fixed at 500.

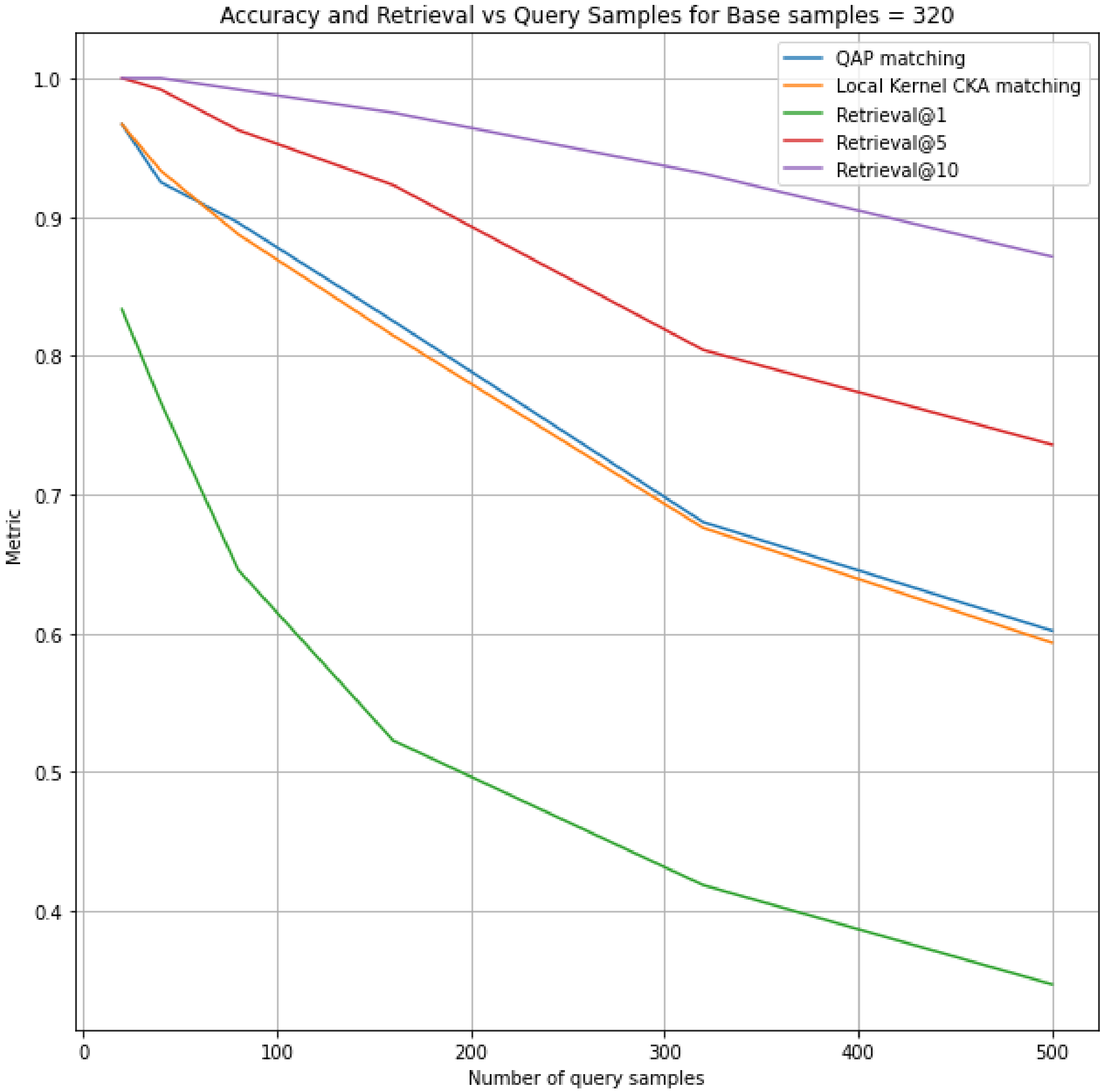


Figure A.2: **Accuracy and Retrieval Scores** of QAP Matching and Local CKA-based retrieval as the number of query samples is varied, keeping the number of base samples fixed at 320.

Table A.1: **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 A.3 and Table A.13.

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

cross-lingual experiments. For these, paraphrase-multilingual-mpnet-base-v2 emerges as the most effective text encoder.

Figure A.3 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 400 million image-caption pairs), demonstrate high CKA with language encoders of various sizes.

6.3 Layerwise CKA Analysis

Figure A.4, Table A.3, and Table A.4 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.

As shown in Table A.3, 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

Table A.2: **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 A.3 and Table A.13

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

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 A.4 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.

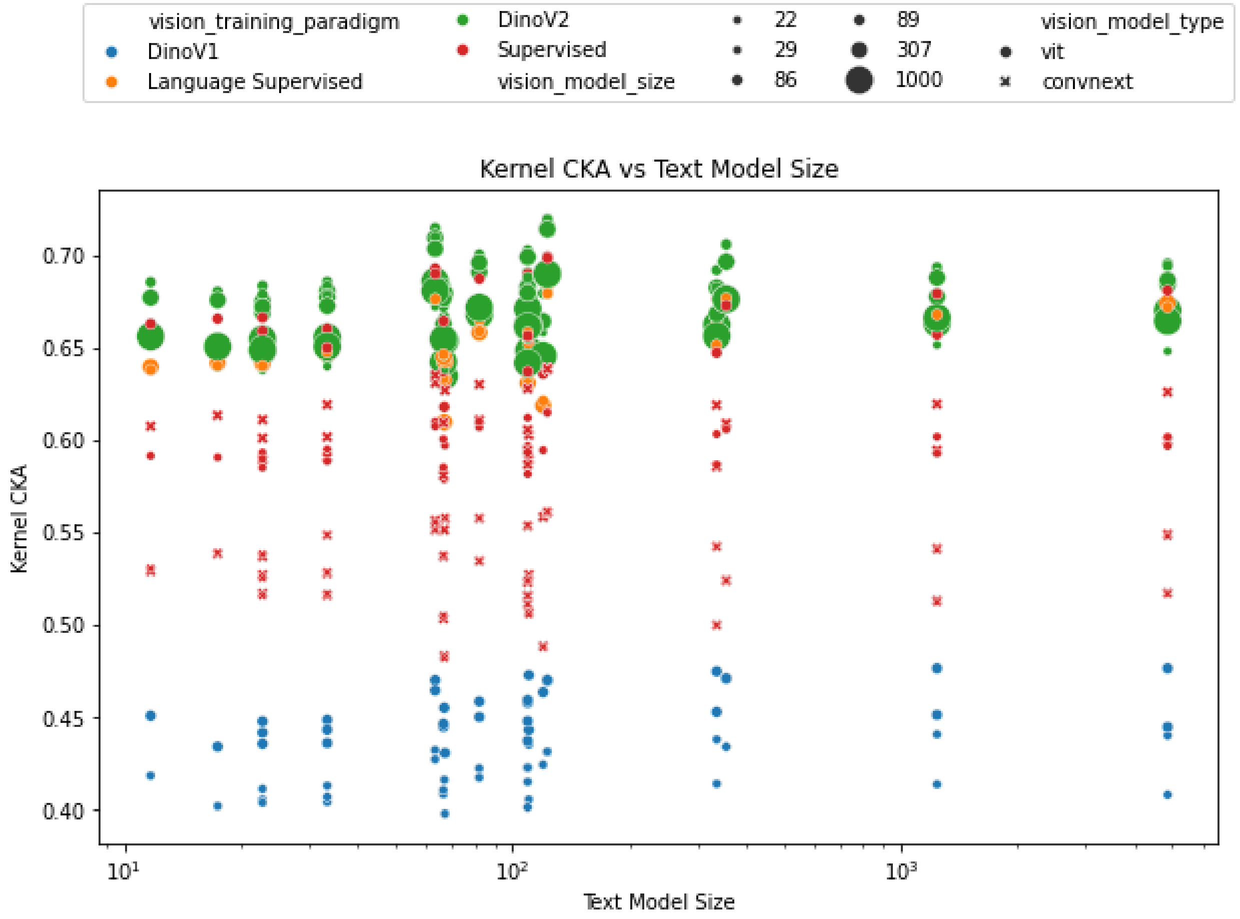
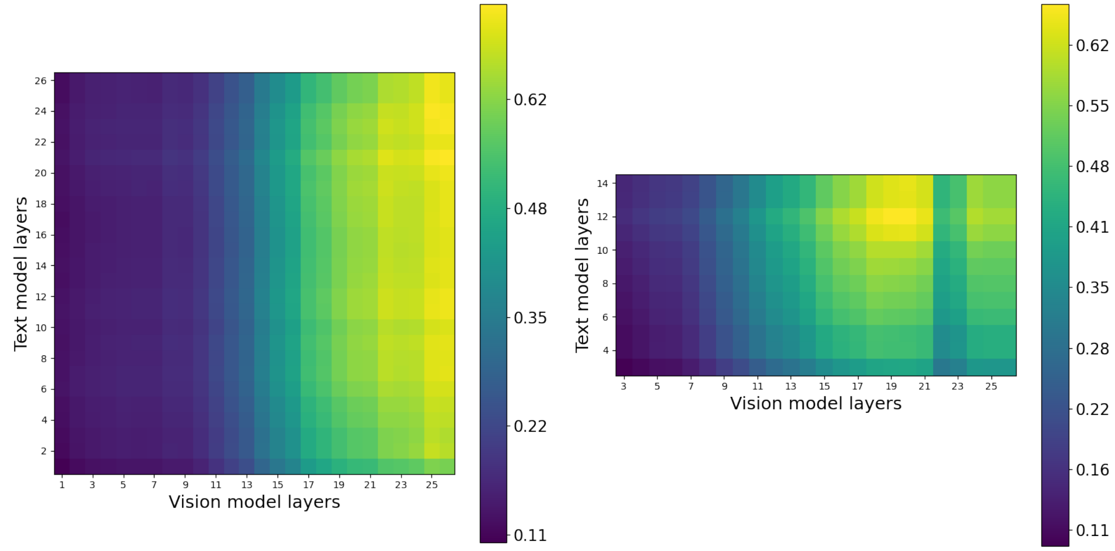


Figure A.3: **CKA *vs.* text model size** for vision encoders of different training paradigms, model types, and model sizes. We see that text model size is not the most important for high semantic similarity with vision models.

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

In this section, we provide derivations that show that the relative representations method [34] can be seen as a particular case of our proposed localCKA method. Denote the set of query and base representations samples respectively as $\mathbf{Q}_A = \left[\boldsymbol{q}_1^A, \dots, \boldsymbol{q}_N^A\right] \in \mathbb{R}^{d_A \times N}$ and $\mathbf{B}_A = \left[\boldsymbol{b}_1^A, \dots, \boldsymbol{b}_M^A\right] \in \mathbb{R}^{d_A \times M}$, where $A \in \{I, C\}$ for images and captions, the retrieval matrix for the relative representations (RR) method is therefore given by:

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



(a) (b)

Figure A.4: **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 w.r.t@let@tokenCKA.

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

$$\arg\max_j R_{ij}^{\mathrm{RR}} = \arg\max_j \, (\boldsymbol{q}_i^I)^\top \mathbf{B}_I \mathbf{B}_C^\top \boldsymbol{q}_j^C. \tag{6.1}$$

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:

$$\mathrm{localCKA}\left(\boldsymbol{q}_i^I, \boldsymbol{q}_j^C\right) = \mathrm{CKA}\left(\mathbf{K}_{[\mathbf{B}_I, \boldsymbol{q}_i^I]}, \mathbf{K}_{[\mathbf{B}_C, \boldsymbol{q}_j^C]}\right). \tag{6.2}$$

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., $\mathrm{CKA}(\mathbf{K}, \mathbf{L}) = \mathrm{tr}\,(\mathbf{KL})$. We first have, for $A \in \{I, C\}$:

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

Table A.3: **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 A.4: **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

Hence, we have:

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

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 A.5 shows the performance comparison between relative representations [34] 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

Table A.5: **Impact of adding noise to the embeddings**. Performance comparison, in terms of matching accuracy, between relative representations [34] 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 [34]	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)

also shown for reference (noise level of 0, *i.e*@let@token, $\sigma = 0$). 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*@let@token, for $\sigma = 0.2$, relative representations matching accuracy drops 6.5% from it 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.

6.5 Other text encoders

Evaluating on COCO with M=320 and N=500, Table A.6 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

Table A.6: 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

6.6 Simple projection

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 A.7 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.

6.7 Effect of unimodal tasks on alignment

Table A.8 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 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 A.7: 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 A.8: 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

6.8 Additional Retrieval Results

While the performance on the image retrieval task was reported in Table 2 of the main manuscript, here in Table A.9, 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,

Table A.9: **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** [2]		**COCO** [27]	
		Matching accuracy	Top-5 retrieval	Matching accuracy	Top-5 retrieval
Cosine Similarity*	CLIP [40]	99.5	99.6	97.1	98.5
Linear regression	CLIP-V [40]	63.6	70.1	72.6	83.9
	ConvNeXt [47]	22.8	38.9	43.8	65.7
	DINOv2 [37]	46.8	59.9	56.2	75.9
Relative representations [34]	CLIP-V [40]	61.3	3.0	61.6	2.9
	ConvNeXt [47]	25.5	2.7	38.6	12.9
	DINOv2 [37]	45.9	38.1	47.7	43.7
Ours: QAP	CLIP-V [40]	67.3	-	72.8	-
	ConvNeXt [47]	45.9	-	65.1	-
	DINOv2 [37]	58.5	-	65.9	-
Ours: Local CKA	CLIP-V [40]	65.1	65.9	71.9	80.5
	ConvNeXt [47]	44.8	33.0	63.8	74.3
	DINOv2 [37]	55.7	64.2	64.3	76.0

leading to a compressed semantic space for the captions. Consequently, caption embeddings become more closer to one another, making the retrieval a lot harder.

6.9 Additional Cross-Lingual Matching Results

For completeness, we report the results in Table A.10 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 notice 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 A.11 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 A.10: **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[34]	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**

Table A.11: **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

6.10 Qualitative results

In Table A.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.

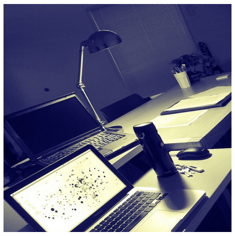
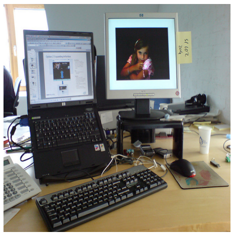
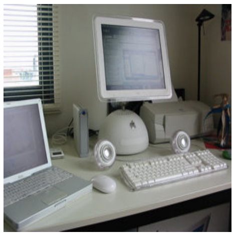
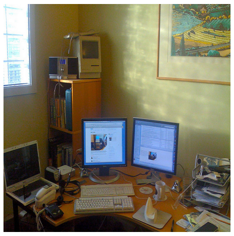
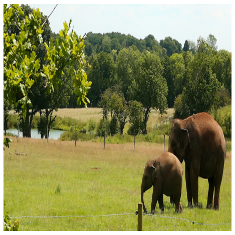
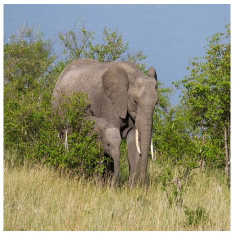
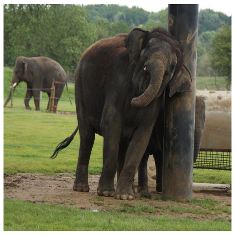
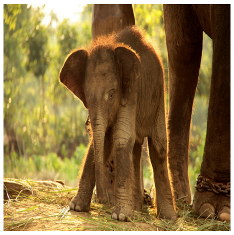
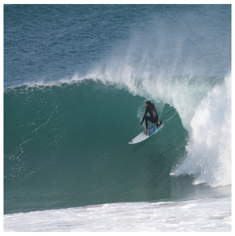
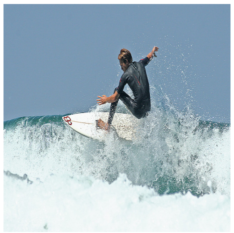
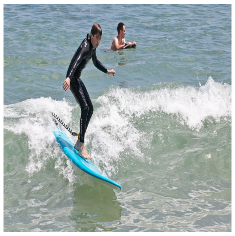
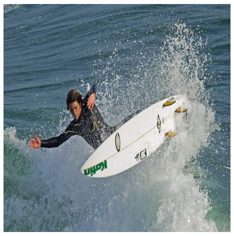
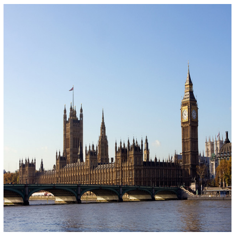
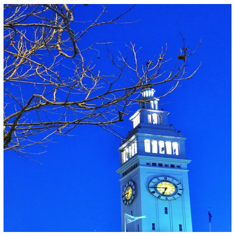
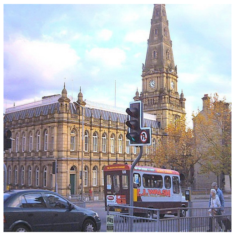
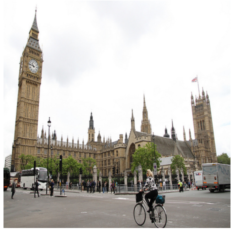
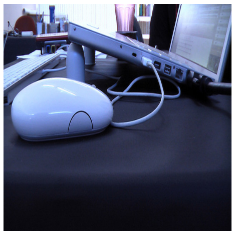
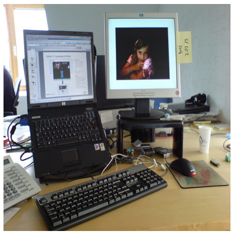
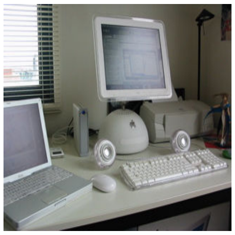
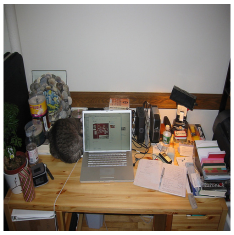
Original Image	Caption	Top-3 Retrieved Images		
	Two desktop computers sitting on top of a desk.			
	A mother and baby elephant walking in green grass in front of a bond.			
	a man is riding a surfboard at the beach			
	The Big Ben clock tower towering over the city of London.			
	A computer mouse is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

Table A.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 A.13: **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