# *Contrastive Sentence Representation Learning: Retrieval, Reasoning and Perception*

XIAO, CHENGHAO

# Contrastive Sentence Representation Learning: Retrieval, Reasoning and Perception

**Chenghao Xiao**

A Thesis presented for the degree of
Doctor of Philosophy

Department of Computer Science
Durham University
United Kingdom
Oct 2025

# Abstract

Representation learning has been centered to many NLP and multimodal tasks. This thesis presents a humble but systematic exploration of a specific form of ideal representations, i.e., representations that enable similarity search in the Euclidean space.

Centered to the training of this form of representation models is a technique called contrastive learning, whose spirit is to push instances which ought having similar semantics closer in the representation space, while pulling ones irrelevant away. Representation models trained using this technique widely facilitate language-only and multimodal applications, such as search engines.

This thesis studies representation models trained by contrastive learning from three progressive perspectives:

It first visits the fundamentals of contrastive learning (**Chapter 4, Chapter 5**), focusing on the mechanism why it works, from theoretical properties such as isotropy, contextualization and learning dynamics. It also studies how these properties connect to more behaviors such as models' length generalization. Using these insights, an unsupervised contrastive learning approach is presented, reaching state-of-the-art performance on information retrieval benchmarks when it was released.

Going beyond traditional measurement and expectation of representation models' capabilities like retrieval, new challenges emerged especially under the context of collaborating with LLMs through paradigms like RAG. And we see the need to call into questions embedding models' generalization to OOD tasks (e.g., can it understand reasoning-level expressions?) and instruction-following capabilities. At that time, co-authors and I were the first to call for measuring reasoning capabilities of embedding models, proposing a visionary paradigm which we termed Reasoning as Retrieval (**Chapter 6**), which was lately widely adopted by the field.

From the benchmarking of reasoning and instruction-following capabilities, I saw the massive potential of generative models' representational power, as opposed to previous widely adoption of BERT-based and CLIP-based representation models. I grew the belief in March 2024 that "training representation models is aligning their

representational capabilities with their generative capabilities", and my research interest grows into proving that the upperbound of this alignment increases by grounding the models in more and more modalities (**Chapter 7, Chapter 8**) (e.g., going from LLMs to MLLMs). We proposed pixel sentence representation learning, a unified framework to model the semantics of visual texts, and trained Pixel Linguist, a powerful model that can understand visual representation of texts. We then built the largest image and multimodal benchmark, MIEB - Massive Image Embedding Benchmark, defined by 8 capability categories we see necessary to measure from multimodal representation models in the new era, incorporating 130 image/multimodal tasks in 38 languages. In the context of multimodality, we again saw concrete evidence between how generative models' representational capabilities can be activated through orders-of-magnitude less alignment training than the CLIP paradigm, revealing latent alignment built in generative pretraining which needs to be activated to be similarity-matchable. We also saw early signs of scaling law between models performance on generative benchmarks and their representation upperbound post-contrastive learning. Such topics are being actively studied at the time of this thesis.

And the above marks the very beginning of the journey to pursue the ultimate form of a generalizable omni-modal representation model.

# Declaration

The work in this thesis is based on research carried out at the Department of Computer Science, Durham University, United Kingdom. No part of this thesis has been submitted elsewhere for any other degree or qualification and it is all my own work unless referenced to the contrary in the text.

# Acknowledgements

This thesis would not have been possible without the support, love, and guidance of numerous individuals.

I thank my father and mother, who instilled in me good values and love. My mother taught me diligence and sparked my early interest in science. My father taught me resilience. I am also grateful for their unconditional support in every random decision I make in life.

I thank my supervisor, Noura Al Moubayed, for her endless guidance, support, and kindness. Noura afforded me the time and freedom to pursue my ideas and become the researcher I am today.

I thank Professor Chenghua Lin, with whom I was fortunate to work throughout my PhD journey. Chenghua taught me the rigor of a scholar.

I thank my lab mates and friends for the friendship and good times.

I thank my collaborators. I appreciate every pure discussion of ideas, especially in a time of fast-paced AI research.

I thank my mentors in the industry for their guidance, and for the opportunities to scale up and turn my ideas and intuition into reality.

Last but not least, I want to thank me for making this happen.

# Contents

# List of Figures

# List of Tables

# Dedication

To Mom.

# CHAPTER 1

# Preface

This thesis is the by-product of my exploration in the past four years on a topic that has not yet stopped fascinating me: Sentence Representation Learning, which I broadly refer to as *learning numerical representation of texts that are equal to or more than one word, such that their distance reflects their relative semantics.*

My PhD research journey interestingly coincides with an important paradigm shift in the research of Natural Language Processing (NLP), with the emergence of large language models happening in the middle. Surprisingly, sentence representation learning turns out to be a field of greater significance in facilitating research in the era of LLMs, instead of one eclipsed (e.g., the dominance of Retrieval-augmented Generation, or RAG, at the time of writing of this thesis).

My understanding of representation went through a severe shift, seen in Chapter 5, where we provided early evidence for the field that generative models have greater potential in attaining representational abilities. I conceptualized representation learning as an alignment to models' own generative capabilities.

And the upper bound of this capability alignment keeps getting surpassed by grounding in more and more modalities...

# CHAPTER 2

# Introduction

Learning numerical representation of texts is a long-standing topic in natural language processing (NLP). Representing texts in vectors allows their geometrical distance to be measured by varied similarity metrics, such that their relative semantics can be ranked numerically. Due to this reason, the most intuitive motivation of studying sentence representation is to find such "universal projectors", such that texts with closer semantics can be projected to have smaller distances than ones with different underlying meanings. This might sound easy, with sentences like *I love you* and *I like you* deemed supposed to have a smaller distance than either of them to the sentence *It's awful innit* [1].

However, the complexity of this problem scales up along the need to ensure the representations a model produces makes the rule hold for a larger exhaust combinations of words in the human vocabulary.

This thesis revolves around the core problem of learning such universal sentence and document-level representations, such that the rule (similarity of embeddings

[1]The very first sentence I heard in Durham, sheltering from the rain under the eaves in front of a bank with an older gentleman, with the strongest northern British accent you can ever imagine, which I luckily managed to decipher thanks to my familiarity with a similar expression: *ain't it*, growing up picking up from Hip Hop music from the other continent.

reflect their real-world semantic relationships) holds for situations beyond domain-specific applications, and for texts that require modeling nuances and conditions in language, such as reasoning-level expressions and instruction-following. In the last two chapters of the thesis, we also seek to extend such insights to multi-modality.

The overall aim of the thesis is to analyze and build general-purpose dense representations for text and multimodality. The thesis addresses this high-level goal through three aspects. First, we provide an analysis framework to analyze properties of representations, which is shown able to provide transferable insights for training representation models. Second, we utilize findings we attain from the analyses to propose training methods of state-of-the-art text-only and multimodal representation models. Last, we construct systematic evaluation benchmarks for comprehensive assessment of representation quality. Throughout the thesis, we aim to reveal two findings: 1) Representation models can retrieve, reason, and perceive, which forms the title of the thesis. 2) More implicitly, we want to prove at a high level, that training a representation model is aligning their representational capabilities with their generative capabilities.

To achieve our overall aim, we ask the following research questions:

**Research Question 1: Why does contrastive learning work to attain useful embeddings?**

As we will see, contrastive learning has become a go-to method to transform a pretrained model into an embedding model that is able to perform similarity search. But just why do the base models not work, the mechanism why contrastive learning makes them work, and what properties have changed in the model during contrastive learning, remain under-explored. In Chapter 4, we first present a systematic study to answer this question, leveraging important metrics including anisotropy, intra-sentence similarity and self similarity, which reveal useful findings about models' encoded uniform embedding space, contextualization and learning dynamics.

**Research Question 2: What vulnerability does contrastive learning have and how can we solve it?**

Although contrastive learning is powerful, we show that vulnerability remains in models trained with contrastive learning. In Chapter 5, we study an important

property for text retrieval models, length robustness, a property central for retrievers to retrieve documents of different lengths accurately. We show that properties we analyzed in Chapter 4 are important to understand why models are not robust to documents across different lengths, such as when models are only trained on short documents, their long document embedding space remains anisotropic. We also show that we can effectively train state-of-the-art embedding models solely through addressing the length vulnerability.

**Research Question 3: Can retriever models solve reasoning problems?**

In the era of LLMs, we are faced with new challenges: Are retrievers good enough to work for paradigms which involve collaboration with LLMs, such as retrieval-augmented generation (RAG)? For example, how well can retrievers reason? If retrievers are not good at reasoning, they can then only play basic roles in RAG, with the reasoning burden mostly posed on LLMs (e.g., what to retrieve, how to decompose the queries).

In Chapter 6, we provide an upperbound assessment to answer this question. Specifically, we assess whether the retrievers can solve reasoning problems themselves. Findings in this chapter shed early light on the role of contrastive learning in the era of embedding models trained from generative backbones: Contrastive learning facilitates the alignment of a model's representational capability with their generative capability.

**Research Question 4: Can tokenization-free/vision encoders attain strong sentence/document embeddings?**

In parallel to the exploration of text-only representation models and methods, we saw a few disadvantages of text-only models. For example, they are not designed to be robust to typos due to the tokenization mechanism. With a one-character typo injected, what looks similar to humans and vision models and become a different set of tokens and thus confusing to text models. At a high level, text models are not able to naturally represent information encoded in complex figures and layouts. Also, they are not able to leverage the shape information of the characters to attain cross-lingual transferability like humans (e.g., simplified Chinese readers have high transferbility to zero-shot understand traditional Chinese characters without training,

which is not the case for text models due to tokenizing into distinct tokens). In Chapter 7, we present a vision-only framework to encode text semantics, which reveal the advantages of vision models in terms of robustness to shape-based perturbations, and their natural advantages to transfer knowledge across languages.

**Research question 5: How to evaluate embeddings holistically?**

As we extend the exploration scope to multimodal contexts, we aim to develop a framework to holistically evaluate embedding quality. Traditional multimodal representation benchmarks assess embedding quality through fragmented tasks and protocols, such as linear probing, full fine-tuning, or off-the-shelf assessment of embeddings through protocols such as retrieval. In Chapter 8, we introduce the largest multimodal embedding benchmark to date, covering 8 capability categories and 130 multimodal tasks across 38 languages. Apart from traditional tasks like linear probing, retrieval, clustering and zero-shot classification, we introduce challenging tasks to reveal advanced capabilities of current multimodal embedding models, including Visual Semantic Textual Similarity (Visual STS), Compositionality Evaluation, Visual Document Retrieval, and Vision-centric QA.

**Main Contributions** The main contribution of this thesis is as follows:

1. We provide a systematic framework to analyze properties in representation models, providing analytical lens to understand why contrastive learning works and explain vulnerabilities of the trained models.

2. We introduce methods to train state-of-the-art representation models, including LA(SER)$^3$ and Pixel Linguist.

3. We introduce two comprehensive benchmarks, RAR-b and MIEB, respectively assessing embedding models' reasoning capabilities and multimodal representation quality.

4. Through all chapters, we collectively reveal a conceptual principle: Training representation models is aligning their representational capabilities with their generative capabilities.

## 2.1 Basics

In this section, we introduce technical and theoretical basics of sentence representation learning.

### 2.1.1 Core Task and Settings

We start by introducing tasks used to assess embedding quality. There are two well-known tasks, Semantic Textual Similarity (8; 9; 10; 11; 12; 13) and Information Retrieval (e.g., the BEIR benchmark (14).) **Semantic Textual Similarity (STS)** assesses whether a model's similarity perception of two sentences aligns with human annotation. Each sentence pair is encoded by the model, yielding a similarity score. The full list of similarity scores is evaluated against human annotated scores, through Spearman Correlation. **Information Retrieval** assesses whether a model is able to retrieve targeted documents given a query. An information retrieval task typically consists of a query dataset, a document dataset (also known as corpus or database), and a table of relevance scores between each query and their relevant documents. In test time, the model encodes the queries and the documents. Similarity scores are computed between each query and all documents, deciding how top the groundtruth documents to a query ranked, among all documents. This "rank" is typically quantified using recall@k or nDCG@k. Recall@k is ranking-insensitive, only concerning whether the groundtruth documents are retrieved as top k. In contrast, nDCG@k is ranking-sensitive, assigning different scores to groundtruth documents depending on how close they are to top 1. Also note that there is a difference between the recall used in the NLP and the CV community. In CV, as long as the top k contains one of the groundtruth documents, that specific query gets a perfect recall score (a score of 1). In NLP, the recall is measured by the number of groundtruth documents in top k, divided by the number of all groundtruth documents to that specific query. We follow the NLP protocol of recall in Chapter 5-6 and the CV protocol in Chapter 8 according to the nature of the tasks.

Information retrieval approaches are generally categorized into two primary paradigms: sparse retrieval and dense retrieval. Sparse retrieval methods, such as

BM25, rely on lexical overlap and represent text as high-dimensional vectors where non-zero values correspond to specific keywords found in the vocabulary. In contrast, dense retrieval utilizes neural networks to project queries and documents into a shared, low-dimensional continuous vector space, enabling the system to capture semantic relationships and context beyond simple keyword matching. It is important to note that this thesis focuses exclusively on studying the representation aspect of these models—analyzing how information is encoded in the latent space—rather than on the retrieval mechanisms themselves.

### 2.1.2 Attaining Sentence Embeddings

With the introduction of Transformer-based language models (15) in 2017 and the pretraining of them starting to be widely-accepted (16; 17) in 2018, it has been de facto to use Transformer-based language models as the backbone since.

Given a Transformer model $f$ that maps a sequence of $n$ tokens $\mathbf{x} = [x_1, x_2, \ldots, x_n]$ to a sequence of $n \times h$ dimensional vectors $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n]$, where each $\mathbf{h}_i \in \mathbb{R}^h$ represents the hidden state corresponding to the $i$-th token:

$$\mathbf{H} = f(\mathbf{x}) \quad \text{where} \quad \mathbf{H} \in \mathbb{R}^{n \times h}$$

To obtain a sentence-level representation $\mathbf{s}$, a pooling operation is applied over the token vectors (i.e., token embeddings). Common pooling methods include mean pooling, max pooling, or taking the representation corresponding to a special token (e.g., [CLS] token in BERT). In sentence representation learning, mean pooling and [CLS] pooling are commonly seen in earler work, compared to max pooling used in applications in computer vision. With the recent emerging popularity of using decoder-based models for learning sentence representation, last-token pooling (using [EOS], the “end-of-sequence" token, of decoder language models) and weighted mean pooling are also frequently seen. Model properties under different pooling methods, including their anisotropy, self-similarity and intra-sentence similarity, are also study in our work, seen in Chapter 4.12.

**Mean pooling** computes the sentence representation by averaging the hidden

states across all tokens:

$$\mathbf{s} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{h}_i$$

Special tokens like [CLS] or [EOS] can either be included in the averaging, or not, and this can also be a hyperparamter to tune.

**Max pooling** computes the sentence representation by taking the element-wise maximum across all hidden states:

$$\mathbf{s}_j = \max_{i=1}^{n}\mathbf{h}_{i,j} \quad \text{for} \quad j = 1, 2, \ldots, h$$

$$\mathbf{s} = [\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_h]$$

where $\mathbf{s}_j$ represents the $j$-th element of the sentence representation vector $\mathbf{s}$.

**[CLS] Token Pooling**: In models like BERT, the hidden state corresponding to the special [CLS] token can be used as the sentence representation:

$$\mathbf{s} = \mathbf{h}_{[\text{CLS}]}$$

,

Intuitively, as [CLS] token is designed as an extra token to attend to the rest of the "regular" tokens in the sequence, one would hope that it can more or less represent the semantics of the sequence. Sadly, it is not true - at least without training [CLS] token to do so.

**Last token pooling** emerges in the era of decoder-based sentence representation learning, seen in a few state-of-the-art work. It takes the hidden state corresponding to the last token in the sequence as the sentence representation:

$$\mathbf{s} = \mathbf{h}_{[\text{eos}]}$$

Also intuitively, as decoder models apply uni-directional attentions, only the last token has the access to attend to all previous tokens, and it is the only token that is "qualified" to represent the sequence.

**Weighted average pooling** builds upon this intuition, and computes the

sentence representation by weighting each token's hidden state according to its position in the sequence. In the same vein as last token pooling, the key spirit is that later tokens should be given higher weights than previous tokens, as they have attended previous tokens. A form of the weight for the $i$-th token can be given by $\frac{i}{n}$, where $n$ is the total number of tokens. Based on this, the sentence representation is then:

$$\mathbf{s} = \sum_{i=1}^{n} \frac{i}{n} \mathbf{h}_i$$

**Summary** In each of these cases, the sentence representation $\mathbf{s} \in \mathbb{R}^h$ is a condensed form of the input sequence's information, and we optimize upon these condensed embeddings, to improve their representational abilities.

Despite the plethora of pooling methods, I tend to believe that all these operations will lead to an identical theoretical upperbound, and only serve to provide optimization advantages that happen to work best for different models. That is because: all these operations essentially define **weight priors** to tokens, which can also be learned through adapting attention weights throughout layers of a model - given enough optimization.

For instance, mean pooling gives an average prior to the tokens, saying that each token should be weighted the same. If this doesn't work the best for the model to fit the optimizaiton goal, e.g., if say the first token should receive higher weights, the model will learn to assign higher weights to the first token in early layers through attention mechanism, before they are simply averaged in the last layer. I provide a theoretical analysis between attention mechanism and pooling in Chapter 5.2.

### 2.1.3 Training

#### Why training?

The fundamental question we need to ask first is why do we need to train the models given the strong capabilities of the pretrained models. The answer lies in a well-documented observation: standard pooling methods applied to token embeddings from off-the-shelf pretrained Transformer-based language models often fail to produce

sentence embeddings that reliably capture real-world semantic relationships between sentences.

This phenomenon is first found in (18), where the [CLS]-pooled and mean-pooled embeddings of BERT even fall behind the representational abilities of the average of static embeddings from GloVe (19). Since then, research has been dedicated to train Transformer-based pretrained language models to better model sentence representations.

#### Desiderata

Given real-world pairwise data $i, j$ sampled from $p_{\text{data}}$, we aim to minimize between the model's similarity perception of $i, j$, and their ground-truth relative semantics $s_{ij}$.

$$\min_{f \in \mathcal{F}} \mathop{\mathbb{E}}_{\substack{\text{i.i.d.} \\ (i,j) \sim p_{\text{data}}}} \left[ \left( f(x_i)^\top f(x_j) - s_{ij} \right)^2 \right] + \lambda \sum_{i=1}^{N} \| f(x_i) \|_p^p , \tag{2.1}$$

where $x_i$ is a text, $f \in \mathcal{F}$ is a language model, and the second half of the equation serving as a regularization term to prevent representation collapse.

The goal has inspired earlier work on this problem, including the earliest version of SBERT (18). The limitation of this approach is that real-world similarities between texts are sparsely annotated, with only datasets like the STS series annotated with a few thousand examples (13), and optimizing the model to fit this objective generally suffers from overfitting. Therefore, it has gradually become established to see Equation 2.1 as the goal instead of the method, but rather optimize the models on orders of magnitudes larger datasets, with contrastive losses.

### 2.1.4 Contrastive Learning

Contrastive learning aims to push closer the representations of semantically similar instances, while pushing away the dissimilar ones. Among different variants of contrastive losses, InfoNCE (20) has emerged as a go-to solution in representation learning nowadays, including in sentence representation learning (3), image representation learning (21; 22), and image-text alignment (23).

Using a InfoNCE objective to fine-tune a PLM on datasets that consist of sentence/document pairs is defined as follows:

$$\ell_i = -\log \frac{e^{sim(e_i, e_i^+)/\tau}}{\sum_{j=1}^{N} e^{sim(e_i, e_j^+)/\tau}}, \tag{2.2}$$

where $e_i$ and $e_i^+$ denote embeddings of a sentence/document pair, whose cosine similarity is to be maximized, while all $e_j^+$ in a same training batch when $j \neq i$ is to be pushed further from $e_i$.

In the following section, we go through previous literature related to the contrastive learning technique, theoretical properties of representations and contrastive learning, and applications that can be facilitated by embeddings which motivate new directions of embedding research including ones done in this thesis.

## 2.2 Thesis Structure

The remaining chapters of this thesis is organized as follows, going from fundamental analysis to multi-modality. Chapter 3 first provides a comprehensive literature review, surveying the current landscape of dense text and multimodal representations, identifying theoretical gaps that motivating our subsequent explorations. Chapter 4 systematically investigates the mechanism of contrastive learning (RQ1), analyzing geometric properties like anisotropy to explain how pre-trained models transform into effective embedding models. Chapter 5 builds on this by identifying and resolving a critical vulnerability in contrastive learning regarding document length (RQ2), demonstrating that addressing specific embedding space flaws yields state-of-the-art retrieval performance. Under the era of LLMs, Chapter 6 transitions to the "Reason" component of the thesis (RQ3), providing an upper-bound assessment of whether retrievers can solve reasoning tasks and showing how contrastive learning aligns representational capabilities of models with their generative capabilities. Chapter 7 explores the "Perceive" aspect by questioning the necessity of tokenization (RQ4), proposing a vision-only framework that learns textual semantics using vision-only models. Finally, Chapter 8 unifies these themes by addressing how to evaluate embeddings holistically (RQ5), introducing a massive multimodal benchmark that rigorously

tests the "Retrieve, Reason, Perceive" capabilities across diverse multimodal tasks and languages.

The publications that are relevant to this thesis are as follows, and the full list of publications are provided in Chapter 10.

**(Chapter 4)** Xiao, C., Long, Y. and Al Moubayed, N., 2023, July. On Isotropy, Contextualization and Learning Dynamics of Contrastive-based Sentence Representation Learning. In Findings of the Association for Computational Linguistics: ACL 2023 (pp. 12266-12283).

**(Chapter 5)** Xiao, C., Li, Y., Hudson, G., Lin, C. and Al Moubayed, N., 2023. Length is a Curse and a Blessing for Document-level Semantics. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 1385–1396, Singapore. Association for Computational Linguistics.

**(Chapter 6)** Xiao, C., Hudson, G.T. and Al Moubayed, N., 2024. Rar-b: Reasoning as retrieval benchmark. arXiv preprint arXiv:2404.06347.

**(Chapter 7)** Xiao, C., Huang, Z., Chen, D., Hudson, G.T., Li, Y., Duan, H., Lin, C., Fu, J., Han, J. and Al Moubayed, N., 2024. Pixel sentence representation learning. arXiv preprint arXiv:2402.08183.

**(Chapter 8)** Xiao, C., Chung, I., Kerboua, I., Stirling, J., Zhang, X., Kardos, M., Solomatin, R., Al Moubayed, N., Enevoldsen, K., Muennighoff, N., 2025. MIEB: Massive Image Embedding Benchmark. In Proceedings of the IEEE/CVF international conference on computer vision

# CHAPTER 3

# Related Work

The pursuit of learning effective numerical representations for text is a foundational and rapidly evolving field in Natural Language Processing (NLP). This section situates the contributions of this thesis within the broader research landscape, tracing the progression from early contextual embeddings to the current frontiers of reasoning, multimodality, and novel representation paradigms. We structure this review into four key areas: (1) Sentence and document representation learning; (2) Dense retrieval and benchmarks; (3) Reasoning, RAG, and instruction-tuned embeddings; and (4) Tokenization-free and multimodal encoders. For each area, we synthesize the state-of-the-art and identify specific theoretical and practical gaps that this thesis addresses.

## 3.1 Sentence and Document Representation Learning

**From Static to Contextual Embeddings** Early approaches to sentence representation relied on averaging static, non-contextual word embeddings such as GloVe (19). The introduction of pre-trained Transformer-based language models (PLMs)

like BERT (16; 17) marked a paradigm shift. However, a critical "representation gap" was quickly identified: standard pooling methods (e.g., [cls] or mean pooling) applied to off-the-shelf PLMs often yielded sentence embeddings that performed poorly on semantic tasks, frequently lagging behind simple GloVe averages. (18; 19). This representation gap highlighted that the powerful contextual understanding at the token level did not automatically transfer to the sentence level.

Early and effective solutions emerged from supervised fine-tuning. A pioneering work, Sentence-BERT (SBERT) utilized Siamese network architectures to fine-tune pre-trained models on sentence-pair tasks like Natural Language Inference (NLI) (24; 25) and Semantic Textual Similarity (STS) (13). This approach successfully adapted the models to produce embeddings where spatial distance in the vector space corresponded to semantic similarity, establishing a powerful baseline that is still widely used today.

**The Rise of Contrastive Learning for Text Embedding models** More recently, the field has adopted contrastive learning (CL) as the dominant training paradigm. Utilizing objectives like InfoNCE, CL pulls semantically similar instances together while pushing dissimilar ones apart. SimCSE (3) represented a breakthrough in unsupervised learning by using standard dropout as a data augmentation technique, creating positive pairs from the same input processed twice. This has inspired a wave of research into unsupervised contrastive methods, including InfoCSE (2), DiffCSE (26), DeCLUTR (27) and so on, which explored different augmentation and auxiliary objective strategies.

**Isotropy and Post-Processing** A core motivation for these advancements is the "representation degeneration" problem , where embeddings from PLMs occupy a narrow, anisotropic cone in the vector space. This anisotropy limits the expressiveness of the representations. Several post-processing methods, such as BERT-flow (6) and BERT-whitening (5), attempted to transform embeddings into a more isotropic distribution (e.g., Gaussian) without further training. However, contrastive learning has been theoretically and empirically shown to promote uniformity (isotropy) more effectively during the training process itself.

**Research Gap** While the benefits of contrastive learning for isotropy are acknowledged, the precise mechanism and learning dynamics remain under-explored. Specifically, how does contrastive fine-tuning alter the internal geometry of the embedding space beyond global uniformity? Does it change how tokens within a sentence interact (intra-sentence similarity) or how the same token behaves across contexts (self-similarity)? Furthermore, there is a lack of deep analysis connecting these geometric shifts to model performance. This gap motivates Chapter 4, which provides a systematic analysis of isotropy, contextualization, and learning dynamics in contrastive sentence representation learning (28).

## 3.2 Dense Retrieval Methods and Benchmarks

Dense retrieval uses neural networks to project queries and documents into a shared embedding space, overcoming the lexical mismatch limitations of sparse retrieval methods like BM25.

**Modern Dense Retrivers** The effectiveness of dense retrievers relies heavily on negative sampling and training objectives. Architectures range from dual-encoders (bi-encoders) for efficiency to cross-encoders for higher accuracy but higher computational cost. Models like Contriever (29) demonstrated that strong retrieval performance could be achieved via unsupervised pre-training using creating positive pairs from independent spans of the same document.

**Benchmarking Information Retrieval** To rigorously evaluate generalization, the BEIR (Benchmarking IR) benchmark (14) was introduced. BEIR evaluates models in a zero-shot setting across diverse domains (e.g., bio-medical, finance, news), becoming the standard for assessing retrieval capabilities.

**Research Gap** Despite these advancements, the need to effectively address long documents has become increasingly relevant, and it remains a critical gap whether current state-of-the-art models can generalize effectively across different length ranges.

Furthermore, while standard contrastive learning has been powerful, it remains a question whether it can still introduce vulnerability.

As investigated in the work forming Chapter 5 of this thesis (30; 31), standard contrastive methods can introduce new, previously overlooked vulnerabilities, such as a model's sensitivity to document length. Specifically, we identify "length attacks," where models are fooled into perceiving higher similarity solely due to artificial elongation of content despite no new semantic information being added. This finding highlights that prior work has not provided a solution rooted in the training framework itself, motivating the development of more robust training frameworks. Consequently, Chapter 5 introduces LA(SER)$^3$, a framework that leverages document elongation as a semantic signal to achieve state-of-the-art unsupervised retrieval performance and robustness.

### 3.2.1 Expanding the Frontier: Reasoning, RAG, and Instructions-aware Embeddings

The advent of Large Language Models (LLMs) has transformed the requirements for embedding models, moving them from simple semantic matching to complex components within Retrieval-Augmented Generation (RAG) pipelines (32).

**Retrieval-Augmented Generation** Retrieval-Augmented Generation (RAG) In RAG, retrievers assist LLMs by fetching relevant context to mitigate hallucinations and provide up-to-date knowledge. However, in reasoning-intensive tasks (e.g., multi-hop QA), retrievers often fail to identify the necessary logical steps, placing the entire reasoning burden on the LLM.

**Instruction-aware Embeddings** This shift has given rise to a new class of instruction-aware embedding models. Models like Instructor (33) and TART (34) are trained with task-specific instructions (e.g., "Retrieve scientific papers for..."), allowing a single model to adapt to various domains and intents. Recent advances involve fine-tuning LLM backbones (e.g., E5-Mistral (35), GritLM (36)) to leverage the generative capabilities inherent in generative models for representation tasks.

**Research Gap** Despite the rise of RAG, evaluation of reasoning and retrieval remains orthogonal: reasoning is evaluated on generative models, and retrieval is evaluated on topical matching. There is a distinct absence of a systematic evaluation of representation models' reasoning capabilities. This motivates Chapter 6, where we introduce RAR-b (37), a benchmark that reframes reasoning tasks as retrieval problems to assess embedding models' reasoning capabilities using an upperbound setting: whether embedding models can solve the reasoning problems themselves by retrieving. This evaluation pushes the evaluation frontier beyond traditional STS and IR, and led us to systematically understand generative backbones' great representation potential, and our conceptual principle: Training representation is aligning models' representational capabilities with their generative capabilities.

## 3.3 Tokenization-free & Multimodal Models and Benchmarks

The final frontier of representation learning involves moving beyond text-only modalities toward unified multimodal understanding.

**Tokenization-free Methods** Standard language models rely on discrete subword tokenization, which is sensitive to noise (e.g., typos) and limits the application of computer vision augmentation techniques to text. PIXEL proposed rendering text as images to process language via vision encoders (ViT)(38). This paradigm has expanded with models like CLIPPO (39), which learns joint image-language representations from pixels, and autoregressive approaches concurrent to our work introduced in the thesis like PIXAR (40) and PTP (41) that investigate generation and screenshot understanding. While promising for robustness, vanilla pixel models lag significantly behind token-based models in semantic understanding.

**Vision-only Multimodal Models** Parallel to advancements in text, self-supervised vision-only models such as SimCLR (22), MoCo (21), and DINOv2 (42) have established robust visual representations through contrastive and reconstructive objectives.

CLIP (23) revolutionized multimodal learning by proving that feature transferability can be learned by aligning image and text encoders via massive contrastive training. This has led to numerous successors, including OpenCLIP (43), SigLIP (44), and ALIGN (45). More recently, Multimodal LLMs (MLLMs) like LLaVA have been adapted into embedding models, showing strong potential.

However, the evaluation of such powerful multimodal models has remained fragmented across task-specific protocols (1; 22; 42). CLIP models on zero-shot classification, vision models on linear probing , and retrieval models on specific datasets like MSCOCO. There is no unified, massive-scale benchmark that holistically evaluates image and image-text embeddings across diverse capabilities (retrieval, clustering, visual text understanding, etc.) and languages.

**Research Gap** Two major gaps exist here:

1) Weak Semantics in visual text Models: There is no framework that successfully transfers visual augmentation techniques (typos, shuffling) to learn strong sentence embeddings in tokenization-free models. This motivates Chapter 7, where we introduce Pixel Linguist (46).

2) Lack of Unified Multimodal Evaluation: The field lacks a comprehensive standard to compare vision-only, CLIP-style, and MLLM-based embedding models across a wide spectrum of tasks. This motivates Chapter 8, where we introduce MIEB (Massive Image Embedding Benchmark).

# CHAPTER 4

## On Properties of Contrastive Sentence Representation Learning

In this chapter, we reveal important empirical patterns we find about token and sentence-level embeddings that occur in the process of learning sentence representation with contrastive learning. The goal is to understand why contrastive learning works, aiming to provide fundamental insights that inspire the works in the following chapters, such as proposing enhancements to training methods (**Chapter 5, 7**). More specifically, we study three properties: **anisotropy, intra-sentence similarity, and self similarity**. These properties will reveal how sentence-level embeddings distribute in the semantic space (anisotropy), and how token-level embeddings interact with each other in the same sentence/document (through the lens of intra-sentence similarity and self-similarity). We will see how the findings in this chapter connects to interpreting the length vulnerability of retrievers (Chapter 5) and facilitate revealing cross-lingual transferability potential of visual text representation model (Chapter 7). In short, findings and analysis angles in this chapter provide transferable insights to understanding and enhancing representation models of different types. We now start by summarizing this chapter's content.

Incorporating contrastive learning objectives in sentence representation learning (SRL) has yielded significant improvements on many sentence-level NLP tasks.

However, it is not well understood why contrastive learning works for learning sentence-level semantics. In this chapter, we aim to help guide designs of sentence representation learning methods by taking a closer look at contrastive SRL through the lens of isotropy, contextualization and learning dynamics. We interpret its successes through the geometry of the representation shifts and show that contrastive learning brings isotropy, and drives high intra-sentence similarity: when in the same sentence, tokens converge to similar positions in the semantic space. We also find that what we formalize as "spurious contextualization" is mitigated for semantically meaningful tokens, while augmented for functional ones. We find that the embedding space is directed towards the origin during training, with more areas now better defined. We ablate these findings by observing the learning dynamics with different training temperatures, batch sizes and pooling methods.

## 4.1 Introduction

Since vanilla pre-trained language models do not perform well on sentence-level semantic tasks, Sentence Representation Learning (SRL) aims to fine-tune pre-trained models to capture semantic information (3; 6; 18). Recently, it has gradually become *de facto* to incorporate contrastive learning objectives in sentence representation learning (2; 3; 27; 47).

Representations of pre-trained contextualized language models (16; 17; 48) have long been identified not to be isotropic, i.e., they are not uniformly distributed in all directions but instead occupying a narrow cone in the semantic space (49). This property is also referred to as the representation degeneration problem (50), limiting the expressiveness of the learned models. The quantification of this characteristic is formalized, and approaches to mitigate this phenomenon are studied in previous research (50; 51; 52).

The concept of learning dynamics focuses on what happens during the continuous progression of fine-tuning pre-trained language models. This has drawn attention in the field (53; 54), with some showing that fine-tuning mitigates the anisotropy of embeddings (55), to different extent according to the downstream tasks. However, it

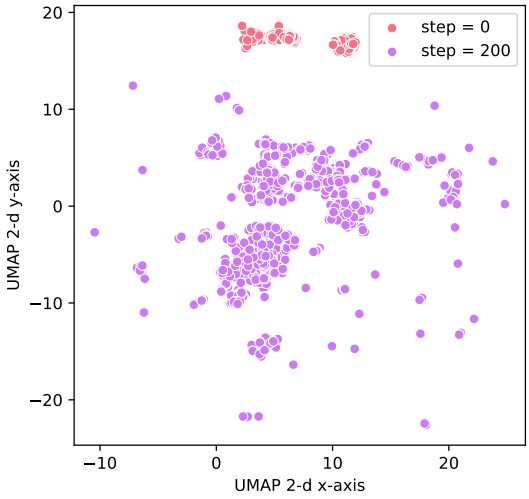


Figure 4.1: Expanded semantic space produced by contrastive learning (CL), visualized with UMAP. At the beginning of training, all embeddings occupied a narrow cone. After 200 steps of fine-tuning with a contrastive loss, they spread out to define a larger semantic space.

is argued that the performance gained in fine-tuning is not due to its enhancement of isotropy in the embedding space (55). Moreover, little research is conducted on isotropy of sentence embedding models, especially contrastive learning-based sentence representations.

Vanilla Transformer models are known to underperform on sentence-level semantic tasks even compared to static embedding models like Glove (18; 19), whether using the [cls] token or averaging word embeddings in the output layer. Since Reimers et al.(18) proposed SBERT, it has become the most popular Transformers-based framework in sentence representation tasks. The state-of-the-art is further improved by integrating contrastive learning objectives (2; 3; 47). The other line of works concern post-processing of embeddings in vanilla language models (5; 6; 56) to attain better sentence representations.

Learning dynamics in fine-tuning was previously investigated, revealing isotropy shifts in the process (3; 55), but few studies have systematically investigated relevant pattern shifts in sentence representation models, and none has drawn connections between these metrics and the performance gains on sentence-level semantic tasks.

While some implicitly studied this problem by experimenting on NLI datasets (53; 54; 55), we argue that a more extensive study on the geometry change during fine-tuning SOTA sentence embedding models with contrastive objectives is necessary.

To summarize to existing work and research gap: 1) Prior analysis work on anisotropy only focuses on base models. 2) Previous learning dynamics research focuses on standard fine-tuning such as classification. There is no work systematically analyzing token- and sentence-level geometric analysis of contrastive SRL and its link to performance.

In this work, we demystify the mechanism of why contrastive fine-tuning works for sentence representation learning.[1] Our main findings and contributions are as follows:

- Through measuring isotropy and contextualization-related metrics, we uncover a previously unknown pattern: contrative learning leads to extremely high intra-sentence similarity. Tokens converge to similar positions when given the signal that they appear in the same sentence.
- We find that functional tokens fall back to be the "entourage" of semantic tokens, and follow wherever they travel in the semantic space. We argue that the misalignment of the "spurious contextualization change" between semantic and functional tokens may explain how CL helps capturing semantics.
- We ablate all findings by analyzing learning dynamics through the lens of temperature, batch size, and pooling method, not only to validate that the findings are not artifacts to certain configurations, but also to interpret the best use of these hyperparamaters.

Our study offers fundamental insights into using contrastive objectives for sentence representation learning. With these, we aim to shed light on future designs of sentence representation learning methods.

[1] Our code is publicly available.

## 4.2 Isotropy and Contextualization Analysis of Contrastive-based Sentence Embedding models

### 4.2.1 Preliminary

Anisotropy of token embeddings produced by pre-trained language models has drawn attention in the field, and been validated both theoretically and empirically (49; 50; 52; 57).

For an anisotropic model, the embeddings it encodes have a high expected value of pair-wise cosine similarity: $\mathbb{E}_{u,v \in S} cos(u, v) >> 0$, where $u$ and $v$ are contextualized representations of tokens randomly sampled from corpus $S$.

A contrastive learning objective to fine-tune a PLM on datasets that consist of sentence/document pairs is defined as follows:

$$\ell_i = -\log \frac{e^{sim(e_i, e_i^+)/\tau}}{\sum_{j=1}^{N} e^{sim(e_i, e_j^+)/\tau}}, \tag{4.1}$$

where $e_i$ and $e_i^+$ denote embeddings of a sentence/document pair, whose cosine similarity is to be maximized, while all $e_j^+$ in a same training batch when $j \neq i$ is to be pushed further from $e_i$.

The central question posed in this chapter revolves around the mechanism involved in the contrastive learning process that diminishes anisotropy, leading to an isotropic model. If anisotropy is neutralized, we would observe a new mathematical expectation of cosine similarity, represented by $\mathbb{E}_{u,v \in S} cos(u, v) \approx 0$. However, the precise process and the underlying mechanism that facilitate this transition remain the key questions we aim to address.

Therefore, metrics such as self-similarity of same tokens in different contexts, and intra-sentence similarity of different tokens in the same context, are pertinent. More importantly, we could further trace the contextualization shift that brings mitigated anistropy to word type, i.e., are functional words and semantic words less/more contextualized after contrastive learning? We show that, this finding could potentially attribute to the performance gain on sentence-level semantic tasks brought by contrastive fine-tuning.

We summarize the main metrics used in Table 4.1 and explain in detail in the following subsection.

| | **Metric Computation** | **Interpretation** |
|---|---|---|
| Anisotropy | Expected cosine similarity of random sentences. | How collapsed the embedding space is. |
| Intra-sentence Similarity | Similarity between all tokens in the same sentence. | How tokens interact within the same sentence. |
| Self-Similarity | Similarity of the same token in different sentences. | How one token contextualizes in different contexts. |

Table 4.1: Metric overview table.

### 4.2.2 Metrics

We adopt the metrics defined in (49), who studied the extent to which word representations in pre-trained ELMo, BERT, and GPT-2 are contextualized, taking into consideration their anisotropy baselines. We reimplement the computation on self-similarity, intra-sentence similarity, and anisotropy baselines. We then break the similarity measures down into dimension level to inspect whether certain rogue dimensions (57) dominate these metrics and therefore making the similarity measures only artifacts of a small set of dimensions.

**Self Similarity:** Self similarity measures the similarity among different contextualized representations of a token across different contexts. Higher self-similarity indicates less contextualization. Given a token $x$, we denote the set of token embeddings of $x$ contextualized by different contexts in corpus $S$ as $S_{\vec{\mathcal{X}}}$. Self similarity is then defined as the empirical mean of pair-wise cosine similarity of contextualized embeddings of token $x$ in all these contexts:

$$selfsim(x) \triangleq \mathbb{E}_{u,v \in S_{\vec{\mathcal{X}}}}[\bar{cos}(u, v)] \tag{4.2}$$

**Intra-sentence Similarity:** By contrast, intra-sentence similarity measures the similarity across tokens in the same context.

Given a sentence $s$ with $n$ tokens $x_{i\in\{1,2,...,n\}}$, we first attain sentence representation $\vec{s}$ by mean-pooling, i.e., averaging all token embeddings $\vec{x_i}$. Intra-sentence similarity is then defined as the average cosine similarity between token representations $\vec{x_i}$ and the sentence representation $\vec{s}$.

$$\begin{aligned} \vec{s} &\triangleq \frac{1}{n}\sum_{x_i \in s} \vec{x_i} \\ intrasim(s) &\triangleq \frac{1}{n}\sum_{n} cos(\vec{x_i}, \vec{s}) \end{aligned} \tag{4.3}$$

Intra-sentence similarity provides a quantitative measure of the extent to which tokens in the same sentence are similar, allowing us later to derive insights on: whether token representations would converge in the semantic space only because they appear in a same sentence.

**Anisotropy Baselines:** While self and intra-sentence similarity are computed given the restrictions of respectively 1) same word in different contexts 2) different words in the same context, these values are not reflective of the general distribution across different words and different contexts.

In line with (49), we adjust the above two metrics by substracting the anisotropy baseline of a model from them, i.e., average cosine similarity between randomly sampled tokens from different contexts as defined in preliminary.

**Dimension-level Inspection of the Metrics** Due to the fact that cosine similarity is highly sensitive to outlier dimensions, we inspect whether the outcomes of the above measurements are only artifacts of these dimensions, i.e. rogue dimensions (57).

Formally, the cosine similarity of two embeddings is defined as: $cos(u, v) = \frac{u \cdot v}{\|u\|\|v\|}$, where $u$ and $v$ are two embeddings to measure against. Since the term $u \cdot v$ is just a sum of the element-wise dot product of the $i^{th}$ dimension of the embeddings, it is convenient to inspect the contribution each dimension makes to the global similarity: $cos(u, v) = \sum_{i=1}^{d} \frac{u_i v_i}{\|u\|\|v\|}$.

Given a set $S$ that consists of $n$ randomly sampled representations, the expected contribution of the $i^{th}$ dimension in a model to a similarity metric could be

approximated as:

$$cos_i = \mathbb{E}_{u,v \in S} \frac{u_i v_i}{\|u\| \, \|v\|}, \tag{4.4}$$

By breaking the global metrics down to dimension level, whether the output of a metric is a global property of all embeddings in the language model or is only dominated by a set of rogue dimensions $D$ could be inspected by whether $\sum_{i \in D} cos_i >> \frac{\|D\|}{d} \mathbb{E}_{u,v \in S} cos(u, v)$, with $d$ being the dimensionality of word embeddings.

Nonetheless, we could mathematically derive that, dominating dimensions dominate corpus-level similarity metric computations mostly because of their high average distances to the origin at the corresponding dimensions. However, if the values in these dimensions do not have high variation, then eliminating the top $\|D\|$ of these dimensions from the embeddings would not significantly bring semantic shifts to the original representations and therefore would not affect the corresponding relative similarity relationship between sentence pairs.

Therefore, we will also need to inspect whether there is a misalignment between the existence of the rogue dimensions, and their actual impact on informativity (57). Given a $f(t, k)$ that maps a token $t$ to its representation, with top $k$ rogue dimensions eliminated, we could compare the correlation between similarity measures yielded by the original representatations and those with top-k rogue dimensions removed. Formally, given:

$$cos_{original}(\mathcal{O}) = cos_{x,y \in \mathcal{O}}(f(x, 0), f(y, 0)) \tag{4.5}$$

$$cos_{post}(\mathcal{O}) = cos_{x,y \in \mathcal{O}}(f(x, k), f(y, k)), \tag{4.6}$$

we compute: $r = Corr[cos_{original}, cos_{post}]$, which is an indicator of the "authenticity" of the representations left without these rogue dimensions.

With the corresponding dimension-level inspections of the three metrics, we could take a step further to investigate whether fine-tuning a vanilla language model to sentence embedding tasks with the contrastive objective mitigates the dominance of rogue dimensions.

### 4.2.3 Models

We analyze two models that achieve SOTA performances on sentence embedding tasks and semantic search tasks, `all-mpnet-base-v2` [2] and `all-MiniLM-L6-v2`. [3] They have both been fine-tuned with a contrastive loss on 1B+ document pairs, with the goal of predicting the right match to a document $d_i$ given its ground-true match $d_i^+$ and the rest of the in-batch $d_j^+$ as natural negative examples. The prediction is conducted again reversely with $d_i^+$, $d_i$ and other in-batch $d_j$. The loss is averaged for these two components for every batch. The representation of each document $d$ is by default the mean-pooled embedding of each token.

We compare the results to their vanilla versions, *mpnet-base* (58) and *MiniLM*[4] (59) to get a closer look to the initial state of their corresponding pre-trained counterparts, and how the metrics change after fine-tuning on the goal of getting better sentence and document-level representations. In Section 4.2.5 and Section 4.3, we used these four off-the-shelf checkpoints described above (mpnet: before vs. after contrastive learning; MiniLM: before vs. after contrastive learning). And in Section 4.4.2, we re-train a model using the InfoNCE loss with mpnet-base to study the learning dynamics of the properties studied in Section 4.2.5 and Section 4.3.

### 4.2.4 Data

We use STS-B (13), which comprises a selection of datasets from the original SemEval datasets between 2012 and 2017. We attain the dataset through Hugging Face Datasets[5]. Notably, the models that we are looking at were not exposed to these datasets during their training. Therefore, the pattern to be found is not reflective of any overfitting bias to their training process.

We use the test set and only use sentence 1 of each sentence pair to prevent the potential doubling effect on self-similarity measure, i.e., providing tokens with one more sentence where they are in the similar contexts. Following the description, 1359

[2]https://huggingface.co/sentence-transformers/all-mpnet-base-v2
[3]https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2
[4]https://huggingface.co/nreimers/MiniLM-L6-H384-uncased. Notably, we use a 6-layer version.
[5]https://huggingface.co/datasets/stsb_multi_mt

sentences are selected as inputs.

### 4.2.5 Result

We show that after fine-tuning with contrastive loss, the anisotropy is almost eliminated in the output layer of both models, and is mitigated in the middle layers to different levels. This empirically validates the theoretical promise of uniformity brought by contrastive learning (3; 60) in the context of sentence representation learning (Figure 4.2).

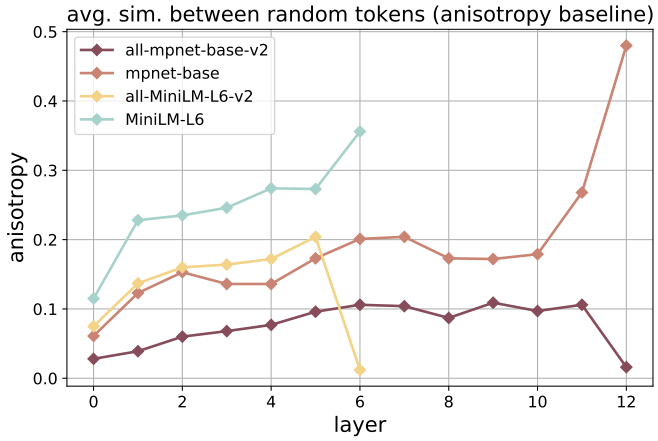


Figure 4.2: Anisotropy baseline of models

Complementing the enhanced isotropy, the average L2 norm of the randomly sampled token representations is also measured, showing a similar drastic shift in mostly the output layer of both models. Geometrically, the embeddings of tokens are pushed toward the origin in the output layer of a model, compressing the dense regions in the semantic space toward the origin, making the embedding space more defined with concrete examples of words (see also Figure 4.1), instead of leaving many poorly-defined areas (6). This property potentially contributes to models' performance gains on sentence embedding tasks.

Figure 4.4 and Figure 4.5 present respectively the self similarity and intra-sentence similarity of models adjusted (subtracted) by their anisotropy baselines (Unadjusted measures in Section 4.8).

As for the adjusted self similarity, we can see that the fine-tuned models generally show higher self similarities across contexts (meaning tokens are less contextualized

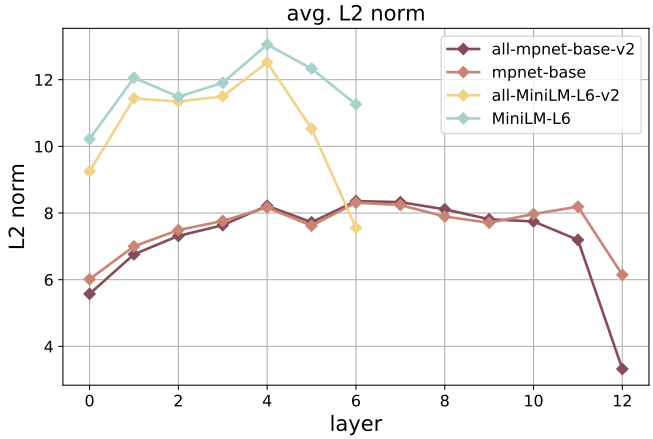


Figure 4.3: Avg. L2 norm of embeddings

after fine-tuning) in all layers, except for the output layer of the fine-tuned mpnet. However, in general there does not exist a large difference on this metric (See why in Section 4.3).

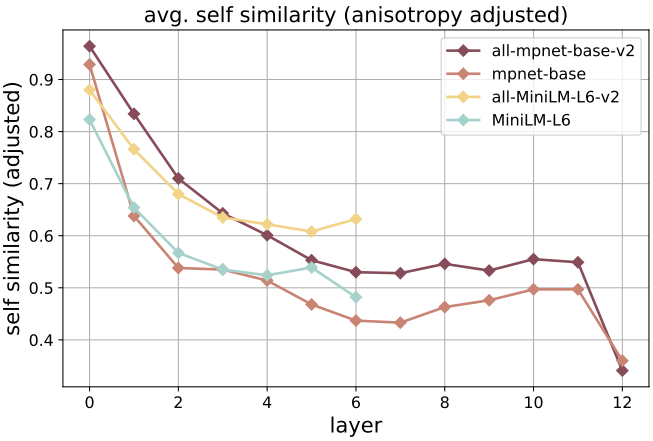


Figure 4.4: Adjusted self similarity of tokens: each self similarity is adjusted by the anisotropy of the corresponding model

We observe that intra-sentence similarity dramatically goes up in the output layer after contrastive fine-tuning. In the output layer of fine-tuned mpnet, the intra-sentence similarity reaches 0.834 (adjusted), meaning that tokens are 83.4% similar to one another if they appear in a same sentence. Since this pattern does not exist in the vanilla pre-trained models, the pattern is a unique behavior that accompanies the performance gain brought by contrastive learning. We argue that given contrastive examples and the goal of distinguishing between similar and non-similar in each

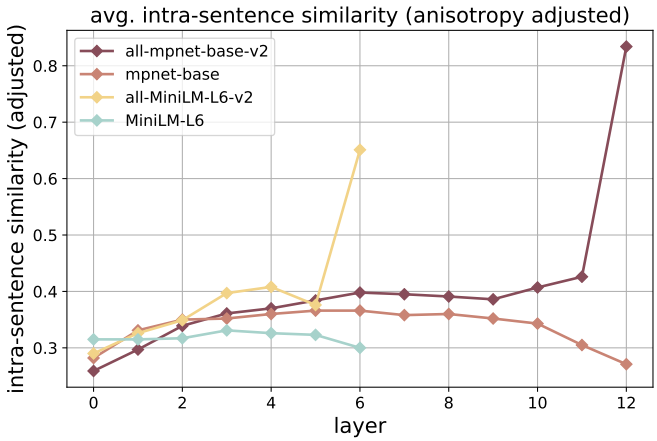


Figure 4.5: Adjusted intra-sentence similarity of tokens: each intra-sentence similarity is adjusted by the anisotropy of the corresponding model

batch, the model learns to provide more intense cross-attention among elements inside an input, and thus could better assign each example (sentence/document) to a unique position in the semantic space. With mean-pooling and positive pairs, the model learns to decide important tokens in a document $d_i$, in order to align with its paired document $d_i^+$, and other secondary tokens are likely to **imitate** the embeddings of these important tokens because they need to provide an average embedding together to match with their counterpart (In Section 4.12 we conduct an ablation study with other pooling methods). Further, with limited space in the now compressed space, inputs have now learned to converge to one another to squeeze to a point while keeping its semantic relationship to other examples. Therefore, we reason that, the unique behavior of this "trained intra-sentence similarity" is highly relevant to the models' enhanced performance on sentence-level semantic tasks.

To summarize the key observations: 1. After contrastive learning, anisotropy is largely removed from the models, which is shown by that embeddings from the models' last layer present an expeceted cosine similarity of near zero. 2. Contrastive learning brings extremely high intra-sentence similarity, making all tokens in the same sentence very similar with each other. 3. Average self similarities of tokens remain similar after contrastive learning, which as we further explore in Section 4.3, is due to the opposite self-similarity changes of semantic tokens and functional tokens being averaged.

| Model | Top 1 | Top 2 | Top 3 |
|---|---|---|---|
| $\text{mpnet}_{\text{vanilla}}$ | .548 | .723 | .741 |
| $\text{mpnet}_{\text{fine-tuned}}$ | .005 | .010 | .014 |
| $\text{minilm}_{\text{vanilla}}$ | .081 | .129 | .163 |
| $\text{minilm}_{\text{fine-tuned}}$ | .008 | .014 | .020 |
| | **10%** | **20 %** | **50%** |
| $\text{mpnet}_{\text{vanilla}}$ | 1 | 1 | 1 |
| $\text{mpnet}_{\text{fine-tuned}}$ | 28 | 64 | 209 |
| $\text{minilm}_{\text{vanilla}}$ | 2 | 5 | 31 |
| $\text{minilm}_{\text{fine-tuned}}$ | 19 | 40 | 121 |

Table 4.2: Dimension-wise inspection on vanilla and contrastive learning-based fine-tuned sentence representation models (last/output layer only). The upper part of the table presents the contribution percentage of the top 1 to 3 dominating dimensions. The lower part provides the number of top dimensions needed to account for {10,20,50}% of similarity metric computation.

Complementing the global properties found above, we present in Table 4.2 the dimension-level inspection on the measures. The analysis is conducted on self similarity. In line with previous work (57), there exists a significantly unequal contribution among dimensions. This inequality is most pronounced in the vanilla mpnet, with the top 1 dimension (out of the total 768) contributing to almost 55% of the similarity computation. After contrastive fine-tuning, this phenomena is largely removed, with dominating dimensions greatly "flattened" (3). For the fine-tuned mpnet, it now requires 209 (out of 768, 27.2%) dimensions to contribute to 50% of the metric computation, and for fine-tuned minilm, this number is 121 (out of 384, 31.5%).

In Section 4.11, we present the informativity analysis by removing top-k dominating dimensions, we see a reallocation of information after contrastive fine-tuning and a misalignment between dominance toward similarity computation and informativity.

## 4.3 Connecting to Frequency Bias

The imbalance of word frequency has long been identified to be relevant to the anisotropy of trained embeddings (50). This has been also empirically observed in pre-trained Transformers like BERT (6). (6) draw connection between frequency bias and the unideal performance of pre-trained language models on STS tasks, through

deriving individual words as connections of contexts, concluding that rare words fail to play the role of connecting context embeddings. (55) show that when fine-tuning pre-trained language models under the setting of Siamese architecture on STS-b datasets, the frequency bias is largely removed, with less significant frequency-based distribution of embeddings. However, it is also pointed out that these trained models are still highly anisotropic, which as we showed in Section 4.2.5, does not hold in the context of contrastive training, which, with sufficient data, has theoretical promise toward uniformity (3; 60).

Therefore, it is of interest to see the corresponding behaviors of frequency bias shifts in the context of contrastive learning, and more importantly, how this correlates with our surprising finding on intra-sentence similarity.

### 4.3.1 How Self Similarities Change for Frequent Words?

Since word frequency has produced many problematic biases for pre-trained Transformer models, we would like to know whether contrastive learning eases these patterns. Thus, how the self-similarity measurement manifests for frequent words after the models are fine-tuned with the contrastive objective? Are they more/less contextualized now?

**Validity of Measuring Self-Similarity Change** We first define Self-Similarity Change and prove that this measurement is not prone to stochasticity in the training process.

The top 400 frequent tokens are first extracted from the constructed STS-b subset. Then, we measure the avg. self-similarity before and after fine-tuning for each word, adjusted for their anisotropy baseline. Formally, we define Self-Similarity Change (SSC) of a token as:

$$ssc = (ss_f - ani_f) - (ss_v - ani_v), \tag{4.7}$$

where $ss_f$, $ss_v$, $ani_f$ and $ani_v$ stand for self-similarity and anisotropy baseline of fine-tuned and vanilla models respectively.

To validate that this measurement is not a product of stochasticity occurs in training but a common phenomenon that comes with contrastive learning, we compute

the Self-Similarity Change for every token using both *mpnet* (vanilla & fine-tuned) and *MiniLM* (vanilla & fine-tuned). If the statistics produced by both models show high correlation, then there exists a pattern that would affect how self-similarity changes for different tokens during contrastive fine-tuning. Otherwise, the changes are a product of randomness.

We iterate $n = 1$ to 400 to compute the Pearson correlation of SSCs of the top $n$ tokens produced by both *mpnet* and *MiniLM* and find the position where these statistics correlate the most, which is: $\arg\max_n(corr(ssc_{mpnet}[:n], ssc_{MiniLM}[:n]))$. Throughout the iteration, the top 204 frequent tokens give the highest Pearson correlation, which reaches a surprisingly high number of 0.857, validating the universal pattern for similarity shifts of frequent words. After inspection, we find that these are tokens that appear more than 9 times in the 1359 sentences. Notably, even the full set of 400 tokens gives a correlation of over 0.8, again proving the robustness of this pattern for frequent words (Refer to Section 4.13 for the full statistics of the validation).

### 4.3.2 Reaching to the connection

Table 4.3 provides a glimpse of the top 10 tokens (among the top 400 frequent tokens) that are now most more contextualized (with top negative self-similarity changes) and most less contextualized (with top positive self-similarity changes).

| | mpnet | | minilm | |
|---|---|---|---|---|
| | SS (↓) | SS (↑) | SS (↓) | SS (↑) |
| 0 | has | onion | [SEP] | hands |
| 1 | is | piano | . | fire |
| 2 | , | unfortunately | ; | run |
| 3 | ’ | cow | ? | house |
| 4 | are | chair | ) | japan |
| 5 | that | potato | the | hat |
| 6 | been | read | an | ukraine |
| 7 | while | dow | - | jumping |
| 8 | was | guitar | / | coffee |
| 9 | with | drums | a | points |

Table 4.3: Top Self-Similarity Changes

After contrastive fine-tuning, tokens that contribute more to the semantics (tokens

that have POS like nouns and adjectives) are now more reflective of their real-world limited connotations - tokens like “onion" and “piano" are not supposed to be that different in different contexts as they are in pre-trained models. We formalize this as **“Spurious Contextualization"**, and establish that **contrastive learning actually mitigates this phenomena for semantically meaningful tokens**. We speculate that these tokens are typically the ones that provide aligning signals in positive pairs and contrastive signals in negative pairs.

By contrast, however, the spurious contextualization of stopwords is even augmented after contrastive learning. “Has" is just supposed to be “has" - as our commonsense might argue - instead of having $n$ meanings in $n$ sentences. We speculate that, **stopwords fall back to be the “entourage" of a document after contrastive learning**, as they are likely the ones that do not reverse the semantics and thus do not provide contrastive signals in the training. Connecting this to our finding on high intra-sentence similarity, we observe that given a sentence/document-level input, certain semantic tokens drive the embeddings of all tokens to converge to a position, while functional tokens follow wherever they travel in the semantic space.

## 4.4 Ablation Analysis

In this section, we provide a derivation to interpret the role of temperature in CL, inspiring the searching method of its optimal range. We also show that contrastive frameworks are less sensitive to batch size at optimal temperature for SRL, unlike in visual representation learning.

### 4.4.1 Rethinking Temperature

Given a contrastive learning objective:

$$\ell_i = -\log \frac{e^{sim(e_i, e_i^+)/\tau}}{e^{sim(e_i, e_i^+)/\tau} + \sum_{j=1}^{N} \mathbb{1}_{\{j \neq i\}} e^{sim(e_i, e_j^+)/\tau}} \tag{4.8}$$

we first look at its denominator, where the goal is to minimize the similarity

between the anchor $e_i$ and negative pairs $e_j$ when $j \neq i$:

$$e^{sim(e_i,e_j^+)/\tau} \in (\frac{1}{e}^{\frac{1}{\tau}}, e^{\frac{1}{\tau}}) \tag{4.9}$$

Let $x$ be $e^{sim(e_i,e_i^+)}$ we get:

$$e^{sim(e_i,e_j^+)/\tau} = x^{1/\tau}, x \in (\frac{1}{e}, e) \tag{4.10}$$

If $\tau << 1$, as long as $x < 1$, $x^{1/\tau}$ shrinks exponentially. While when $x > 1$, $x^{1/\tau}$ explodes exponentially. Therefore, $x = 1$, or $sim(e_i, e_j^+) = 0$ when $i \neq j$ is an important threshold when negative pairs are to decide whether or not to further push away, and this "thrust", is exactly what temperature provides: In-batch negatives are not motivated to be too dissimilar under a lower temperature, since once the similarity reaches below 0, the exponent $1/\tau$ is already doing the job of making them exponentially vanishing in the denominator.

We analyze the upper bound and lower bound of $sim(e_i, e_j^+)$ under 0, giving us $sim(e_i, e_j^+) = 0$ and $sim(e_i, e_j^+) = -1$ for every $sim(e_i, e_j^+)$ in batch when $i \neq j$. For both cases we pair them with $sim(e_i, e_i^+) \to 1^-$ since positive pairs are drawn closer regardless. Therefore,

$$\begin{aligned} \ell_{upperbound}(\tau) &= -\log \frac{e^{sim(e_i,e_i^+)/\tau}}{e^{sim(e_i,e_i^+)/\tau} + \sum^{n-1} e^{0/\tau}} \\ &= -\log \frac{e^{sim(e_i,e_i^+)/\tau}}{e^{sim(e_i,e_i^+)/\tau} + (n-1)}, \end{aligned} \tag{4.11}$$

while given $sim(e_i, e_i^+) \to 1^-$,

$$\begin{aligned} \ell_{lowerbound}(\tau) &= -\log \frac{e^{sim(e_i,e_i^+)/\tau}}{e^{sim(e_i,e_i^+)/\tau} + \sum^{n-1} e^{-1/\tau}} \\ &= -\log \frac{e^{(sim(e_i,e_i^+)+1)/\tau}}{e^{(sim(e_i,e_i^+)+1)/\tau} + (n-1)} \\ &\approx -\log \frac{e^{2*sim(e_i,e_i^+)/\tau}}{e^{2*sim(e_i,e_i^+)/\tau} + (n-1)} \end{aligned} \tag{4.12}$$

Therefore, $\ell_{lowerbound}(2\tau) \approx \ell_{upperbound}(\tau)$.

We find that temperature affects making embeddings isotropic: to push in-batch negatives to the lower bound, the temperature needs to be twice as large than to push them to the upper bound. For example, if when temperature = 0.05, two sentences are pushed in training to have $-1$ cosine similarity, now given temperature = 0.025, the gradient is only around enough to push them to have 0 cosine similarity with each other.

The findings suggest that searching for the optimal value of this hyperparameter using a base of 10, as empirically shown in previous research (3), may not be the most efficient approach. Instead, we argue that a base of 2 would be more appropriate, and even to conduct finer-grained searching when a range of upper bound temperature that is twice the lower bound temperature is found to provide adequate performance.

Our analysis serves as a complementation to (61), who show that a lower temperature tends to punish hard-negative examples more (especially at the similarity range of $(0.5, 1)$), while a higher temperature tends to give all negative examples gradients to a same magnitude. This provides more theoretical justification to our approximation, since at the similarity range of $(-1, 0)$, all negative examples have gradients to the same magnitude (61) regardless. We suggest that this range plays a main role in making the entire semantic space isotropic.

### 4.4.2 Experiment Setup

We use a vanilla mpnet-base (58) as the base model, and train it on a concatenation of SNLI (24) and MNLI datasets (25). In accordance with our analysis, for the temperature $\tau$ subspace we deviate from the commonly adopted exponential selection with a base of 10 (e.g., (3)), but we analyze around the best value found empirically, with a base of 2, i.e., $\{0.025, 0.05, 0.1\}$. We provide the same analysis on $\{0.001, 0.01, 0.05, 0.1, 1\}$ in Section 4.9 for comparison. To better illustrate the effect of temperature, we only use entailment pairs as positive examples, under supervised training setting. We do not consider using contradiction as hard negatives to distract our analysis, nor unsupervised settings using data augmentation methods such as standard dropout. We use all instances of entailment pairs as training set, yielding a training set of 314k. We truncate all inputs with a maximum sequence length

of 64 tokens. All models are trained using a single NVIDIA A100 GPU. We train the models with different temperatures for a single epoch with a batch size of 64, yielding 4912 steps each, with 10% as warm-up. We save the models every 200 steps and use them to encode the subset of STS-B we have constructed.

### 4.4.3 Results

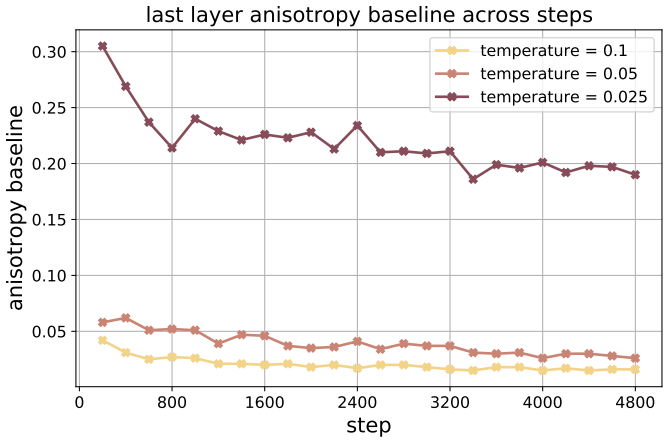


Figure 4.6: Anisotropy changes throughout training under different temperatures

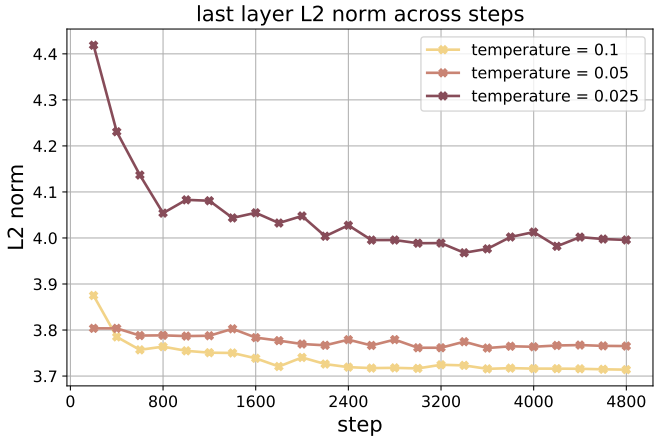


Figure 4.7: L2-norm under different temperatures

Firstly, we present the centered property we are measuring, anisotropy. Figure 4.6 shows the last-layer anisotropy change throughout steps. The trend is in line with our hypothesis about temperature being a "thurst". Knowing that the vanilla model starts from encoding embeddings to be stuck in a narrow angle, temperature serves

as the power to push them further through forcing negative pairs to be different. With a higher temperature, the cosine similarity between negative pairs has to be lower to reach a similar loss. Figure 4.7 further validates this through showing that higher temperatures compress the semantic space in general, pushing instances to the origin.

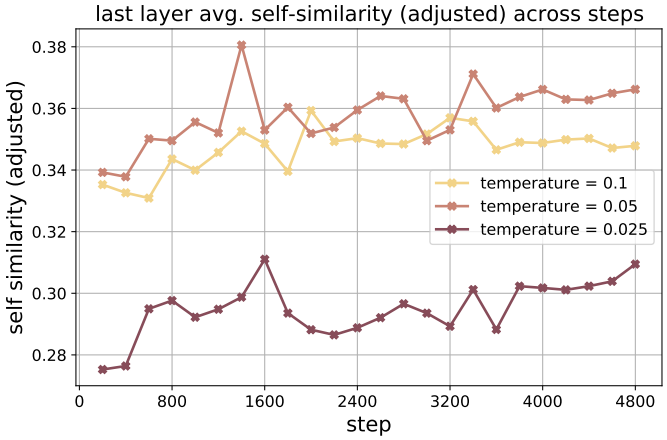


Figure 4.8: Self similarity under different temperature, adjusted by anisotropy baseline

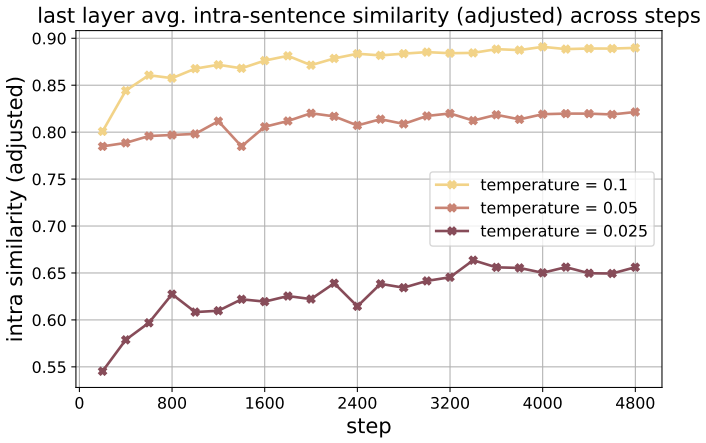


Figure 4.9: Intra-sentence similarity under different temperature, adjusted by anisotropy baseline

Figure 4.8 and Figure 4.9 present the adjusted self and intra-sentence similarity. Following the closer look at the contradicted pattern for frequency bias analyzed in Section 4.3, the behavior here becomes self-explainable. We could see that under the temperature of 0.1, the self similarity stays at a lower level compared to 0.05

in the last steps. This matches with the opposite result in intra-sentence similarity. According to our analysis in Section 4.3, it is the less meaningful tokens that drag down the self-similarity, and because they learn to follow the semantically meaningful tokens wherever their embeddings go in the semantic space, the corresponding intra-sentence similairty would become much higher. We speculate that, while a high intra similarity explains the performance gain of models trained with contrastive loss on semantic tasks, its being too intense (as shown when $\tau = 0.1$) might also account for the performance drop, making semantic meaningful tokens too dominating compared to auxiliary/functional tokens. Therefore, it again justifies the importance of selecting **a moderate temperature** that provides enough gradients, but not over-intensifying the attention leaning toward dominating tokens.

In Section 4.10, we provide the analysis on batch size, revealing that batch size plays a less significant role, if given a relatively optimal temperature. This is the opposite of what is commonly found in visual representation learning. Section 4.12 compares the three commonly used pooling methods, showing that the found patterns are not just artifacts of a certain pooling method (mean pooling), but consistent across pooling methods.

## 4.5 Limitations

This chapter only considers analyzing contrastive learning in the fine-tuning stage, but we note that with isotropy being a desiderata for pre-trained language models (49), recent works have considered incorporating contrastive objectives in the pre-training stage (29; 62). We leave analysis on this line of research for future work.

We further note that the analysis in this work focuses on theoretical properties occurred during contrastive SRL (e.g., high intra-sentence similarity), thus only focuses on semantic textual similarity (STS) data as a proof of concept. However, with the growing attention on contrastive learning, we argue that the typical STS-B is perhaps no longer sufficient for revealing the full ability of models trained with newer contrastive SRL frameworks. We call for a standard practice that the performance of contrastive SRL should be assessed on both semantic textual similarity and

information retrieval tasks (e.g., (14)). We leave analysis on information retrieval tasks leveraging our analysis pipeline for future studies. For example, how high intra-sentence similarity is related to the learned attention towards tokens that enable document retrieval with better performance.

## 4.6 Top Self Similarity Change (SSC): Token Examples

Table 4.3 presents top 10 positive and negative self similarity change of frequent tokens, before and after contrastive fine-tuning.

Although function tokens are found to be highly contextualized in pre-trained language models (49), this phenomenon is even intensified after contrastive fine-tuning. While for semantic tokens, the spurious contextualization is alleviated to a great extent.

## 4.7 Expanded semantic space (Eased Anisotropy)

We provide a visualization of embedding geometry change in Figure 4.1.

We first use the vanilla mpnet to encode the STS-B subset we have constructed. During fine-tuning, we save the models every 200 steps and use them to encode the subset, We find that with optimal hyperparameters, the representations go through less change after 200 steps. We perform UMAP dimensionality reduction on embeddings provided by models up to 1000 step to preserve better global structure, and visualize only vanilla and 200-step embeddings.

## 4.8 Unadjusted measures of Section 4.2.5

Figure 4.10 and Figure 4.11 display respectively the unadjusted avg. self similarity and intra-sentence similarity. These values as we elucidated in previous sections, however, are likely to be artifact of anisotropy, and therefore are supposed to be

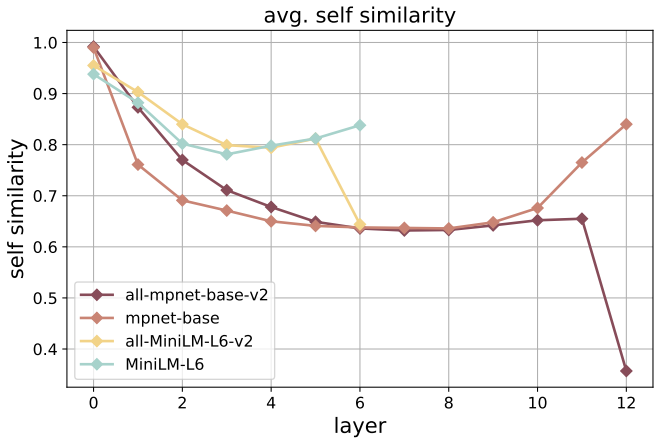


Figure 4.10: Unadjusted self similarity of tokens

adjusted by the anisotropy baseline of each model, based on the computation on randomly sampled token pairs.

As shown in main sections, to offset the effect of each model's intrinsic non-uniformity, we adjust them by the degree of anisotropy of each model, based on pair-wise average similarity among 1000 token representations that we randomly sample from each of the 1000 sentences (to avoid the sampling to bias toward long sentences).

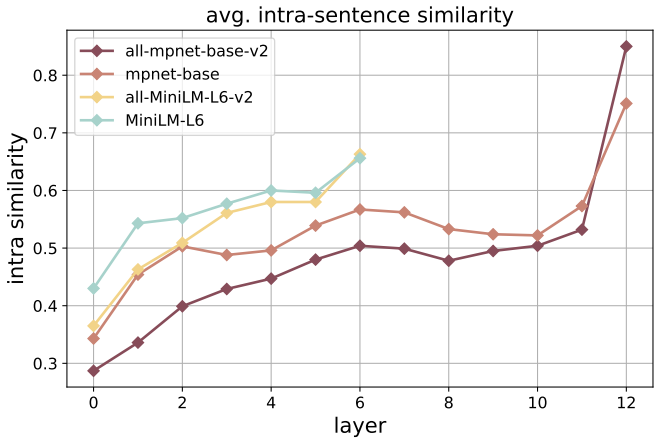


Figure 4.11: Unadjusted intra-sentence similarity of tokens

## 4.9 Temperature Search: why searching to the order of magnitude by 10 is not optimal?

We have also run the search range of temperature in previous research, which is carried out to the order of magnitude by 10. We compare the metrics on the models run with these temperatures with the vanilla mpnet model's performance.

It is shown that, not all values of temperature push the metrics from the vanilla baseline toward a same direction. Therefore, there exists a relatively optimal range to search, which is empirically implemented in a few works (47; 63), but few seems to have discussed why the range should not be that large, while we show this through the math analysis in Section 4.4 and their contradicted performance on our studied metrics here.

Specifically, for anisotropy baseline, temperature being too low even augments the vanilla model's unideal behavior, and the same applies for L2-norm, by that temperature being too low actually pushes the embeddings even further from the origin.

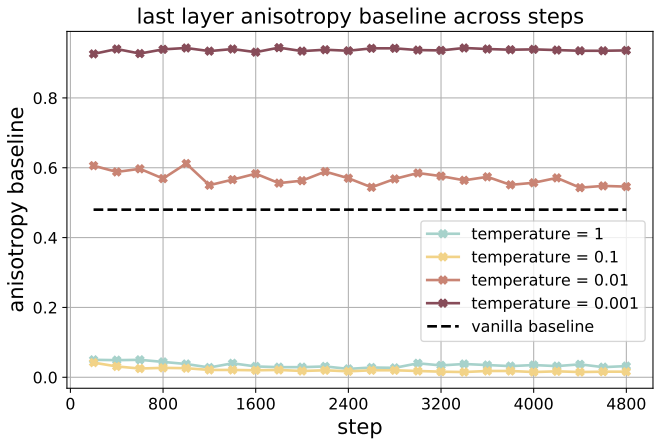


Figure 4.12: Anisotropy changes throughout training under different temperatures

For the adjusted self similarity and intra-sentence similarity, the metrics for low temperature are largely offset by anisotropy, meaning that for these temperature (especially $\tau = 0.001$), tokens are not more similar to itself in different contexts, nor to other tokens they share contexts with, compared to just with a random token in whatever context.

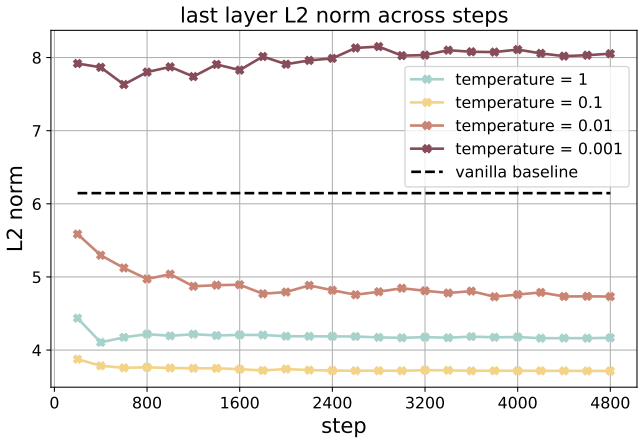


Figure 4.13: L2-norm under different temperatures

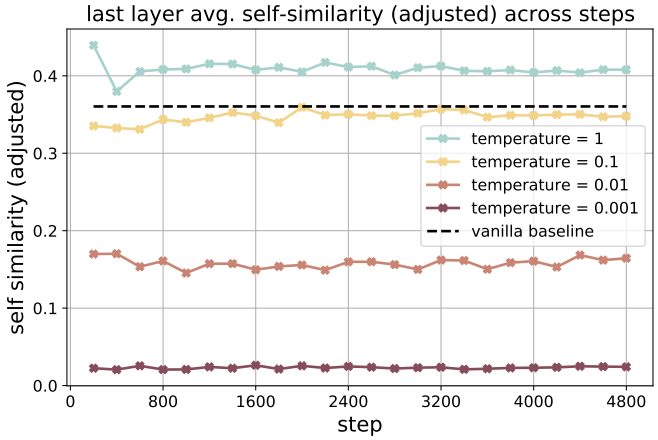


Figure 4.14: Self similarity under different temperatures, adjusted by anisotropy baseline

(50) and (3) take a singular spectrum perspective in understanding regularization to anisotropy. (50) propose a regularization term to the original log-likelihood loss in training machine translation model to mitigate the representation degeneration problem (or anisotropy). The regularization is proportional to $Sum(WW^T)$ , where $W$ is the stack of normalized word embeddings. If all elements are positive, then minimizing $Sum(WW^T)$ is equivalent to minimizing the upper bound for the largest top eigenvalue of $Sum(WW^T)$. Therefore, this regularization term shows theoretical promise to flatten the singular spectrum and make the representation more uniformly distributed around the origin. (3) extend this analysis to show the same theoretical promise brought by the uniformity loss proposed by (60), by deriving that uniformity

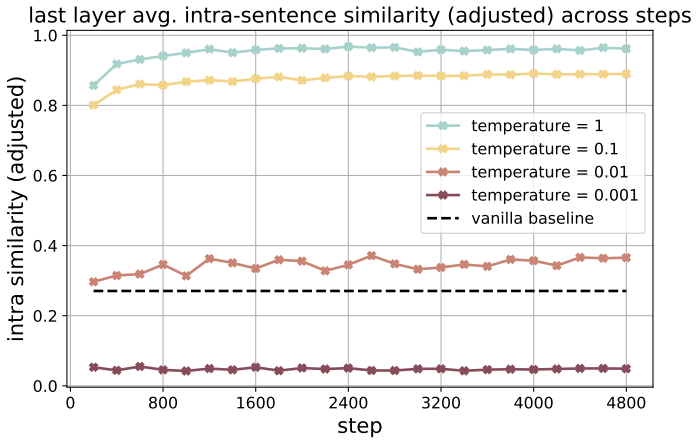


Figure 4.15: Intra-sentence similarity under different temperatures, adjusted by anisotropy baseline

loss is in fact greater or equal to $\frac{1}{\tau m^2} \sum_{i=1}^{m} \sum_{j=1}^{m} h_i^T h_j$, which is also equivalent to flattening the spectrum of the similarity matrix. Our results show that despite the intuition reached by singular spectrum perspective, the assumption could probably only hold on a relatively optimal temperature. Thus, the effect of temperature should be considered using this perspective, which is beyond the scope of this chapter.

## 4.10 Batch size

Batch size on the other hand, does not produce impact as significant as temperature. We have run three models with the optimal $\tau = 0.05$ paired with a batch size range of $\{16, 64, 256\}$.

The metrics yielded by different batch sizes all stay in small range at the end of the epoch, albeit showing different rates and stability of convergence.

| Model | k = 1 | k = 2 | k = 3 | k = 5 | k = 10 | k = 20 | k = 50 | k = 100 | k = 300 | k = 700 |
|---|---|---|---|---|---|---|---|---|---|---|
| $\text{mpnet}_{\text{vanilla}}$ | .386 | .338 | .210 | .169 | .168 | .182 | .201 | .195 | .175 | .040 |
| $\text{mpnet}_{\text{fine-tuned}}$ | **.999** | **.998** | **.996** | **.994** | **.990** | **.983** | **.960** | **.922** | **.783** | **.229** |
| $\text{minilm}_{\text{vanilla}}$ | .993 | **.980** | **.970** | **.947** | **.886** | .796 | .559 | .543 | **.375** | / |
| $\text{minilm}_{\text{fine-tuned}}$ | **.998** | .846 | .836 | .830 | .817 | **.805** | **.768** | **.690** | .285 | / |

Table 4.4: $r^2$ between the similarity matrices of sampled token embeddings, before and after removing the same top-k rogue dimensions from every token embedding.

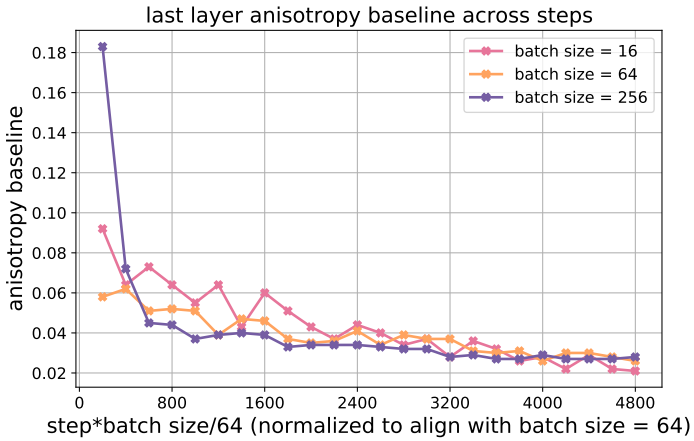


Figure 4.16: Anisotropy changes throughout training under different batch sizes

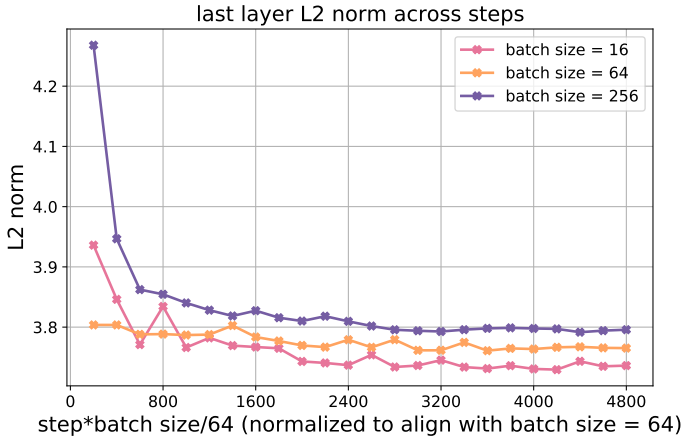


Figure 4.17: L2-norm under different batch sizes

## 4.11 Informativity

In this section we present the informativity analysis outlined in Section 4.2. Specifically, after we identify how dominant are the top rogue dimensions, to what degree is semantics affected with these rogue dimensions removed? Do these dimensions only have large mean but do not contribute to large variance? We sample 1k token embeddings to compute their pair-wise similarity. After removing top-k dimensions from every embedding, we compute the similarity matrix again, and compute the Pearson Correlation $r$ between flattened lower triangles of the matrices of the two excluding their diagonals. We then report the $r^2$ which represents the proportion of variance in the original similarity matrix explained by the post-processed matrix.

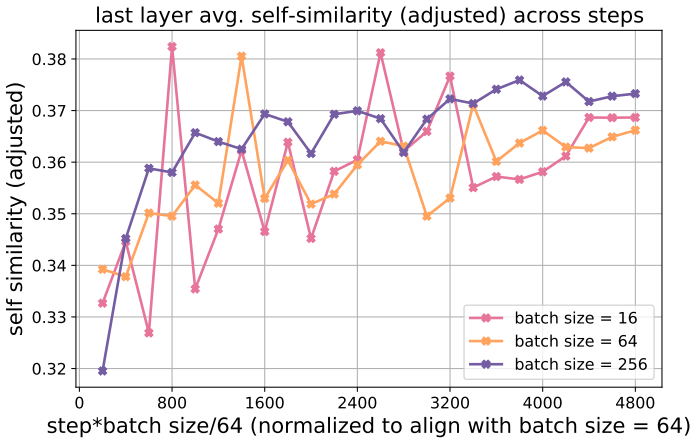


Figure 4.18: Self similarity under different batch sizes, adjusted by anisotropy baseline

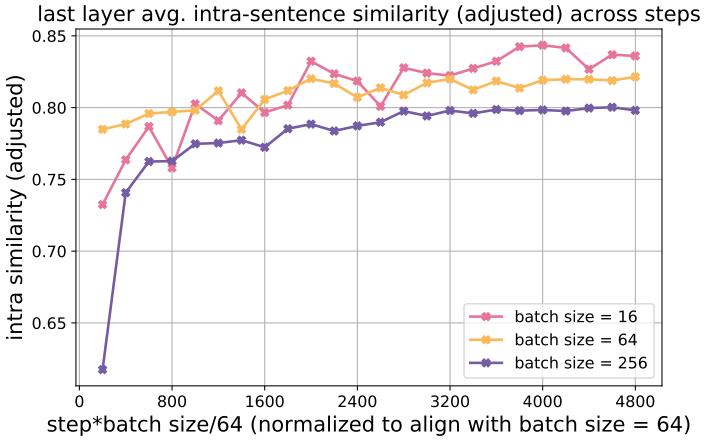


Figure 4.19: Intra-sentence similarity under different batch sizes, adjusted by anisotropy baseline

At a high level, Table 4.4 shows that dominance $\neq$ informativity. Specifically, MiniLM presents a misalignment between dominance toward similarity computation and the actual information stored in these dimensions. For instance, removing the top 1 dominant dimension of $\text{minilm}_{\text{finetuned}}$ seems to not affect the embeddings' relative similarity to one another at all, preserving an $r^2$ of .998. Also, recall from Section 4.2 that contributions of dimensions from $\text{minilm}_{\text{vanilla}}$ to similarity computation are relatively flatter than $\text{mpnet}_{\text{vanilla}}$, the results show that along with the even more flattened contributions after fine-tuning, the informativity seems to have been reallocated. For instance, from removing $k = 100$ to $k = 300$, the explainable variance goes down from .690 to .285, meaning this range of dimensions

store a lot more information compared with the vanilla version. In general, that $\text{minilm}_{\text{vanilla}}$ and $\text{minilm}_{\text{fine-tuned}}$ take turn to yield higher $r^2$ with top-k removed demonstrates that there is generally no strong correlation between dominance and informativity, but it is rather random - especially when the dominance is already quite evenly distributed in the vanilla model.

## 4.12 Pooling Method

In line with previous analysis, this section presents the measurement on different pooling methods. We follow the same setting in Section 4.4 to also investigate whether the patterns found in Section 4.2 are only attributable to mean pooling. We compare mean pooling with [cls] pooling and max pooling. Albeit the different performance on the metrics, contrastive learning in general presents consistent behaviors across pooling methods, such as eased anisotropy and enhanced intra-sentence similarity For anisotropy, we observe that [cls] pooling shows a slow convergence on producing isotropy. At the end of the epoch, it is still on a decreasing trend. By contrast, mean pooling and max pooling demonstrate a faster convergence, with mean pooling being most promising on isotropy. Their performance on L2-norm is also well-aligned, again showing strong correlation between isotropy and L2-norm in the training process utilizing contrastive loss. And this correlation seems agnostic to pooling methods. The following analysis focuses on their differences:

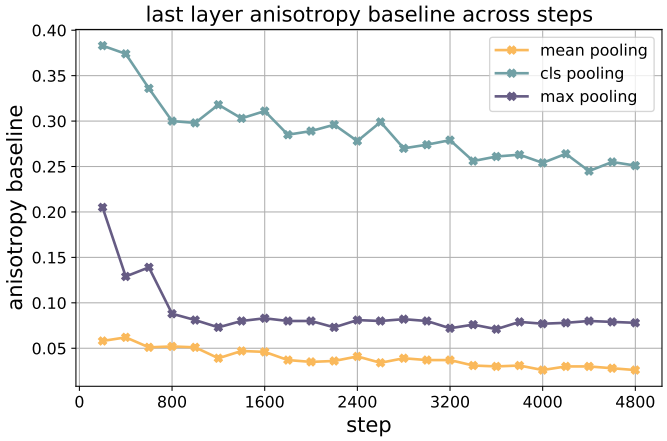


Figure 4.20: Anisotropy changes throughout training under different pooling methods

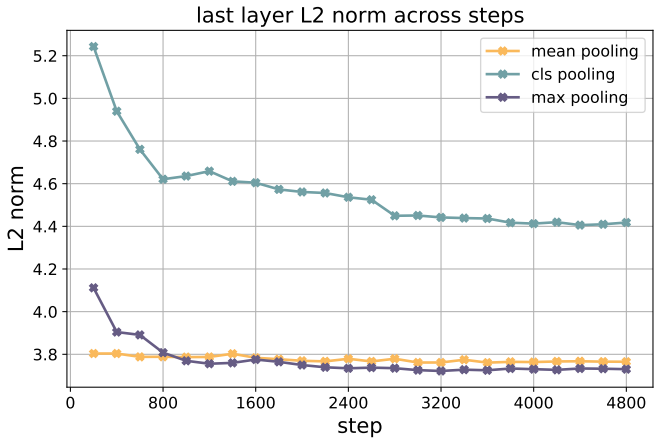


Figure 4.21: L2-norm under different pooling methods

For self similarity, [cls] pooling and mean pooling show a similar performance, which max pooling deviates from.

Max pooling presents an "unacceptably" high intra-sentence similarity. Although intra-sentence similarity is a potentially ideal property uniquely brought by contrastive learning, this metric could not be over-intensified, as also shown in Section 4.4, Section 4.9, and Section 4.10. There exists an ideal range for intra-sentence similarity, compatible to a model's performance on other metrics.

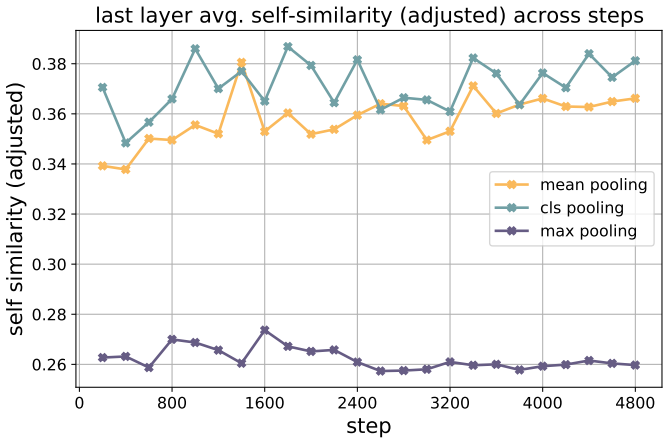


Figure 4.22: Self similarity under different pooling methods, adjusted by anisotropy baseline

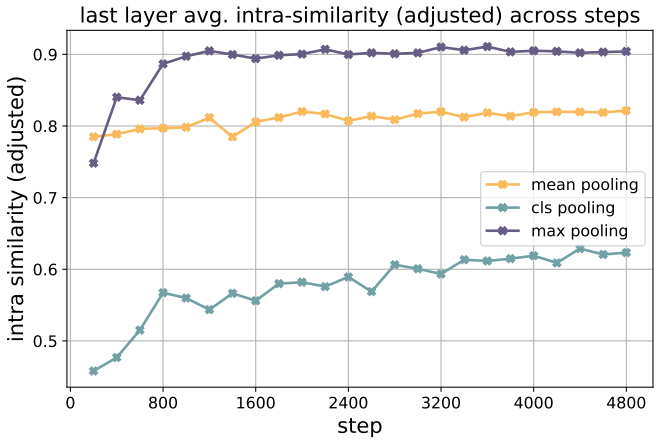


Figure 4.23: Intra-sentence similarity under different pooling methods, adjusted by anisotropy baseline

## 4.13 Self Similarity Change and Correlation across Models

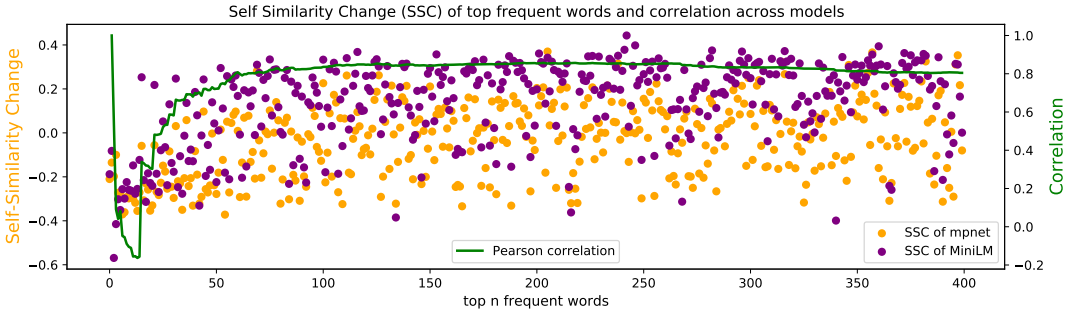


Figure 4.24: Self Similarity Change

In Figure 4.24 we plot the Self Similarity Change (SSC) across models (mpnet and MiniLM), for the top 400 frequent tokens of the SST-b subset we construct.

The Pearson correlation between the two accumulated lists of the first $[:n]$ tokens is also plotted. The perfect correlation at the beginning is ignored because the most frequent words at the top are the [pad], [cls] and [sep] tokens. Excluding these, the correlation reaches the peak at 204 as mentioned in the main section, before which the correlation has been slowly stabilized with more tokens considered, while starting to drop after. This shows that the pattern mostly holds for tokens that are above certain frequency, which again provides empirical ground for our analysis on drawing

connection of self and intra-sentence similarity to frequency bias.

## 4.14 Conclusion

In this chapter, we demystify the successes of using contrastive objectives for sentence representation learning through the lens of isotropy and learning dynamics. We showed the theoretical promise of uniformity brought by contrastive learning through measuring anisotropy, complemented by showing the flattened domination of top dimensions. We then uncovered a very interesting yet under-covered pattern: contrastive learning learns to converge tokens in a same sentence, bringing extremely high intra-sentence similarity. We then explained this pattern by connecting it to frequency bias, and showed that semantically functional tokens fall back to be the by products of semantically meaningful tokens in a sentence, following wherever they travel in the semantic space. Lastly, we ablate all findings through temperature, batch size and pooling method, providing a closer look at these patterns through different angles.

As we will show in following chapters, findings of this chapter provide transferable analysis angles to different behavioral patterns of embedding models. For instance, The finding that contrastive SRL induces very high intra-sentence similarity and length-sensitive isotropy motivates Chapter 5's analysis of lenght attacks and the design of the LA(SER)$^3$ method.

# CHAPTER 5

## Learning Sentence Representation for Retrieval

Dense representation plays a dominant role in information retrieval. It has become the technique behind applications like search engines. Current dense retrievers are mostly facilitated by training with contrastive learning. Although contrastive learning is powerful, we show that there remains vulnerability in models trained with CL. In this chapter, we study an important property for text retrieval models, length robustness, a property central for retrievers to retrieve documents of different lengths accurately. We show that properties we analyzed in Chapter 4 are important to understand why models are not robust to documents across different lengths. For example, we find the isotropy promise found in Chapter 4 is highly length-sensitive: when models are only trained on short documents, their long document embedding space remains anisotropic. We also propose the LA(SER)$^3$, a length-robust dense retriever training framework, showing that we can effectively train state-of-the-art embedding models solely through addressing the length vulnerability.

In recent years, contrastive learning (CL) has been extensively utilized to recover sentence and document-level encoding capability from pre-trained language models. In this chapter, we question the length generalizability of CL-based models, i.e., their vulnerability towards length-induced semantic shift. We verify not only that

length vulnerability is a significant yet overlooked research gap, but we can devise unsupervised CL methods solely depending on the semantic signal provided by document length. We first derive the theoretical foundations underlying length attacks, showing that elongating a document would intensify the high intra-document similarity that is already brought by CL. Moreover, we found that isotropy promised by CL is highly dependent on the length range of text exposed in training. Inspired by these findings, we introduce a simple yet universal document representation learning framework, **LA(SER)$^3$**: length-agnostic self-reference for semantically robust sentence representation learning, achieving state-of-the-art unsupervised performance on the standard information retrieval benchmark.

## 5.1 Introduction

In recent years, contrastive learning (CL) has become the go-to method to train representation encoder models (3; 21; 22; 33). In the field of natural language processing (NLP), the effectiveness of the proposed unsupervised CL methods is typically evaluated on two suites of tasks, namely, semantic textual similarity (STS) (13) and information retrieval (IR) (e.g., (14)). Surprisingly, a large number of works only validate the usefulness of the learned representations on STS tasks, indicating a strong but widely-adopted assumption that methods optimal for STS could also provide natural transferability to retrieval tasks.

Due to the document length misalignment of these two types of tasks, the potential gap in models' capability to produce meaningful representation at different length ranges has been rarely explored (30). Studies of document length appear to have been stranded in the era where methods are strongly term frequency-based, because of the explicit reflection of document length to sparse embeddings, with little attention given on dense encoders. Length preference for dense retrieval models is observed by (14), who show that models trained with dot-product and cosine similarity exhibit different length preferences. However, this phenomenon has not been attributed to the distributional misalignment of length between training and inference domains/tasks, and it remains unknown what abilities of the model are

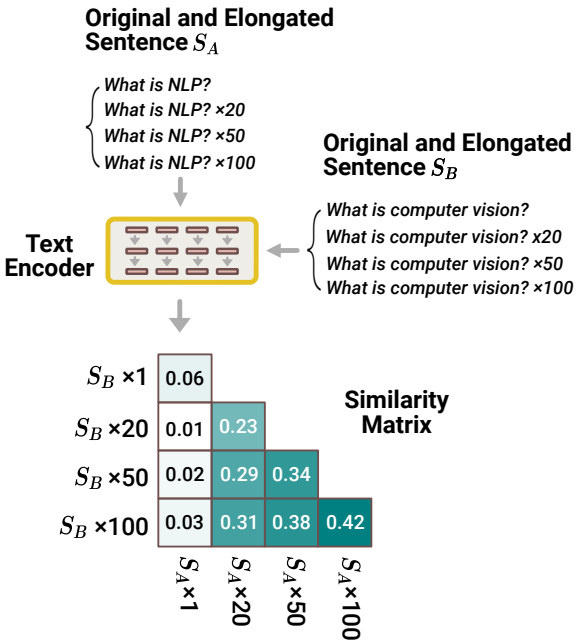


Figure 5.1: Demonstration of Elongation Attack on Sentence Similarity. The similarity between sentence $S_A$ and $S_B$ incorrectly increases along with elongation, i.e., copy-and-concatenate the original sentence for multiple times, despite the semantics remain unaltered.

enhanced and diminished when trained with a certain length range. On the other hand, while there are a line of unsupervised IR methods (2; 3; 4), none provides a general length-robust training recipe or analyses length-dependent properties in embedding models, such as isotropy and intra-sentence similarity.

In this chapter, we provide an extensive analysis of length generalizability of standard contrastive learning methods. Our findings show that, with default contrastive learning, models' capability to encode document-level semantics largely comes from their coverage of length range in the training.

We first depict through derivation the theoretical underpinnings of the models' vulnerability towards length attacks. Through attacking the documents by the simple copy-and-concatenating elongation operation, we show that the vulnerability comes from the further intensified high intra-document similarity that is already pronounced after contrastive learning. This hinders a stable attention towards the semantic tokens in inference time. Further, we show that, the uniformity/isotropy promised by contrastive learning is heavily length-dependent. That is, models' encoded embeddings are only isotropic on the length range seen in the training,

but remain anisotropic otherwise, hindering the same strong expressiveness of the embeddings in the unseen length range.

In the quest to bridge these unideal properties, we propose a simple yet universal framework, **LA(SER)$^3$**: **L**ength-**A**gnostic **SE**lf-**R**eference for **SE**mantically **R**obust **SE**ntence **R**epresentation learning. By providing the simple signal that *"the elongated version of myself 1) should still mean myself, and thus 2) should not become more or less similar to my pairs"*, this framework could not only act as an unsupervised contrastive learning method itself by conducting self-referencing, but could also be combined with any contrastive learning-based text encoding training methods in a plug-and-play fashion, providing strong robustness to length attacks and enhanced encoding ability.

We show that, our method not only improves contrastive text encoders' robustness to length attack without sacrificing their representational power, but also provides them with external semantic signals, leading to state-of-the-art unsupervised performance on the standard information retrieval benchmark.

## 5.2 Length-based Vulnerability of Contrastive Text Encoders

Length preference of text encoders has been observed in the context of information retrieval (14), showing that contrastive learning-based text encoders trained with dot-product or cosine similarity display opposite length preferences. (30) further devised "adversarial length attacks" to text encoders, demonstrating that this vulnerability can easily fool text encoders, making them perceive a higher similarity between a text pair by only copying one of them $n$ times and concatenating to itself.

In this section, we first formalize the problem of length attack, and then analyze the most important pattern (misaligned intra-document similarity) that gives rise to this vulnerability, and take an attention mechanism perspective to derive for the first time the reason why contrastive learning-based text encoders can be attacked.

**Problem Formulation: Simple Length Attack** Given a sentence $S$ with $n$ tokens $\{x_1, x_2, ..., x_n\}$, we artificially construct its elongated version by copying it $m$ times, and concatenating it to itself. For instance, if $m = 2$, this would give us $\widetilde{S} = \{x_1, ..., x_n, x_1, ..., x_n\}$.

In the context of information retrieval, where downstream tasks are mostly defined by topical matching of queries and documents, we hold a **semantics-preserving assumption** for repetition-based elongation operation, where repeating a document $d$ by $m$ time should not make it more similar to a query $q$. However, we do acknowledge edge cases where repetitions may change pragmatics, such as sentiments (e.g., cases where repeating a positive statement many times can become sarcastic), which is not the focus of information retrieval tasks.

In fact, using pure statistical representation such as tf-idf (64), the original sentence and the elongated version yield exact same representations:

$$\widetilde{S} \triangleq f(S, m) \tag{5.1}$$

$$\text{tf-idf}(S) = \text{tf-idf}(\widetilde{S}) \tag{5.2}$$

where $f(\cdot)$ denotes the elongation operator, and $m$ is a random integer.

Therefore, no matter according to the semantics-preserved assumption discussed previously, or reference from statistics-based methods (64), one would hypothesize Transformer-based models to behave the same. Formally, we expect, given a Transformer-based text encoder $g(\cdot)$ to map a document into a document embedding, we could also (**ideally**) get:

$$g(S) = g(\widetilde{S}) \tag{5.3}$$

**Observation 1: Transformer-based text encoders perceive different semantics in original texts and elongation-attacked texts.** The central problem is: given a Transformer-based text encoder $g(\cdot)$, it is found empirically that:

$$g(S) \neq g(\widetilde{S}). \tag{5.4}$$

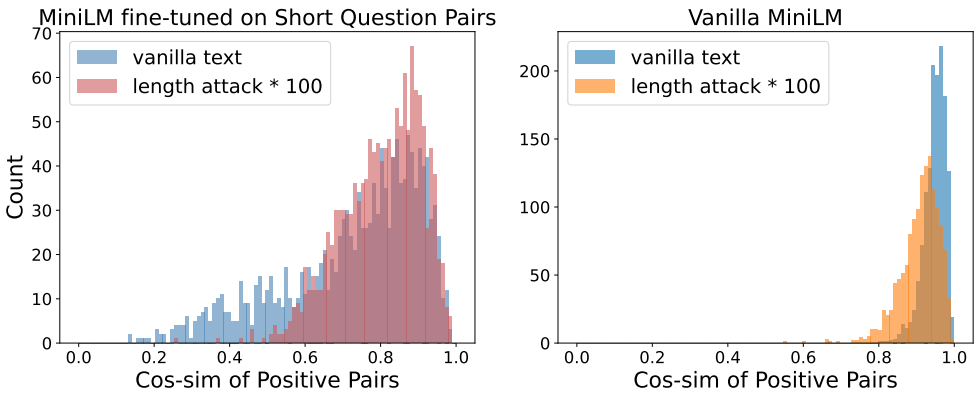


Figure 5.2: Distribution of positive pair cosine similarity. Left: MiniLM finetuned on only short document pairs with contrastive loss displays a favor towards attacked documents (longer documents). Right: the vanilla model displays an opposite behavior.

We verify this phenomenon with Proof of Concept Experiment 1 (Figures 5.1, 5.2), showing that Transformer-based encoders perceive different semantics before and after elongation attacks.

**Proof of Concept Experiment 1** To validate Observation 1, we fine-tune a vanilla MiniLM (59) with the standard infoNCE loss (20) with in-batch negatives, on the Quora duplicate question pair dataset (QQP). Notably, the dataset is composed of questions, and thus its length coverage is limited (average token length = 13.9, with 98.5% under 30 tokens).

With the fine-tuned model, we first construct two extreme cases: one with a false positive pair ("*what is NLP?*" v.s., "*what is computer vision?*"), one with a positive pair ("*what is natural language processing?*" v.s., "*what is computational linguistics?*"). We compute cosine similarity between mean-pooled embeddings of the original pairs, and between the embeddings attained after conducting an elongation attack with $m = 100$ (Eq. 5.1).

We found surprisingly that, while "*what is NLP?*" and "*what is computer vision?*" have 0.06 cosine similarity, their attacked versions achieve 0.42 cosine similarity - successfully attacked (cf. Figure 5.1). And the same between "*what is natural language processing?*" and "*what is computational linguistics?*" goes from 0.50 to 0.63 - similarity pattern augmented.

On a larger scale, we then construct an inference set with all the document pairs from Semantic Textual Similarity benchmark (STS-b) (13). We conduct an elongation attack on all sentences with $m = 100$ (Eq. 5.1). The distributions of document pair cosine similarity are plotted in Figure 5.2. For the fine-tuned MiniLM (Figure 5.2, left), it is clearly shown that, the model perceives in general a higher cosine similarity between documents after elongation attacks, greatly increasing the perceived similarity, even for pairs that are not positive pairs. This phenomenon indicates a built-in vulnerability in contrastive text encoders, hindering their robustness for document encoding. For reference, we also plot out the same set of results on the vanilla MiniLM (Figure 5.2, right), demonstrating an opposite behavior, which will be further discussed in Proof of Concept Experiment 2.

**Observation 2: Intra-document token interactions experience a pattern shift after elongation attacks.** Taking an intra-document similarity perspective (49), we can observe that, tokens in the elongated version of same text, do not interact with one another as they did in the original text (see Proof of Concept Experiment 2). Formally, given tokens in $S$ providing an intra-document similarity of $sim$, and tokens in the elongated version $\widetilde{S}$ providing $\tilde{sim}$, we will show that $sim \neq \tilde{sim}$. This pattern severely presents in models that have been finetuned with a contrastive loss, while is not pronounced in their corresponding vanilla models (PoC Experiment 2, Figure 5.3).

A significant increase on intra-document similarity of contrastive learning-based models is observed by (28), opposite to their vanilla pre-trained checkpoints (49). It is further observed that, after contrastive learning, semantic tokens (such as topical words) become *dominant* in deciding the embedding of a sentence, while embeddings of functional tokens (such as stop-words) follow wherever these semantic tokens travel in the embedding space. This was formalized as the "entourage effect" (28). Taking into account this conclusion, we further derive from the perspective of attention mechanism, the reason why conducting elongation attacks would further intensify the observed high intra-document similarity.

The attention that any token $x_i$ in the sentence $S$ gives to the dominant tokens

can be expressed as:

$$\text{Attention}(\underset{i\in S}{x_i} \rightarrow x_{\text{dominant}}) = \frac{e^{q_i k^T_{dominant}/\sqrt{d_k}}}{\sum\limits_n e^{q_i k^T_n/\sqrt{d_k}}}, \tag{5.5}$$

where $q_i$ is the query vector produced by $x_i$, $k^T_{dominant}$ is the transpose of the key vector produced by $x_{dominant}$, and $k^T_n$ is the transpose of the key vector produced by every token $x_n$. We omit the $V$ matrix in the attention formula for simplicity.

After elongating the sentence $m$ times with the copy-and-concat operation, the attention distribution across tokens shifts, taking into consideration that the default prefix `[cls]` token is not elongated. Therefore, in inference time, `[cls]` tokens share less attention than in the original sentence.

To simplify the following derivations, we further impose the assumption that positional embeddings contribute little to representations, which loosely hold empirically in the context of contrastive learning (65). In Section 5.6, we conduct an extra group of experiment to present the validity of this imposed assumption by showing the positional invariance of models after CL.

With this in mind, after elongation, the same token in different positions would get the same attention, because they have the same token embedding without positional embeddings added. Therefore:

$$\begin{aligned}
&\widetilde{\text{Attention}}(\underset{i\in \widetilde{S}}{x_i} \rightarrow x_{\text{dominant}}) \\
=& \frac{m e^{q_i k^T_{dominant}/\sqrt{d_k}}}{m\sum\limits_n e^{q_i k^T_n/\sqrt{d_k}} - (m-1)e^{q_i k^T_{[cls]}/\sqrt{d_k}}} \\
=& \frac{e^{q_i k^T_{dominant}/\sqrt{d_k}}}{\sum\limits_n e^{q_i k^T_n/\sqrt{d_k}} - \frac{m-1}{m} e^{q_i k^T_{[cls]}/\sqrt{d_k}}} \\
>& \text{Attention}(\underset{i\in S}{x_i} \rightarrow x_{\text{dominant}})
\end{aligned} \tag{5.6}$$

Based on Eq. 5.6, we can see that attentions towards dominant tokens would increase after document elongation attack. However, we can also derive that the

same applies to non-dominant tokens:

$$\widetilde{\text{Attention}}(\underset{i\in\widetilde{S}}{x_i} \rightarrow x_{\text{non-dominant}})$$
$$> \text{Attention}(\underset{i\in S}{x_i} \rightarrow x_{\text{non-dominant}})$$

In fact, every unique token except `[cls]` would experience an attention gain. Therefore, we have to prove that, the attention gain $G_d$ of dominant tokens (denoted as $x_d$) outweighs the attention gain $G_r$ of non-dominant (regular, denoted as $x_r$) tokens. To this end, we define:

$$G_d \triangleq \widetilde{\text{Attention}}(\underset{i\in\widetilde{S}}{x_i} \rightarrow x_{\text{d}}) - \text{Attention}(\underset{i\in S}{x_i} \rightarrow x_{\text{d}})$$

$$G_r \triangleq \widetilde{\text{Attention}}(\underset{i\in\widetilde{S}}{x_i} \rightarrow x_{\text{r}}) - \text{Attention}(\underset{i\in S}{x_i} \rightarrow x_{\text{r}})$$

Let $e^{q_i k^T_{\text{dominant}}/\sqrt{d_k}}$ be $l_d$, $e^{q_i k^T_{\text{non-dominant}}/\sqrt{d_k}}$ be $l_r$, $e^{q_i k^T_n/\sqrt{d_k}}$ be $l_n$, and $e^{q_i k^T_{[cls]}/\sqrt{d_k}}$ be a $l_c$, we get:

$$\begin{aligned} & G_d \\ & \triangleq \widetilde{\text{Attention}}(\underset{i\in\widetilde{S}}{x_i} \rightarrow x_{\text{d}}) - \text{Attention}(\underset{i\in S}{x_i} \rightarrow x_{\text{d}}) \\ & = \frac{l_d}{\sum\limits_n l_n - \frac{m-1}{m} l_c} - \frac{l_d}{\sum\limits_n l_n} = \frac{l_d \frac{m-1}{m} l_c}{\sum\limits_n l_n (\sum\limits_n l_n - \frac{m-1}{m} l_c)} \end{aligned} \tag{5.7}$$

Similarly, we get:

$$G_r = \frac{l_r \frac{m-1}{m} l_c}{\sum\limits_n l_n (\sum\limits_n l_n - \frac{m-1}{m} l_c)} \tag{5.8}$$

Also note that $l_d > l_r$: that's why they are called "dominating tokens" in the first place (28). Therefore, we prove that $G_d > G_r$.

As a result, with elongation operation, every token is going to assign even more

attention to the embeddings of the dominating tokens. And this effect propagates throughout layers, intensifying the high intra-document similarity ("entourage effect") found in (28).

In summary, we show that elongation increases attention to dominant tokens, amplifying intra-document similarity and harming stable semantics. Note that in above derivations, we impose an assumption that positional embeddings contribute little to representations after contrastive learning, which we further empirically validate in Section 5.6.1.

**Proof of Concept Experiment 2** With the derivations, we conduct PoC Experiment 2, aiming to demonstrate that intra-document similarity experiences a pattern shift after elongation attack, intensifying the "entourage effect", for contrastive fine-tuned models.

Taking the same fine-tuned MiniLM checkpoint from PoC Experiment 1, we compute the intra-document similarity of all the model outputs on STS-b. For each document, we first compute its document embedding by mean-pooling, then compute the average cosine similarity between each token embedding and the document embedding.[1] The results are shown in Figure 5.3. After elongation attacks, we can see an increase in the already high intra-document similarity, meaning that all other tokens converge even further towards the tokens that dominate the document-level semantics.

When using the vanilla MiniLM checkpoint, the intra-document similarity pattern is again reversed. This opposite pattern is well-aligned with the findings of (49) and (28): Because in vanilla language models, the intra-document similarity generally becomes lower in the last few layers, while after contrastive learning, models show a drastic increase of intra-document similarity in the last few layers. Also, our derivations conclude that: if the intra-document similarity shows an accumulated increase in the last few layers, this increase will be intensified after elongation; and less affected otherwise.

[1]Notably, we further adjust these scores by the model's anisotropy estimation (average pair-wise similarity of random sampled tokens), because of the representation degeneration problem (49; 50).

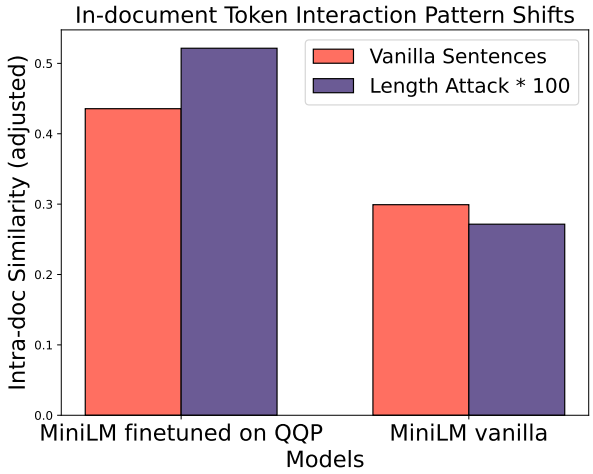


Figure 5.3: In-document Token Interactions experience a pattern shift before and after contrastive fine-tuning: Using the vanilla model, tokens in the elongated version of a document become less like one another than in the original un-attacked text; after contrastive fine-tuning, tokens in the attacked text look more alike to one another. This empirically validates our math derivation. Notably, measurements of both models have been adjusted by their anisotropy estimation (displayed value = avg. intra similarity - estimated anisotropy value).

Complementing the intensified intra-document similarity, we also display an isotropy misalignment before and after elongation attacks in Figure 5.4. With the well-known representation degeneration or anisotropy problems in vanilla pre-trained models (Figure 5.4, right, green, (49; 50)), it has been previously shown that after contrastive learning, a model's encoded embeddings will be promised with a more isotropic geometry (Figure 5.4, left, green, (3; 28; 60)). However, in this chapter, we question this general conclusion by showing that the promised isotropy is strongly length-dependent. After elongation, the embeddings produced by the fine-tuned checkpoint start becoming anisotropic (Figure 5.4, left, pink). This indicates that, if a model has only been trained on short documents with contrastive loss, only the short length range is promised with isotropy.

On the other hand, elongation attacks seem to be able to help vanilla pre-trained models to escape from anisotropy, interestingly (Figure 5.4, right, pink). However, the latter is not the key focus of this chapter.

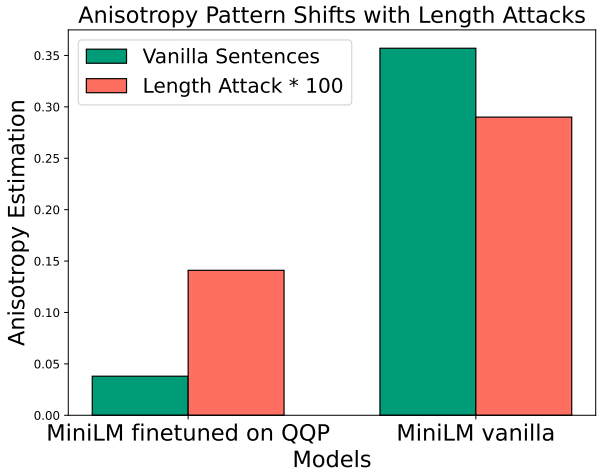


Figure 5.4: Isotropy Pattern Shifts. Albeit contrastive learning has an isotropy promise, we question this by showing the model is only isotropic in its trained length range, remaining anisotropic otherwise (shown by increased anisotropy after length attacks).

## 5.3 Method: LA(SER)$^3$

| | Models → | SimCSE† | | ESimCSE† | | DiffCSE† | | InfoCSE†♠ | | LA(SER)$^3$ (Ours) | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Test Dataset ↓ | base | large | base | large | base | large | base | large | base-64 | base-128 |
| zero-shot setting | **trec-covid** | 0.2750 | 0.2264 | 0.2291 | 0.2829 | 0.2368 | 0.2291 | **0.3937** | 0.3166 | 0.2728 | 0.3463 |
| | **nfcorpus** | 0.1048 | 0.1356 | 0.1149 | 0.1483 | 0.1204 | 0.1470 | 0.1358 | 0.1576 | 0.1652 | **0.1919** |
| | **nq** | 0.1628 | 0.1671 | 0.0935 | 0.1705 | 0.1188 | 0.1556 | **0.2023** | 0.1790 | 0.1556 | 0.1354 |
| | **fiqa** | 0.0985 | 0.0975 | 0.0731 | **0.1117** | 0.0924 | 0.1027 | 0.0991 | 0.1000 | 0.1057 | 0.1090 |
| | **arguana** | 0.2796 | 0.2078 | 0.3376 | 0.2604 | 0.2500 | 0.2572 | 0.3244 | 0.4133 | 0.4182 | **0.4227** |
| | **webis-touche2020** | **0.1342** | 0.0878 | 0.0786 | 0.1057 | 0.0912 | 0.0781 | 0.0935 | 0.0920 | 0.1105 | 0.1167 |
| | **quora** | 0.7375 | 0.7511 | 0.7411 | 0.7615 | 0.7491 | 0.7471 | 0.8241 | **0.8268** | 0.7859 | 0.7741 |
| | **cqadupstack** | 0.1349 | 0.1082 | 0.1276 | 0.1196 | 0.1197 | 0.1160 | **0.2097** | 0.1881 | 0.1687 | 0.1691 |
| | **dbpedia-entity** | 0.1662 | 0.1495 | 0.1260 | 0.1650 | 0.1537 | 0.1571 | **0.2101** | 0.1838 | 0.1645 | 0.1663 |
| | **scidocs** | 0.0611 | 0.0688 | 0.0657 | 0.0796 | 0.0673 | 0.0699 | 0.0837 | **0.0859** | 0.0764 | **0.0859** |
| | **climate-fever** | **0.1420** | 0.1065 | 0.0796 | 0.1302 | 0.1019 | 0.1087 | 0.0937 | 0.0840 | 0.1311 | 0.1197 |
| | **scifact** | 0.2492 | 0.2541 | 0.3013 | 0.2875 | 0.2666 | 0.2811 | 0.3269 | 0.3801 | 0.3960 | **0.4317** |
| | **hotpotqa** | 0.2382 | 0.1896 | 0.1213 | 0.1970 | 0.1730 | 0.2068 | **0.3177** | 0.2781 | 0.2827 | 0.2937 |
| | **fever** | **0.2916** | 0.1776 | 0.0756 | 0.1689 | 0.1416 | 0.1849 | 0.1978 | 0.1252 | 0.2388 | 0.2691 |
| | **average** | 0.2197 | 0.1948 | 0.1832 | 0.2135 | 0.1916 | 0.2030 | 0.2509 | 0.2436 | 0.2480 | **0.2594** |

Table 5.1: Unsupervised BERT nDCG@10 performances on BEIR information retrieval benchmark. †: Results are from (2). ♠: Unfair comparison. Notably, InfoCSE benefits from the pre-training of an auxiliary network, while the rest of the baselines and our method fully rely on unsupervised contrastive fine-tuning on the same `training`$^{\texttt{wiki}}$ setting as described in §5.4. Note that with a batch size of 64, our method already outperforms all baselines to a large margin except InfoCSE. Since we train with a max sequence length of 256 (all baselines are either 32 or 64), we find that training with a larger batch size (128) further stabilizes our training, achieving state-of-the-art results. Further, we achieve state-of-the-art with only a BERT$_{\text{base}}$.

After examining the two fundamental reasons underlying the built-in vulner-

ability brought by standard contrastive learning, the formulation of our method emerges as an intuitive outcome. Naturally, we explore the possibility of using only *length* as the semantic signal to conduct contrastive sentence representation learning, and propose **LA(SER)$^3$**: Length-Agnostic Self-Reference for Semantically Robust Sentence Representation Learning. LA(SER)$^3$ builds upon the semantics-preserved assumption that *"the elongated version of myself 1) should still mean myself, and thus 2) should not become more or less similar to my pairs"*. **LA(SER)$^3$** leverages elongation augmentation during the unsupervised constrastive learning to improve 1) the robustness of in-document interaction pattern in inference time; 2) the isotropy of larger length range. We propose two versions of reference methods, for different format availability of sentences in target training sets.

**Self-reference** In LA(SER)$^3_{\text{self-ref}}$ setting, we take a sentence from the input as an anchor for each training input, and construct its positive pair by elongating the sentence to be $\boldsymbol{m}$ times longer.

**Intra-reference** LA(SER)$^3_{\text{intra-ref}}$ conducts intra-reference within the document. The two components of a positive pair are constructed from different spans of the same document. Since we are only to validate effectiveness of LA(SER)$^3_{\text{intra-ref}}$, we implement this in the simple mutually-excluded span setting. In other words, the LA(SER)$^3_{\text{intra-ref}}$ variant takes a sentence (either the first or a random sentence) from the text as an anchor, uses the rest of the text in the input as its positive pair, and elongates the anchor sentence $\boldsymbol{m}$ times as the augmented anchor.

For both versions, we use the standard infoNCE loss (20) with in-batch negatives as the contrastive loss.

In Table 5.2, we provide a clear summary of what are the anchors, positives and negatives for the two versions of LA(SER)$^3$. The formulation and augmented view construction methods provide a clear distinction from previous methods such as SimCSE, providing lexical difference in augmented texts and augmentation specifically targeting length robustness.

| | Anchor | Positive | Negatives |
|---|---|---|---|
| LA(SER)$^3_{\text{self-ref}}$ | a sentence or a document. | the elongated version of this sentence or document. | in-batch negatives from InfoNCE loss. |
| LA(SER)$^3_{\text{intra-ref}}$ | the elongated version of the first or a random sentence from a document | the rest of sentences of this document excluding the anchor sentence. | in-batch negatives from InfoNCE. |

Table 5.2: LA(SER)$_3$ overview table.

## 5.4 Experiments

| Training Setting→ | | Trained on wiki Self-reference | | | Trained on MSMARCO Self-reference | | | Trained on MSMARCO Intra-reference | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Models → Test dataset ↓ | SimCSE | LA(SER)$^3$ | Perf. Gain | SimCSE | LA(SER)$^3$ | Perf. Gain | COCO-DR (PT-unsup) | LA(SER)$^3$ | Perf. Gain |
| zero-shot setting | **trec-covid** | 0.1473 | 0.2129 | **44.52%** | 0.1467 | 0.1646 | **12.22%** | 0.2597 | 0.2511 | **-3.33%** |
| | **nfcorpus** | 0.0764 | 0.1265 | **65.54%** | 0.0796 | 0.0933 | **17.31%** | 0.1853 | 0.1508 | **-18.62%** |
| | **nq** | 0.0370 | 0.0836 | **125.88%** | 0.0302 | 0.0391 | **29.55%** | 0.0268 | 0.0405 | **51.10%** |
| | **fiqa** | 0.0288 | 0.0590 | **104.94%** | 0.0260 | 0.0435 | **67.36%** | 0.0821 | 0.1030 | **25.48%** |
| | **arguana** | 0.2277 | 0.3130 | **37.48%** | 0.2081 | 0.1961 | **-5.74%** | 0.3441 | 0.3834 | **11.42%** |
| | **webis-touche2020** | 0.0289 | 0.0483 | **66.99%** | 0.0177 | 0.0296 | **67.71%** | 0.0736 | 0.0896 | **21.73%** |
| | **quora** | 0.6743 | 0.7095 | **5.22%** | 0.6527 | 0.6515 | **-0.19%** | 0.7976 | 0.7911 | **-0.82%** |
| | **cqadupstack** | 0.0889 | 0.1279 | **43.90%** | 0.0864 | 0.1105 | **27.95%** | 0.1380 | 0.1560 | **13.06%** |
| | **dbpedia-entity** | 0.0837 | 0.1138 | **36.04%** | 0.0541 | 0.0558 | **3.03%** | 0.0924 | 0.0825 | **-10.76%** |
| | **scidocs** | 0.0259 | 0.0516 | **99.54%** | 0.0178 | 0.0309 | **73.19%** | 0.0305 | 0.0492 | **61.56%** |
| | **climate-fever** | 0.0127 | 0.0789 | **522.24%** | 0.0136 | 0.0198 | **45.11%** | 0.0652 | 0.1108 | **69.84%** |
| | **scifact** | 0.2174 | 0.3525 | **62.12%** | 0.2330 | 0.2276 | **-2.32%** | 0.4056 | 0.4076 | **0.49%** |
| | **hotpotqa** | 0.0829 | 0.1646 | **98.56%** | 0.0560 | 0.0750 | **34.07%** | 0.0383 | 0.0539 | **40.85%** |
| | **fever** | 0.0363 | 0.1001 | **175.88%** | 0.0263 | 0.0340 | **29.41%** | 0.1421 | 0.2524 | **77.60%** |
| | **average** | 0.1263 | 0.1816 | **43.78%** | 0.1177 | 0.1265 | **7.48%** | 0.1915 | **0.2087** | **8.97%** |

Table 5.3: Unsupervised Performance Trained with MiniLM-L6 Model. For self-reference settings, we compare with SimCSE (3). Notably, LA(SER)$^3_{\text{self-ref}}$ can be viewed as a plug-and-play module to SimCSE, as SimCSE takes an input itself as both the anchor and the positive pair, while LA(SER)$^3_{\text{self-ref}}$ further elongates this positive pair. For intra-referecne setting, we compare with COCO-DR (4). Notabaly, we only experiment with the unsupervised pre-training part of COCO-DR, as LA(SER)$^3_{\text{intra-ref}}$ can be viewed as a plug-and-play module to this part. We believe combining with our method for a better unsupervised pretrained checkpoint, the follow-up supervised fine-tuning in COCO-DR can further achieve better results.

**Training datasets** We conduct our experiments on two training dataset settings: 1) $\texttt{training}^{\texttt{wiki}}$ uses 1M sentences sampled from Wikipedia, in line with previous works on contrastive sentence representation learning (2; 3; 66); 2) $\texttt{training}^{\texttt{msmarco}}$ uses MSMARCO (67), which is equivalent to in-domain-only setting of the BEIR information retrieval benchmark (14).

**Evaluation datasets** The trained models are mainly evaluated on the BEIR benchmark (14), which comprises 18 datasets on 9 tasks (fact checking, duplicate question retrieval, argument retrieval, news retrieval, question-answering, tweet retrieval, bio-medical retrieval and entity retrieval). We evaluate on the 14 public zero-shot datasets from BEIR (BEIR-14). And we use STS-b (13) only as the auxiliary experiment.

The reasons why we do not follow the *de facto* practice, which mainly focuses on cherry-picking the best training setting that provides optimal performance on STS-b are as follows: Firstly, performances on STS-b do not display strong correlations with downstream tasks (68). In fact, document-level encoders that provide strong representational abilities do not necessarily provide strong performance on STS-b (69). Furthermore, recent works have already attributed the inferior predictive power of STS-b performance on downstream task performances to its narrow length range coverage (70). Therefore, we believe a strong sentence and document-level representation encoder should be evaluated beyond semantic textual similarity tasks. However, for completeness, we also provide the results of STS-b in Appendix 5.7.

**Baselines** We compare our methods in two settings, corresponding to the two versions of **LA(SER)$^3$**: 1) Self-Reference. Since we assume using the input itself as its positive pair in this setting, it is natural to compare LA(SER)$^3_{\text{self-ref}}$ to the strong baseline SimCSE (3). In the $\texttt{training}^{\texttt{wiki}}$ setting, we further compare with E-SimCSE, DiffCSE, and InfoCSE (2; 26; 66). Notably, these four baselines all have public available checkpoints trained on $\texttt{training}^{\texttt{wiki}}$.

2) Intra-Reference. The baseline method in this case is: taking a sentence (random or first) from a document as anchor, then use the remaining content of the document as its positive pair. Notably, this baseline is similar to the unsupervised pretraining part of COCO-DR (4), except COCO-DR only takes two sentences from a same document, instead of one sentence and the remaining part. Compared to the baseline, LA(SER)$^3_{\text{intra-ref}}$ further elongates the anchor sentence. In the result table, we refer to baseline of this settings as COCO-DR$_{\text{PT-unsup}}$.

**Implementation Details** We evaluate the effectiveness of our method with BERT (71) and MiniLM[2] (59). To compare to previous works, we first train a LA(SER)$^{\mathbf{3}}_{\text{self-ref}}$ on $\texttt{training}^{\texttt{wiki}}$ with BERT-base (uncased). We then conduct most of our in-depth experiments with vanilla MiniLM-L6 due to its low computational cost and established state-of-the-art potential after contrastive fine-tuning.[3]

All experiments are run with 1 epoch, a learning rate of 3e-5, a temperature $\boldsymbol{\tau}$ of 0.05, a max sequence length of 256, and a batch size of 64 unless stated otherwise. All experiments are run on Nvidia A100 80G GPUs.

Notably, previous works on contrastive sentence representation learning (2; 3; 26; 66) and even some information retrieval works such as (4) mostly use a max sequence length of 32 to 128. In order to study the effect of length, we set the max sequence length to 256, at the cost of constrained batch sizes and a bit of computational overhead. This max length selection provides large length coverage and facilitates more different elongation times, and is shown to bring gain on information retrieval tasks, justified by detailed analyses on max sequence length are in ablation analysis (§5.5).

For the selection of the anchor sentence, we take the first sentence of each document in the main experiment (we will discuss taking a random sentence instead of the first sentence in the ablation analysis in §5.5.1). For LA(SER)$^{\mathbf{3}}_{\text{self-ref}}$, we elongate the anchor sentence to serve as its positive pair; for LA(SER)$^{\mathbf{3}}_{\text{intra-ref}}$, we take the rest of the document as its positive pair, but then elongate the anchor sentence as the augmented anchor. For the selection of the elongation hyperparameter $\boldsymbol{m}$, we sample a random number for every input depending on its length and the max length of 256. For instance, if a sentence has 10 tokens excluding `[cls]`, we sample a random integer from [1,25], making sure it is not exceeding maximum length; while for a 50-token sentence, we sample from [1, 5]. We find that a random elongation times outperforms a fixed elongation times, providing generalization instead of overfitting to certain elongation pattern. We provide the effect of elongating to twice longer,

[2]We use a 6-layer version by taking every second layer. https://huggingface.co/nreimers/MiniLM-L6-H384-uncased

[3]For instance, sentence-transformers/all-MiniLM-L6-v2 is a SOTA sentence encoder fine-tuned with MiniLM-L6.

instead of random-times longer in ablation §5.5.2.

**Results** The main results are in Tables 5.1 and 5.3. In summary, LA(SER)$^{\mathbf{3}}$ achieves state-of-the-art nDCG@10 performance on the BEIR benchmark, across previous unsupervised information retrieval methods. Specifically, using only a BERT-base, it outperforms baseline methods (both BERT-base and BERT large) using the standard Wiki 1M dataset (our $\texttt{training}^{\texttt{wiki}}$ setting); Using the widely used MiniLM-L6 model, both LA(SER)$^{\mathbf{3}}{}_{\text{self-ref}}$ and LA(SER)$^{\mathbf{3}}{}_{\text{intra-ref}}$ outperforms corresponding fair baselines, across $\texttt{training}^{\texttt{wiki}}$ and $\texttt{training}^{\texttt{MSMARCO}}$.

Table 5.1 shows that our method leads to state-of-the-art average results compared to previous public available methods and checkpoints, when training on the same $\texttt{training}^{\texttt{wiki}}$ with BERT.

Our method has the exact same setting (training a vanilla BERT on the same $\texttt{training}^{\texttt{wiki}}$) with the rest of the baselines except InfoCSE, which further benefits from the training of an auxiliary network. Note that with a batch size of 64, our method already outperforms all the baselines to a large margin except InfoCSE. Since we train with a max sequence length of 256 (all baselines are either 32 or 64), we find that training with a larger batch size (128) further stabilizes our training, achieving state-of-the-art results. Moreover, we achieve state-of-the-art with only a $\text{BERT}_{\text{base}}$.

In general, we find that our performance gain is more pronounced when the length range of the dataset is large. On BERT-base experiments, large nDCG@10 performance gain is seen on NFCorpus (doc. avg. length 232.26, SimCSE: 0.1048 -> LA(SER)$^{\mathbf{3}}$: 0.1919), Scifact (doc. avg. length 213.63, SimCSE: 0.2492 -> LA(SER)$^{\mathbf{3}}$: 0.4317), Arguana (doc. avg. length 166.80, SimCSE: 0.2796 -> LA(SER)$^{\mathbf{3}}$: 0.4227). On the other hand, our performance gain is limited when documents are shorter, such DBPedia (avg. length 49.68) and Quora (avg. length 11.44).

Table 5.3 further analyzes the effect of datasets and LA(SER)$^{\mathbf{3}}$ variants with MiniLM-L6, showing a consistent improvement when used as a plug-and-play module to previous SOTA methods.

We also found that, even though MiniLM-L6 shows great representational power

if after supervised contrastive learning with high-quality document pairs (see popular Sentence Transformers checkpoint `all-MiniLM-L6-v2`), its performance largely falls short under unsupervised training settings, which we speculate to be due to that the linguistic knowledge has been more unstable after every second layer of the model is taken (from 12 layers in MiniLM-L12 to 6 layers). Under such setting, LA(SER)$^{\mathbf{3}}_{\text{intra-ref}}$ largely outperforms LA(SER)$^{\mathbf{3}}_{\text{self-ref}}$, by providing signals of more lexical differences in document pairs.

## 5.5 Ablation Analysis

In this section, we ablate two important configurations of LA(SER)$^{\mathbf{3}}$. Firstly, the usage of LA(SER)$^{\mathbf{3}}$ involves deciding which sentence in the document to use as the anchor (§ 5.5.1). Secondly, how do we maximize the utility of self-referential elongation? Is it more important for the model to know "me * m = me", or is it more important to cover a wider length range (§ 5.5.2)?

### 5.5.1 Selecting the Anchor: first or random?

If a document consists of more than one sentence, LA(SER)$^{\mathbf{3}}$ requires deciding which sentence in the document to use as the anchor. We ablate this with both LA(SER)$^{\mathbf{3}}_{\text{self-ref}}$ and LA(SER)$^{\mathbf{3}}_{\text{intra-ref}}$ on `training`$^{\texttt{msmarco}}$, because `training`$^{\texttt{wiki}}$ consists of mostly one-sentence inputs and thus is not able to do intra-ref or random sentence.

The results are in Table 5.4. In general, we observe that taking a random sentence as anchor brings certain noise. This is most corroborated by the performance drop of LA(SER)$^{\mathbf{3}}_{\text{self-ref}}$ + random sentence, compared to its SimCSE baseline. However, LA(SER)$^{\mathbf{3}}_{\text{intra-sim}}$ + random sentence seems to be able to act robustly against this noise.

We hypothesize that as LA(SER)$^{\mathbf{3}}$ provides augmented semantic signals to contrastive learning, it would be hurt by overly noisy in-batch inputs. By contrast, LA(SER)$^{\mathbf{3}}_{\text{intra-sim}}$ behaves robustly to this noise because the rest of the document apart from the anchor could serve as a stabilizer to the noise.

| Anchor Sentence | Method | Zero-shot Average | Performance Change |
|---|---|---|---|
| First | SimCSE | 0.1177 | ↑7.48% |
| | LA(SER)$^{3}_{\text{self-ref}}$ | 0.1265 | |
| Random | SimCSE | 0.1127 | ↓10.05% |
| | LA(SER)$^{3}_{\text{self-ref}}$ | 0.1013 | |
| First | COCO-DR$_{\text{pt-unsup}}$ | 0.1915 | ↑ 8.97% |
| | LA(SER)$^{3}_{\text{intra-ref}}$ | **0.2087** | |
| Random | COCO-DR$_{\text{pt-unsup}}$ | 0.1930 | **↑ 5.33%** |
| | LA(SER)$^{3}_{\text{intra-ref}}$ | 0.2033 | |

Table 5.4: Taking First sentence or Random sentence as the anchor? - ablated with MiniLM-L6 on `training`$^{\texttt{msmarco}}$.

### 5.5.2 Importance of Self-referential Elongation

With the validated performance gain produced by the framework, we decompose the inner-workings by looking at the most important component, elongation. A natural question is: is the performance gain only brought by coverage of larger trained length range? Or does it mostly rely on the semantic signal that, "my-longer-self" still means myself?

| Elongation Mode | Max Seq. Length | Zero-shot Average |
|---|---|---|
| None | 256 | 0.1263 |
| Twice | 256 | 0.1523 |
| Random | 64 | 0.1778 |
| | 128 | 0.1764 |
| | 256 | **0.1816** |

Table 5.5: 1) Elongdating to fixed-times longer or a random time? 2) Do length range coverage matter? - ablated with MiniLM-L6 on `training`$^{\texttt{wiki}}$.

Table 5.5 shows that, elongating to random-times longer outperforms elongating to a fixed two-times longer. We hypothesize that, a fixed augmentation introduces certain overfitting, preventing the models to extrapolate the semantic signal that "elongated me = me". On the other hand, as long as they learn to extrapolate this signal (by * random times), increasing max sequence length provides decreasing marginal benefits.

## 5.6 Auxiliary Property Analysis

### 5.6.1 Positional Invariance

Recalling in Observation 2 and PoC experiment 2, we focused on analyzing the effect of elongation attack on intra-sentence similarity, which is already high after CL (28). Therefore, we have imposed the absence of positional embeddings with the aim to simplify the derivation in proving that, with elongation, dominant tokens receive higher attention gains than regular tokens. Here, we present the validity of this assumption by showing models' greatly reduced sensitivity towards positions after contrastive learning.

We analyze the positional (in)sensitivity of 4 models (MiniLM (59) and mpnet (58) respectively before and after contrastive learning on Sentence Embedding Training Data[4]). Models after contrastive learning are Sentence Transformers (18) models `all-mpnet-base-v2` and `all-MiniLM-L12-v2`.

We take the sentence pairs from STS-b test set as the inference set, and compute each model's perceived cosine similarity on the sentence pairs (distribution 1). We then randomly shuffle the word orders of all sentence 1s in the sentence pairs, and compute each model's perceived cosine similarity with sentence 2s again (distribution 2).

The divergence of the two distributions for each model can serve as a proxy indicator of the model's sensitivity towards word order, and thus towards positional shift. The lower the divergence, the more insensitive that a model is about positions.

We find that the Jenson Shannon divergence yielded by MiniLM has gone from 0.766 (vanilla) to 0.258 (after contrastive learning). And the same for mpnet goes from 0.819 (vanilla) to 0.302 (after contrastive learning). This finding shows that contrastive learning has largely removed the contribution of positions towards document embeddings, even in the most extreme case (with random shuffled word orders). This has made contrastively-learned models acting more like bag-of-words models, aligning with what was previously found in vision-language models (65).

[4] `https://huggingface.co/datasets/sentence-transformers/embedding-training-data`

Moreover, MiniLM uses absolute positional embeddings while mpnet further applies relative positional embeddings. We believe that the positional insensitivity pattern holds for both models can partly make the pattern and **LA(SER)$^3$**'s utility more universal, especially when document encoders are trained with backbone models that have different positional encoding methods.

## 5.7 Results of STS-b

| **Method** | **Max Seq.** | **sts-b** |
|---|---|---|
| BERT-whitening | - | 68.19/71.34 |
| BERT-flow | 64 | 58.56/70.72 |
| SimCSE | 32 | 76.85 |
| LA(SER)$^3$-mean | 256 | 75.61 |
| LA(SER)$^3$-`[cls]` | 256 | 76.19 |

Table 5.6: STS-b test set results, compared with unsupervised sentence representation methods. SimCSE and LA(SER)$^3$ are trained on the same $\texttt{training}^{\texttt{wiki}}$. The two numbers of BERT-whitening and BERT-flow correspond to optimizing on NLI or target data (sts-b). Results are from the original works (3; 5; 6).

In this section, we present the results of STS-b test set (Table 5.6). As discussed in the main sections, we position that STS-b is not correlated with downstream semantic tasks performance (68; 69), and effectiveness of document-level representation encoders should be evaluated beyond this task. The inferior predictability of STS-B on downstream task performances have been attributed to length ranges (70). We hypothesize that, training with a large max sequence length increases the uncertainty of elongation hyperparameter $\boldsymbol{m}$ of LA(SER)$^3$, resulting in a diverse length range, and less corresponding concrete examples at each length.

We show that, while out-performing SimCSE by a large margin on other downstream semantic tasks (Main Section, Table 5.1), our long sequence length poses a certain level of instability in converging, showing a small performance drop on shorter sentences (STS-b). The converging instability is further confirmed by training an extra LA(SER)$^3$ with `[cls]`-pooling, as `[cls]`-pooling is faster in converging - as it involves only optimizing one token. Notably, SimCSE also uses `[cls]`-pooling. Therefore, we roughly stay on-par with SimCSE on encoding shorter documents,

while out-performing it by a large margin on other downstream tasks.

## 5.8 Limitations

We position that the focus of our work lies more in analyzing theoretical properties and inner-workings, and thus mostly focus on unsupervised contrastive learning settings due to compute constraints. However, we believe that with a better unsupervised checkpoint, further supervised fine-tuning will yield better results with robust patterns. We leave this line of exploration for future work. We also acknowledge the synthetic nature of the copy-and-concatenate attack. Further, we only focus on bi-encoder dense retrievers. In information retrieval, there are other methods involving using cross-encoders to conduct re-ranking, and sparse retrieval methods. Although our method can be used as a plug-and-play module to many of these methods, it is hard to exhaust testing with every method. We thus experiment the plug-and-play setting with a few representative methods. We hope that future works could evaluate the effectiveness of our method combining with other lines of baseline methods such as cross-encoder re-ranking methods or MLLM-based retrievers.

## 5.9 Conclusion

In this chapter, we questioned the length generalizability of contrastive learning-based text encoders. We observed that, despite their seemingly strong representational power, this ability is strongly vulnerable to length-induced semantic shifts. we formalized length attack, demystified it, and defended against it with LA(SER)$^{\mathbf{3}}$. We found that, teaching the models "my longer-self = myself" provides a standalone semantic signal for more robust and powerful unsupervised representation learning.

From the chapter, we also see how vulnerabilities of embedding models can be connected and explained by fundamental properties we explored in Chapter 4. In the case of this chapter, length vulnerability is relevant to the length-sensitive isotropy and further intensified intra-document similarity. The study on length also opens a new angle for analysis. We will see how this angle can provide valuable design

concern for IR corpus (seen in Chapter 6).

# CHAPTER 6

# Reasoning as Retrieval

In the past few chapters, we have built a solid understanding of mechanism of contrastive learning, training and evaluation settings of information retrieval. Entering the large language model era, new challenges emerge, prompting reconsideration of retrievers' capability requirements and their role in the interaction with LLMs in new paradigm such as RAG. Meanwhile, we have started to see the field of embedding models going from developing in orthogonal directions with LLMs, to starting to leverage LLMs' strengths, such as switching from training embedding models using encoder backbones to using LLMs as the backbone.

However, even in advanced paradigms like RAG, the expectation on embedding models' capabilities is typically low. A hard query is typically rewritten or decomposed into basic queries suitable for traditional embedding models. This poses a gap at the intersection between LLMs themselves and the advancements of embedding models that leverage LLM backbones. The field lacks a clear understanding of embedding models' reasoning capability, the important capability instrumental in understanding how the field should use embedding models in the LLM era.

Under the above context, in this chapter, we provide the first systematic evaluation of reasoning capabilities of embedding models. Previous reasoning benchmarks

solely focus on the LLM, while previous embedding models are mostly evaluated on semantic benchmarks like STS and BEIR. We introduce RAR-b, *reasoning as retrieval benchmark*, and the corresponding methodological framework that reconceptualizes reasoning as a retrieval problem for holistically evaluating the reasoning capabilities of embedding models. The core research questions we ask are, how much reasoning is encoded in embeddings, and how this differs for different types of embedding models? RAR-b provides an upper-bound assessment of embedding models' reasoning capabilities by not using embedding models to assist LLMs, but instead using the embedding models themselves to solve reasoning problems.

Semantic textual similarity (STS) and information retrieval tasks (IR) tasks have been the two major avenues to record the progress of embedding models in the past few years. Under the emerging Retrieval-augmented Generation (RAG) paradigm, we envision the need to evaluate next-level language understanding abilities of embedding models, and take a conscious look at the reasoning abilities stored in them. Addressing this, we pose the question: **Can retrievers solve reasoning problems?** By transforming reasoning tasks into retrieval tasks, we find that without specifically trained for reasoning-level language understanding, current state-of-the-art retriever models may still be far from being competent for playing the role of assisting LLMs, especially in reasoning-intensive tasks. Moreover, albeit trained to be aware of instructions, instruction-aware IR models are often better off without instructions in inference time for reasoning tasks, posing an overlooked retriever-LLM behavioral gap for the research community to align. However, recent decoder-based embedding models show great promise in narrowing the gap, highlighting the pathway for embedding models to achieve reasoning-level language understanding. We also show that, although current off-the-shelf re-ranker models fail on these tasks, injecting reasoning abilities into them through fine-tuning still appears easier than doing so to bi-encoders, and we are able to achieve state-of-the-art performance across all tasks by fine-tuning a reranking model. We release Reasoning as Retrieval Benchmark (RAR-b), a holistic suite of tasks and settings to evaluate the reasoning abilities stored in retriever models. **RAR-b** is available at `https://github.com/gowitheflow-1998/RAR-b`.

## 6.1 Introduction

Semantic textual similarity (STS) and information retrieval tasks (IR) have been two principle measures to record the progress of dense representation models (10; 11; 12; 13; 14; 72). Despite still heavily being evaluated in sentence representation research, STS is known for its limited alignment with real-world use cases (73; 74), ambiguity (75), and performance orthogonality with IR and other downstream tasks (31; 68; 69).

In the LLM era, Retrieval-augmented Generation (RAG) (32; 73; 76; 77) has become a go-to alternative method to vanilla end-to-end generative language models (78; 79). This shift is motivated by the inherent weaknesses of LLMs towards factual errors, due to hallucinations (80), knowledge outdatedness (81), rarity in long-tailed knowledge (82; 83), and reasoning failure such as on logical deduction (84).

Retrieval-Augmented Generation (RAG) is employed differently across various NLP tasks:

- For knowledge-intensive tasks, RAG is employed to retrieve the most up-to-date and reliable knowledge references (81; 83), serving as new prompts for LLMs to extract information and formulate responses. This method mitigates models' natural tendencies to hallucinate (80) and reduces the need for frequent fine-tuning of LLMs.

- In reasoning-dependent tasks, RAG aims to fetch the most relevant chunks from extensive inputs to guide the focus of LLMs, e.g., in multi-hop question answering scenarios where reasoning across chunks from multiple documents is required. Reasoning with such long context is not only impossible for LLMs with short context windows, but also challenging for LLMs with long context capabilities (76).

Despite the promise shown by dense retrievers in fetching references for knowledge-intensive tasks, these systems still fall short in retrieving reliable and cite-worthy references (83), compared to state-of-the-art proprietary LLMs (78) in a standalone manner, highlighting the undesirable behavior of retrievers in assisting LLMs (85; 86).

This discrepancy is more pronounced in reasoning-intensive tasks, where retrieval-augmented generation methods present inconsistent gains, or even performance degradation to LLMs (76; 87).

With the complexities of the role that dense representation models play in the LLM era, the need for accurately assessing their advanced language understanding abilities becomes crucial. We advocate for evaluating these models' capabilities beyond mere factual recall or semantic matching, focusing on their proficiency in complex thought processes and logical deduction.

This paper introduces the Reasoning as Retrieval Benchmark (RAR-b), a novel framework that reframes reasoning tasks as retrieval tasks, offering a comprehensive reevaluation of **"the actual reasoning representation compressed"**[1] in dense representation models, and challenges the status quo of reasoning-oriented RAG.

Historically, lexical-based retrieval methods have long been seen as baselines of reasoning tasks (88; 89; 90), and often proved insufficient (91). For instance, the ARC challenge (91) itself is constructed by filtering the examples where retrieval systems fail. With the advancements of dense retrieval systems and the increased adoption of RAG, RAR-b emerges as a timely and crucial framework, prompting the essential question: Can dense representation models effectively encode and utilize the reasoning abilities necessary for complex language understanding?

RAR-b's contributions are three-fold:

- With the established consensus of the semantic similarity understanding ability (13) and topical alignment ability (14) possessed by sentence and document representation models, RAR-b takes a step further and envisions their possession of **reasoning abilities**, and calls for evaluating them today.

- We extensively evaluate state-of-the-art open-source and proprietary retrievers and re-rankers, covering unsupervised, supervised, instruction-aware ones, providing an in-depth understanding of reasoning behaviors of current retrieval models.

[1] we use 🗜️ "RAR" to convey the connotation of "compression"

- We reveal key insights through comparing behaviors yielded by systems with different architectures and training recipes, highlighting pathways for embedding models to achieve reasoning-level language understanding and directing future research in the field.

## 6.2 RAR-b

In this chapter, we construct and release **RAR-b**: Reasoning as Retrieval Benchmark. Deviating from evaluating on reasoning tasks with a full RAG pipeline (retriever+LLM), we instead focus on evaluating only the retrievers. By constructing reasoning tasks into retrieval tasks, we investigate how good retrievers are on solving them in a standalone manner, and use this as a proxy of the upperbound of retrievers' capabilities in assisting LLMs, in a standard RAG system.

We design three levels of tasks, resulting in the integration of 12 tasks derived from 17 datasets. We convert the original datasets into retrieval formats with both multiple-choice retrieval and full-dataset retrieval settings. We first benchmark the performance of state-of-the-art bi-encoder models,

spanning across three model categories: unsupervised dense retrieval models, supervised dense retrieval models, and instruction-aware dense retrieval models. We evaluate both representative open-source and proprietary models such as Cohere and OpenAI Embedding Models. We further benchmark the performance of representative re-ranking models, both on using them to solve multiple-choice retrieval setting independently, and on further re-ranking the documents retrieved by bi-encoders in the Full-dataset retrieval setting.

### 6.2.1 Problem Formulation

We propose the novel framework of evaluating reasoning tasks as retrieval tasks, assessing whether representations of reasoning expressions are semantically well-encoded by retrievers. In other words, we assess whether the groundtruth answer to a reasoning problem is encoded to have the highest similarity with the query in the semantic space. Given a retriever $\boldsymbol{R}$, and a query $\boldsymbol{q} \in \boldsymbol{Q}$, and a groundtruth answer

$\boldsymbol{g}$ hidden in a corpus $\boldsymbol{\mathcal{C}}$ that is intentionally made very large, the most ideal scenario would be:

$$\arg\max_{d \in C} \boldsymbol{S}(\boldsymbol{R}(\boldsymbol{q}), \boldsymbol{R}(\boldsymbol{d})) = \boldsymbol{g} \tag{6.1}$$

However, this is typically not possible given the complexity of reasoning language and the fact that current sentence encoders are typically not yet well-trained to model such expressions. Therefore, we in turn are interested in whether:

$$\boldsymbol{g} \in \arg\max\nolimits^{(n)}_{d \in C} \boldsymbol{S}(\boldsymbol{R}(\boldsymbol{q}), \boldsymbol{R}(\boldsymbol{d})) \tag{6.2}$$

For the **Full-dataset retrieval** setting, we can quantitatively measure $\boldsymbol{R}$'s performance with commonly-adopted information retrieval metrics such as nDCG@n and recall@n, respectively concerning how well the top-n retrieved answers are ranked (nDCG@n), and if the correct answers are even retrieved as the top n (recall@n). Specifically, we use nDCG@10 and recall@10 for all tasks.

For the **Multiple-choice Retrieval** setting, we simply assess whether the query is perceived by the retrievers to be more similar to the groundtruth answer, than to other candidate answers in its original dataset format. Models' performance in this setting can be easily quantified by accuracy.

We evaluate both **FULL** and **MCR** settings whenever possible. As will be introduced in the next section, we are able to evaluate 11 tasks in the FULL setting (all tasks except C-STS) and 9 tasks in the MCR setting (tasks that are originally multiple–choice problem).

### 6.2.2 Datasets

**Dataset and Processing** The datasets we consider can all be seen as reasoning problems. Firstly, we include commonsense reasoning datasets, $\boldsymbol{\alpha}$NLI (92), HellaSwag (93), PIQA (94), SocialiQA (95), ARC-Challenge (91), Quail (89), Winogrande (90), and C-STS (75). For all datasets in the full retrieval settings, we pool all the candidate answers across all splits to serve as the corpus, aligning with the setting in most retrieval benchmarks, and evaluate them using test set questions as the queries. All commonsense reasoning datasets are suitable to test in Full-dataset retrieval

setting, except for C-STS (75) due to the potential sparse annotation problem when constructing it into Full-dataset retrieval settings (14) (detailed explanation in Appendix 6.7).

Apart from commonsense reasoning abilities, we further envision the possession of temporal, spatial, numerical, and symbolic reasoning abilities in dense retrievers. Temporal and spatial reasoning abilities are evaluated respectively with TempReason (96) and SpartQA (97). For TempReason, we evaluate each of its sub-level tasks separately, and construct the pure, fact and context settings to assess different aspects of retrievers' behaviors (detailed analyses in Section 6.4.1). For numerical reasoning abilities, we concatenate MATH (98) and GSM8K (99), two datasets commonly used to evaluate LLMs, using questions as queries, the pool of all answers as the corpus. Because of their small scales, we enlarge the corpus with MetaMathQA (100), which is created synthetically with LLMs, using the training set of MATH. Therefore, it is assured that no examples in MetaMathQA can act as the groundtruth answer for any of our evaluated queries (which are from the test set of GSM8K and MATH).

With the common user cases of code retrieval and that code understanding can serve as a proxy of the understanding of symbolic language (101), we further include code retrieval tasks. We concatenate HumanEvalPack (102) and MBPP (103) to form the evaluation queries, because of their validated quality, ubiquitously seen in the evaluation of LLMs. Notably, HumanEvalPack (102) is an extended version of HumanEval (104), by translating the original Python split to cover Javascript, Java, Go, C++, and Rust. To enlarge the corpus, we further sample 200k code from CodeSearchNet (105) and 100k answers from a synthetic dataset TinyCode[2] to cover pure code text and the mixture of natural language and code. We extensively explore the optimal setting to construct the code corpus and evaluate the code tasks (see analyses in Section 6.4.2) to make it as difficult as possible while keeping the evaluation computationally efficient. For HumanevalPack, we evaluate the code continuation setting; when enlarging the corpus, we sample examples from CodeSearchNet that fall under similar length range of the groudtruth documents to

[2] https://huggingface.co/datasets/nampdn-ai/tiny-codes

| Split (→) | | | | Train | Dev | Test | | Avg. #Words | |
|---|---|---|---|---|---|---|---|---|---|
| Task (↓) | Domain (↓) | Dataset (↓) | #MCR | #Pairs | #Query | #Query | #Corpus♣ | Query | Document |
| Commonsense | General | $\alpha$NLI (92) | 2 | 169654 | 0 (♠) | 1532 | 241347 | 19.35 | 8.30 |
| Commonsense | General | HellaSwag (93) | 2 | 39905 | 0 (♠) | 10042 | 199162 | 40.12 | 24.71 |
| Commonsense | Physical | PIQA (94) | 2 | 16113 | 0 (♠) | 1838 | 35542 | 7.08 | 18.90 |
| Commonsense | Social | SIQA (95) | 3 | 33410 | 0 (♠) | 1954 | 71276 | 22.28 | 4.39 |
| Commonsense | Multiple | Quail (89) | 4 | 10246 | 0 (†) | 2720 | 32787 | 345.32 | 4.98 |
| Commonsense | Scientific | ARC-C (91) | 4 | 1119 | 299 | 1172 | 9350 | 22.65 | 5.29 |
| Commonsense | General | WinoGrande (90) | 2 | 40398 | 0 (♠) | 1267 | 5095 | 20.11 | 1.22 |
| Commonsense | General | c-STS (75) | 2 | | | | —★ | | |
| Temporal | General | TempReason (96) | — | | | | | | |
| Temporal | General | - TR1 | — | 400000 | 4000 | 4000 | 12504 | 11.56 | 2.00 |
| Temporal | General | - TR2 | — | 16017 | 5521 | 5397 | 15787 | 10.47‡ | 2.91 |
| Temporal | General | - TR3 | — | 13014 | 4437 | 4426 | 15664 | 12.20‡ | 2.91 |
| Spatial | General | SpartQA (97) | 3 | 22749 | 3579 | 3594 | 1592 | 125.67 | 10.09 |
| Numerical | Math | + MATH (98) | — | 7500 | 0 | + 5000 | + 5000 | 31.65 | 84.83 |
| Numerical | Math | + GSM8K (99) | — | 7473 | 0 | + 1319 | +1319 | 47.25 | 52.78 |
| Numerical | Math | + MetaMathQA (100) ⋆ | — | — | — | — | + 383057 | — | 102.39 |
| Numerical | Math | = math-pooled | — | 14973 | 0 | = 6319 | = 389376 | 34.91 | 101.99 |
| Symbolic | Code | + HumanevalPack (102) | — | 0 | 0 | + 984 | + 984 | — | — |
| Symbolic | Code | + MBPP (103) | — | 0 | 0 | + 500 | + 500 | — | — |
| Symbolic | Code | + CodeSearchNet (105) ⋆ | — | — | — | — | + 200000 ⋆ | — | — |
| Symbolic | Code | + TinyCode ⋆ | — | — | — | — | + 100000 ⋆ | — | — |
| Symbolic | Code | = Code-pooled | — | 0 | 0 | = 1484 | = 301484 | — | — |

Table 6.1: **Statistics of datasets** of RAR-b. ♠: We use the original dev set as the test set, and add the original test set candidates to the corpus if available, as the original test set labels of these datasets are not designed to be publicly available. †: We concatenate the validation and "challenge" set as the test set, leaving no dev set. ♣: We pool the unique set of candidates across all splits as the corpus where available, i.e., corpus is shared across train, dev, and test set. ★: c-STS is not suitable for full-dataset retrieval setting, which is because of the effect of sparse annotation problem if doing so. ‡: For TR2 and TR3, we construct Pure, Fact, and Context Setting, where the average query lengths are {10.47, 145.14, 1901.19}, {12.20, 157.07, 2132.85}. Notably, most open-source models are not able to process the context setting at once without loss of information. ⋆: The original MetaMathQA was actually a training set generated from the training set of MATH and GSM8K, but we only use its unique answer set as the corpus, so we do not include non-used statistics here. The same goes to CodeSearchNet and TinyCode in code retrieval, where we respectively sample 200k and 100K to enlarge the code corpus.

make the retrieval more difficult, given that length information is typically encoded in the embeddings (31) and we don't want the models to leverage this information to simplify the retrieval.

**Task Levels** We group the datasets into three levels.

Level-1 dataset tasks are by their nature more in-distribution to typical datasets used to train IR models, and makes more sense to be solved even without instructions. For instance, PIQA consists of physical goals and possible solutions, which are similar to IR training sets that are question-answer pairs. HellaSwag and $\alpha$NLI respectively look for the end of an unfinished story, and the middle of a story given the start

and the ending, which are both extremely similar to the positive pairs constructed in training of unsupervised dense retrieval models such as Contriever (29), which samples overlapping spans of the text as positive pairs.

Level-1.5 datasets on the other hand, are more out-of-distribution in terms of task categories, and make less sense to be solved without a **clearly-stated purpose**. For instance, WinoGrande asks to find the correct answer to fill in an unfinished sentence, which is marked by an underscore. We believe this format is not commonly seen in IR training, and as will be shown, current models struggle to develop a nuanced understanding about this task. We further include a novel task, conditioned-sts (CSTS) (75), which concerns different semantic similarities between the same sentence pairs, under different conditions. Both of the tasks make less sense to solve without prepending the query with an instruction, unless the models has seen the format in their training. We position CSTS as the most challenging dataset on this level, as the task is, to the best of our knowledge, not similar to any datasets that would be used in the training of IR models. We structure C-STS in two settings: CSTS-Easy and CSTS-Hard (Appendix 6.6).

Level-2 datasets include **{**temporal, numerical, spatial, symbolic**}** reasoning, serving as an extra lens to inspect the abilities that researchers and practitioners do not expect the current generation retrievers to have (with the exception of OpenAI's embedding model, which is trained on a large portion of code data (73) from its earliest versions). These abilities are rarely assessed or considered to be retrievers' necessary capabilities. But we instead envisage that representation models have both a necessity and the capacity to attain these abilities. The reasons are as follows:

Consider the following three examples that might occur as queries to a RAG system. These types of requirements are rarely properly evaluated in current IR tasks and benchmarks. In "retrieve all records where total sales $>$ *10*", retrievers are required to have **numerical understanding**; in "what are the main arguments in *yesterday's* report?", they need to use their **temporal perception**; in "what are popular tourist attractions of the state *next to Florida*?", the retrievers need to have **spatial knowledge** or **world knowledge**. We see that these abilities are not stored in current SOTA representation models, and thus relevant problems are at the

moment typically solved using external pipelines or methods (106). For one, one can use an extra agent to help filter the records $>$ 10 in a database system, using meta data, instead of relying this part on retrievers (the RAG pipeline thus complicates to agent - retriever - LLM). On the other hand, one can solve these problems by query rewrites (107) (e.g., chain-of-thoughts retrieval (108) or inductive retrieval (109)). For instance, one can first use an LLM to understand that the states next to Florida are Georgia and Alabama, and rephrase the query to "what are popular tourist attractions of Georgia and Alabama?". In this case, the pipeline is also complicated into LLM - retriever - LLM. However, information about time, numbers and space is undoubtedly encode-able, if certain dimensions of the representations are allocated to do so. Out of this reason, we do not think that complicated pipelines, such as ones that go through LLM - retriever - LLM, are eventually necessary. Instead, retrievers can take on more responsibility before LLM in a typical RAG pipeline, if not end-to-end (110). In conclusion, we call for the evaluation of level-2 tasks today, as a checkpoint of future essential capabilities of retrieval models.

For all datasets, we evaluate on their test sets, and dev/val sets when test set groundtruth labels are not publicly available, leaving room for the research community to investigate leveraging their training sets in the development of future retrievers with reasoning capabilities.

### 6.2.3 Models

We first benchmark state-of-the-art bi-encoders, spanning unsupervised dense retrieval models (U-DR), supervised dense retrieval models (S-DR), supervised instruction-aware dense retrieval models (I-DR). For U-DR models, we have Contriever (29). For S-DR models, we include two Sentence Transformers (18) models that have been the most popular in the last few years, `all-mpnet-base-v2` and `all-MiniLM-L6-v2`; Dragon+ (111), an S-DR model that is progressively distilled with different views of other SOTA models, and BGE-M3 (112). For I-DR models, we include Instructor (33), BGE (v1.5) (113), and consider all of their model sizes (base, large, XL for Instructor and small, base, large for BGE) to understand the scaling laws (if there's any) between model size and reasoning abilities. We further include E5-Mistral (35),

a latest state-of-the-art instruction-aware embedding model that is finetuned from Mistral (114) with {instruction, query, answer} pairs distilled from GPT-4. We recognize the recent advancements in unifying generative and representation abilities into one model, and benchmark GritLM (36), whose data and training process of the embedding abilities are similar to (114), but differing in its full parameters finetuning and joint learning with the generative objective. Lastly, we benchmark popular proprietary models Cohere `Embed-English-v3.0` and OpenAI `text-embedding-ada-002`, `text-embedding-3-small`, `text-embedding-3-large` through API encoding.

For rerankers (cross-encoders), we benchmark a few representative models such as BGE-reranker-large (113), TART-Full (34) and Cohere `rerank-english-v2.0`.

To summarize, we include unsupervised/supervised and instruction-aware dense retrieval models, and reranker models. As we focus on the relationship between representation and reasoning in this chapter, we do not include sparse or hybrid retrievers which are out of score for this chapter.

### 6.2.4 Reasoning as Retrieval

Instead of solving these reasoning tasks via text-to-text generation (115; 116), we explore the possibilities of solving them as a unified representation-to-representation matching problem. That is, all tasks, contexts, and possible answers are all understood through a shared representation space, and the search of the answer, given a task and the context information, falls back to be a simple similarity match problem.

We explore two settings: $\text{RAR}_{\text{w/o Instructions}}$ and $\text{RAR}_{\text{w/ Instructions}}$. For $\text{RAR}_{\text{w/o Instructions}}$, we simply use the original question/query/context in the datasets as the query, without describing the task with an instruction prompt, and use the pool of all possible choices across splits as the candidate documents. For $\text{RAR}_{\text{w/ Instructions}}$, we prepend the query with an instruction that describes the task. For instance, for $\boldsymbol{\alpha}$NLI, an (imperfect) instruction can be "Given the following start and end of a story, retrieve a possible reason that leads to the end".

We construct two task settings: Multiple-choice Retrieval Setting (MCR), and Full-dataset Retrieval Setting (Full). For multiple choice retrieval, we utilize the choices available to each question in the original dataset. For this setting, the performance

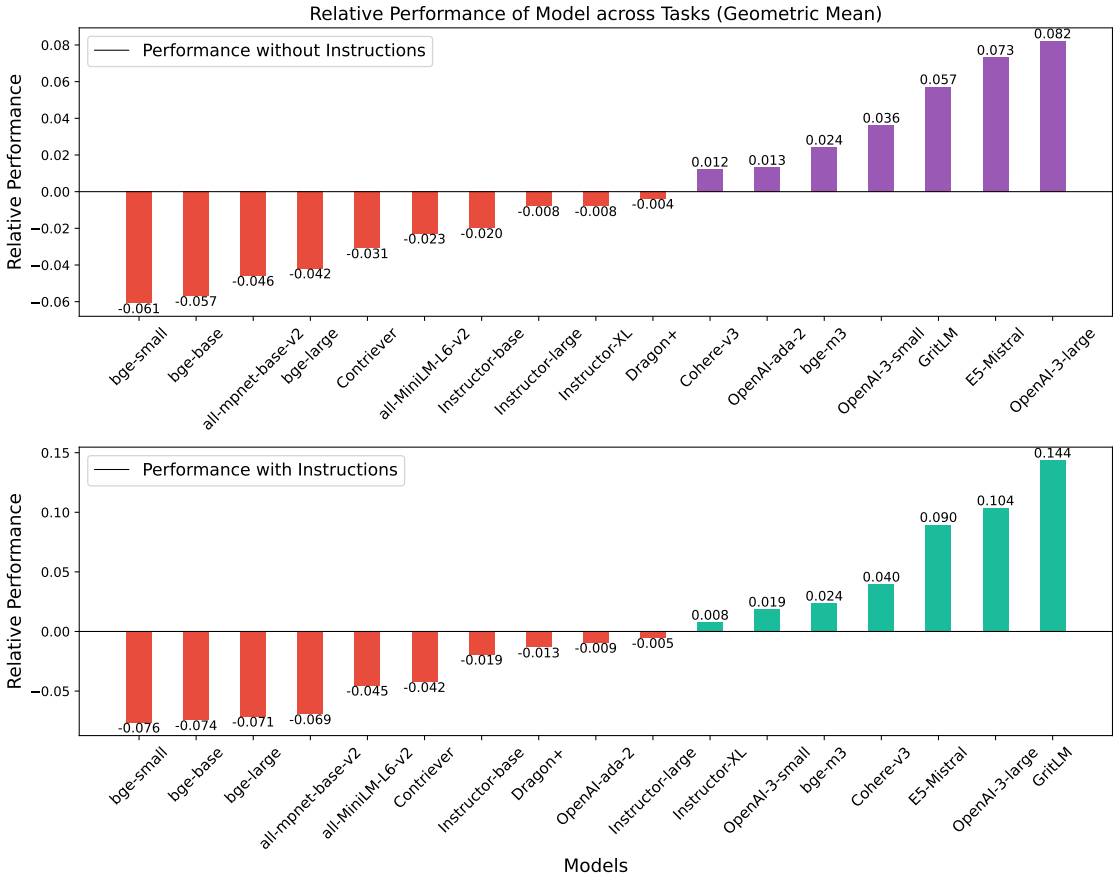


Figure 6.1: Relative Performance of all models on Full-dataset Retrieval setting. We take the geometric mean across tasks to represent each model's performance; and we subtract the mean performance across models from each model's performance to understand their relative performance.

can be easily measured by accuracy, as each input has only 2-4 candidates. For Full-dataset Retrieval Setting, we construct a pool of all candidate choices with all the unique candidates from the same dataset, and investigate whether the groundtruth answer can be retrieved from a candidate pool of a much larger order of magnitude. In line with typical IR benchmarks, the corpus is constructed with candidates from all splits, and it is only shared across them in inference. For Full-dataset Retrieval, we measure the performance with commonly-adopted information retrieval metrics, including nDCG@n and Recall@n.

The multiple-choice setting and the full retrieval setting serve as simulacra of real-life RAG scenarios. Full retrieval performance assesses whether correct answers can be retrieved as top candidates, providing a basis for the later LLMs to make informed decisions. Multiple choice performance gauges whether hard-negative

candidates might be erroneously prioritized over the groundtruth answer, introducing noise into the references presented to LLMs. From a pure retrieval perspective on the other hand, multiple-choice setting evaluates the extent to which a dual dense encoder can substitute a cross-encoder in reasoning-intensive late interactions - if dense encoders are already good at MCR, no re-ranking models (cross-encoders) are needed to rerank the top candidates retrieved by dense retrievers, before them being presented to LLMs. As a comparison, we also benchmark the performance of representative re-rankers, both as standalone solves in the MCR setting, and as re-rankers to re-rank candidates retrieved by bi-encoders in the Full setting.

For Multiple-Choice Retrieval, we evaluate both retrievers and re-rankers, on the datasets that are originally multiple-choice problems ($\boldsymbol{\alpha}$NLI, HellaSwag, PIQA, SIQA, WinoGrande), and ones that can be constructed to a MCR problem (C-STS). For Full-Dataset Retrieval, we evaluate both performance of using only retrievers, and retrieval+reranking performance as in a typical retrieval pipeline. We evaluate the FULL setting for every dataset ($\boldsymbol{\alpha}$NLI, HellaSwag, PIQA, SIQA, WinoGrande, TempReason, SpartQA, Math-pooled, Code-pooled) except C-STS, which is not suitable for full retrieval due to potential sparse annotation effect - its original correct answer might not be the most suitable one throughout the corpus.

## 6.3 Results

In this section, we present detailed results of full-dataset retrieval performance and multiple-choice retrieval performance, where we find i) Embedding models that perform similarly strong on STS tasks and traditional information retrieval task like BEIR, perform largely different on RAR-b tasks, with some encoder-based models presenting near-chance performance on the MCR; (ii) Decoder-based and instruction-tuned embeddings generally perform better, showing the strong generalization of decoder models and positioning conducting contrastive learning on LLMs as an alignment of representation capabilities to their generative capabilities. (iii) Instructions sometimes hurt non-instruction-tuned models, showing the OOD nature of instructions and the difficulty of understanding instruction semantics.

### 6.3.1 Full-dataset Retrieval Setting

| Model ↓ Dataset → | | αNLI | HellaSwag | PIQA | SIQA | Quail | ARC-C | WinoG | TR | SpartQA | Math | Code | Geo. Mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Contriever | w/o Inst. | 0.318 | 0.144 | 0.246 | 0.013 | 0.050 | 0.086 | 0.471 | 0.108 | 0.109 | 0.308 | 0.093 | 0.123 |
| | w/ Inst. | 0.271 | 0.177 | 0.217 | 0.009 | 0.049 | 0.076 | 0.263 | 0.105 | 0.106 | 0.218 | 0.071 | 0.104 |
| all-mpnet-base-v2 | w/o Inst. | 0.224 | 0.263 | 0.290 | 0.024 | 0.034 | 0.118 | 0.207 | 0.058 | 0.002 | 0.718 | 0.531 | 0.108 |
| | w/ Inst. | 0.020 | 0.130 | 0.272 | 0.013 | 0.030 | 0.104 | 0.097 | 0.044 | 0.010 | 0.692 | 0.488 | 0.077 |
| all-MiniLM-L6-v2 | w/o Inst. | 0.282 | 0.242 | 0.253 | 0.016 | 0.039 | 0.095 | 0.473 | 0.080 | 0.017 | 0.682 | 0.440 | 0.131 |
| | w/ Inst. | 0.151 | 0.205 | 0.247 | 0.015 | 0.035 | 0.094 | 0.207 | 0.076 | 0.006 | 0.624 | 0.423 | 0.101 |
| Dragon+ | w/o Inst. | 0.321 | 0.277 | 0.280 | 0.020 | 0.041 | 0.089 | 0.672 | 0.087 | 0.103 | 0.451 | 0.176 | 0.150 |
| | w/ Inst. | 0.252 | 0.241 | 0.264 | 0.017 | 0.042 | 0.082 | 0.609 | 0.081 | 0.108 | 0.362 | 0.128 | 0.133 |
| bge-m3 | w/o Inst. | 0.247 | 0.257 | 0.229 | 0.049 | 0.075 | 0.090 | 0.417 | 0.140 | 0.075 | 0.692 | 0.388 | 0.178 |
| | w/ Inst. | 0.247 | 0.255 | 0.190 | 0.048 | 0.071 | 0.090 | 0.353 | 0.152 | 0.070 | 0.645 | 0.396 | 0.170 |
| Instructor-base | w/o Inst. | 0.246 | 0.263 | 0.282 | 0.024 | 0.036 | 0.102 | 0.301 | 0.059 | 0.036 | 0.584 | 0.415 | 0.134 |
| | w/ Inst. | 0.201 | 0.239 | 0.254 | 0.023 | 0.038 | 0.096 | 0.164 | 0.052 | 0.069 | 0.582 | 0.401 | 0.127 |
| Instructor-large | w/o Inst. | 0.248 | 0.295 | 0.332 | 0.037 | 0.051 | 0.125 | 0.264 | 0.068 | 0.016 | 0.710 | 0.546 | 0.146 |
| | w/ Inst. | 0.234 | 0.266 | 0.286 | 0.031 | 0.044 | 0.108 | 0.226 | 0.062 | 0.034 | 0.672 | 0.527 | 0.141 |
| Instructor-XL | w/o Inst. | 0.328 | 0.323 | 0.364 | 0.040 | 0.059 | 0.144 | 0.564 | 0.084 | 0.003 | 0.599 | 0.504 | 0.146 |
| | w/ Inst. | 0.282 | 0.302 | 0.319 | 0.043 | 0.056 | 0.115 | 0.324 | 0.076 | 0.022 | 0.580 | 0.495 | 0.154 |
| bge-small | w/o Inst. | 0.116 | 0.254 | 0.239 | 0.008 | 0.017 | 0.090 | 0.103 | 0.077 | 0.036 | 0.450 | 0.424 | 0.093 |
| | w/ Inst. | 0.013 | 0.234 | 0.208 | 0.010 | 0.020 | 0.077 | 0.054 | 0.073 | 0.029 | 0.465 | 0.415 | 0.070 |
| bge-base | w/o Inst. | 0.110 | 0.266 | 0.257 | 0.009 | 0.014 | 0.097 | 0.138 | 0.076 | 0.034 | 0.469 | 0.465 | 0.097 |
| | w/ Inst. | 0.041 | 0.240 | 0.230 | 0.002 | 0.012 | 0.088 | 0.103 | 0.073 | 0.027 | 0.456 | 0.463 | 0.072 |
| bge-large | w/o Inst. | 0.131 | 0.285 | 0.280 | 0.010 | 0.018 | 0.100 | 0.192 | 0.111 | 0.030 | 0.574 | 0.481 | 0.112 |
| | w/ Inst. | 0.009 | 0.262 | 0.233 | 0.006 | 0.027 | 0.089 | 0.103 | 0.096 | 0.023 | 0.498 | 0.453 | 0.075 |
| E5-Mistral | w/o Inst. | 0.189 | 0.322 | 0.328 | 0.051 | 0.070 | 0.205 | 0.452 | 0.185 | 0.109 | 0.779 | 0.798 | 0.227 |
| | w/ Inst. | 0.261 | 0.349 | 0.394 | 0.054 | 0.081 | 0.178 | 0.412 | 0.188 | 0.099 | 0.740 | 0.785 | 0.236 |
| GritLM | w/o Inst. | 0.296 | 0.360 | 0.358 | 0.057 | 0.087 | 0.166 | 0.521 | 0.212 | 0.016 | 0.830 | 0.832 | 0.211 |
| | w/ Inst. | 0.340 | 0.395 | 0.444 | 0.072 | 0.116 | 0.266 | 0.537 | 0.268 | 0.094 | 0.824 | 0.838 | 0.290 |
| Cohere-Embed-v3 | w/o Inst. | 0.151 | 0.263 | 0.285 | 0.043 | 0.041 | 0.099 | 0.580 | 0.151 | 0.038 | 0.723 | 0.572 | 0.166 |
| | w/ Inst. | 0.187 | 0.290 | 0.279 | 0.050 | 0.078 | 0.101 | 0.650 | 0.171 | 0.033 | 0.721 | 0.566 | 0.186 |
| OpenAI-ada-002 | w/o Inst. | 0.256 | 0.293 | 0.310 | 0.031 | 0.058 | 0.133 | 0.197 | 0.100 | 0.042 | 0.732 | 0.834 | 0.167 |
| | w/ Inst. | 0.106 | 0.248 | 0.239 | 0.026 | 0.058 | 0.118 | 0.114 | 0.096 | 0.048 | 0.673 | 0.824 | 0.137 |
| OpenAI-3-small | w/o Inst. | 0.306 | 0.309 | 0.337 | 0.030 | 0.061 | 0.146 | 0.315 | 0.125 | 0.066 | 0.711 | 0.720 | 0.190 |
| | w/ Inst. | 0.212 | 0.272 | 0.296 | 0.030 | 0.066 | 0.138 | 0.255 | 0.129 | 0.036 | 0.643 | 0.721 | 0.165 |
| OpenAI-3-large | w/o Inst. | 0.373 | 0.341 | 0.420 | 0.034 | 0.101 | 0.240 | 0.291 | 0.164 | 0.075 | 0.901 | 0.896 | 0.236 |
| | w/ Inst. | 0.342 | 0.314 | 0.375 | 0.050 | 0.136 | 0.212 | 0.339 | 0.211 | 0.074 | 0.877 | 0.894 | 0.250 |

Table 6.2: Full-dataset Retrieval (nDCG@10 performance)

**Main results** Table 6.2 presents the results for nDCG@10 performance. Figure 6.1 outlines the relative performance of the evaluated models. Because of the different scales of nDCG@10 across tasks due to different task difficulties and corpus sizes, we take the geometric mean across tasks to represent each model's average performance, which is given by $\boldsymbol{G} = \left(\prod_{i=1}^{n} \boldsymbol{x_i}\right)^{\frac{1}{n}}$, where $\boldsymbol{n}$ is the number of tasks and $\boldsymbol{x_i}$ represent the performance of each task. We find geometric mean to be more reflective of model's average performance, compared to harmonic mean and Z-scored mean. Figure 6.2 provides insights into the performance gain/degrade brought by prepending task instructions before queries. Similarly, due to the vastly different scales of performance across tasks, simply taking the arithmetic gain ($(\boldsymbol{x}_{\text{with inst.}} - \boldsymbol{x}_{\text{without inst.}}) / \boldsymbol{x}_{\text{without inst.}}$) is misleading (by biasing towards a low $\boldsymbol{x}_{\text{without inst.}}$ base). Therefore, we opt for using logarithmic scaling to dampen the effect of large percentage gains from a low base, given by $\boldsymbol{log_2}(\boldsymbol{x}_{\text{with inst.}_i} + \boldsymbol{1}) - \boldsymbol{log_2}(\boldsymbol{x}_{\text{without inst.}} + \boldsymbol{1})$.

Overall, it is seen that newer models tend to outperform older ones on RAR-b. We believe the enhancement in embedding models' language understanding abilities is relevant to the diversity of training data and instruction-tuning (35), with the current widely-adopted paradigm of instruction-aware information retrieval (33; 34; 35; 36). Injecting instructions understanding abilities into retrievers are intuitively beneficial for modeling more nuanced semantics in natural language, such as intents, improving the alignment of query and groundtruth answer in reasoning tasks in the semantic space, especially for the ones that make less sense without specifying what the task is doing. Further, attaining more accurate understanding of queries from diverse tasks allows generalizing to retrieving correct answers for unseen tasks. Generalization achieved from task diversity and decoder architecture is known for generative tasks (117; 118; 119), but more rigorous studies need to confirm the pattern for embedding models, especially for decoder models, which present the best potential on RAR-b.

**Scaling laws observed** We observe a scaling behavior through the varied versions of models of same classes. The pattern is consistent for Instructor (33), BGE (113) and OpenAI-Embedding-3. Models display a performance gain as their size increases. Since the level-1 datasets are more or less seen in Instructor's training through the SuperNI collection (119), its performance gain on these datasets is more relevant to a stronger fitting ability to training set due to larger model capacities, but not necessarily a stronger generalizability. On the other hand, the same pattern observed in BGE models (113) is more indicative of a stronger generalizability achieved with larger model sizes, because BGE models have not seem these tasks according to the technical report (113). Although the finding is consistent with (120) who find that larger-capacity models are more generalizable retrievers, it is unclear whether this comes from the difference in the pretraining data of the base model or the finetuning data when adapting to representation models, especially in our case of reasoning tasks. While Instructor and BGE are encoder models, the same trend observed in OpenAI v3 models confirms the scaling pattern for decoders, which are well-known to scale for generative tasks (121; 122; 123), but less known for embedding tasks (124).

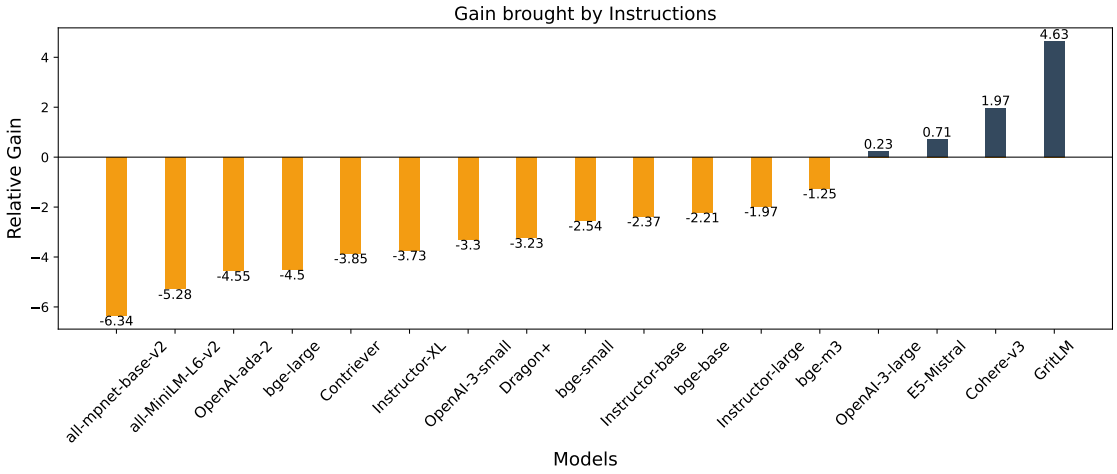


Figure 6.2: Gain brought by instruction (Full Setting). The metric is represented by log-scaled gains averaged across tasks, given by $\sum_{x \in X}(\log_2(x_{\text{w/ inst.}_i} + 1) - \log_2(x_{\text{w/o inst.}} + 1))/|X|$, $X$ being the set of tasks.

**Distraction of Instructions** It is observed that instructions generally distract the retrievers' from focusing on the content, even for the I-DR models. However, this is more observed for encoder models like Instructor and BGE, and smaller variants of decoder models such as `OpenAI-embedding-3-small`.

Latest instruction-aware decoder representational models (35; 36) seem to start bridging this behavioral gap, pointing the promising direction of scaling decoder models for next-level representational abilities. Cohere's proprietary `Embed-English-v3.0` presents a strong gain brought by instructions. Considering its relatively early release time (Nov 02, 2023) compared to latest models that yield better absolute performance (OpenAI-v3, E5-Mistral, GritLM) on RAR-b, this behavior is surprising, and might be relevant to the efforts put in relevance feedback from LLMs [3]. However, no public information is available whether Cohere's current embedding model is an encoder or a decoder, and thus its strong instruction-following abilities can not add to our main narrative of the potentail for decoder representation models.

For models that degrade with instructions, it is more severe in full-dataset retrieval settings compared to the Multiple-choice Retrieval setting as shown in later parts, intuitively because of a larger candidate pool to be distracted by. This phenomenon is most pronounced for older-generation embedding models such as

[3] `https://txt.cohere.com/introducing-embed-v3/`

Sentence Transformer (18) model `all-mpnet-base-v2`. Though sharing the same training set with `all-MiniLM-L6-v2` while both without training to follow instructions, we suspect it's more sensitive than the latter to instructions because of its larger capacity. The key takeaway here is, if a model is not trained to follow instructions, its retrieval performance might be largely distracted by instructions, hypothetically as the model capacity gets larger. However, we note that we manually write an instruction for each task, aligning with previous work (33). Although the prompt for each task is fairly applied to all models, it is possible that the observed instruction sensitivity conflates with prompt distribution shift which hurts more for encoder models. We encourage future work on retrieval instruction following to provide more comprehensive analysis on the effect of instruction style to instruction following performance.

With Sentence Transformer's earliest models being the pioneer in sentence embeddings in the contextualized LM's era, the field has now entered a stage where sentence emebedding models need to model more complex nuances in human language.

**Dual-retrievers present decent performance, if not as late-interaction decision-makers.** In general, the nDCG@10 performance in Full setting is not as bad as the slightly-higher-than-chance performance in MCR setting as will be shown in later section (Table 6.3). The performance patterns are roughly on-par with how they would perform in pure topical-based IR datasets (e.g., the ones in BEIR (14)). Judging from recall@10 (Appendix 6.9), the percentages when correct answers can be extracted within top-10 documents are decent, except for SocialiQA. However, it is still far from meeting the capability of retrieving all candidates to assist later LLMs' decisions, which ideally require that at least the correct answer is in the top-n retrieved documents (if not ranked at top), i.e., recall@n $\approx 1$.

### 6.3.2 Does Reranker Help?

In a standard two-step retrieval pipeline, reranking models are utilized to further rerank the first-round results retrieved with bi-encoder embeddings. As we will show in the next sub-section through multiple-choice retrieval (Table 6.3), nowadays'

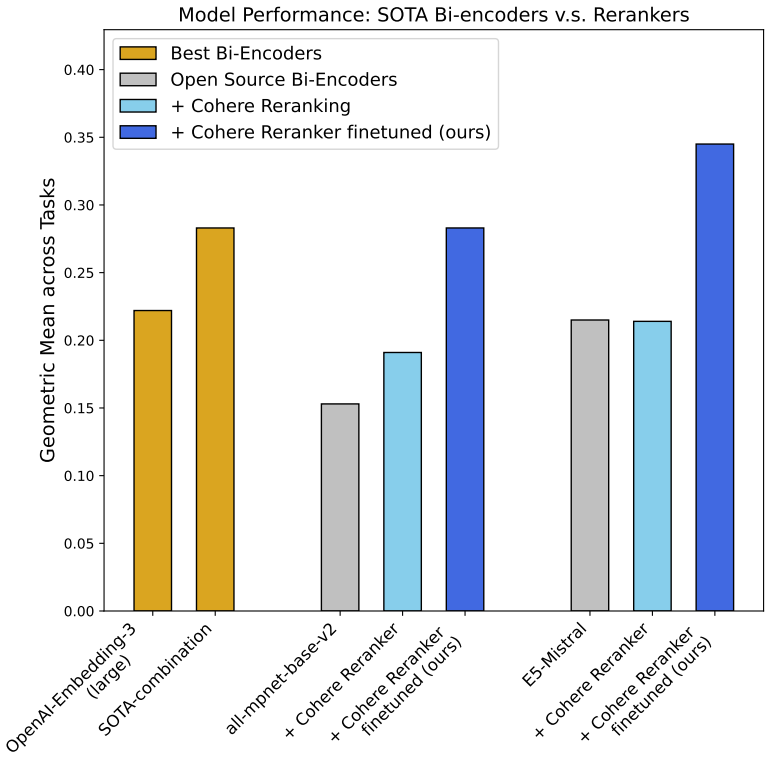


Figure 6.3: Does Reranking improve performance?

reranker models are not trained to understand reasoning-level language, and are already outperformed by recent bi-encoders with decoder backbones trained with instruction tuning (35; 36).

But can we fine-tune the reranker to improve performance on such tasks? Note that it is non-trivial to fine-tune a bi-encoder on each task: with varied sizes of each training set and the de-facto usage of InfoNCE loss (20), we do not want to overfit a bi-encoder to an isotropic distribution for one reasoning task (28). And intuitively, cross-attention enables faster convergence to tasks with deeper language understanding requirements.

In this section, we finetune Cohere's proprietary `rerank-english-v2.0` model through API finetuning [4], because of its well-configured infrastructure. For each task, we construct a training set using its original MCR-format dataset, with the groundtruth being the positive pair for each question, and all non-groundtruth candidates as hard negatives. We evaluate $\boldsymbol{\alpha}$NLI, HellaSwag, PIQA, SocialIQA

[4]https://dashboard.cohere.com/fine-tuning

and Winogrande to understand the pattern, and fine-tune a reranker model for each task. The average performance is outlined in Figure 6.3 (without-instruction nDCG@10 presented), where we can easily achieve state-of-the-art performance by re-ranking the top 100 documents with a fine-tuned Cohere reranker. Due to the cross-encoder nature of rerankers, the models are able to process the query and documents within the same context, making them easier to decompose and process the label information in training and adapt better to reasoning tasks. However, their performance is bounded by first-stage retrieval. Summarizing these findings, we see the important role of rerankers for reasoning-intensive tasks in a full retrieval pipeline.

Although recent papers have confirmed generative models' potential as rerankers (125; 126), they are neither trivial to train nor cheap (yet) in inference. Considering the release time of Cohere's rerankers, we confirm using previous established cross-encoder frameworks is still the simple, robust and strong solution for such tasks. However, we envision decoder models to still be the future for scaling for complicated tasks, and position the effective training of decoder-based rerankers and resource-efficient inference of them as a valuable research direction.

### 6.3.3 Multiple-choice Retrieval Setting

Table 6.3 presents the results of the Multiple-choice Retrieval (MCR) setting, where each dataset has a baseline characterized by random predictions. The MCR setting provides an in-depth check of retrievers' nuanced understanding. Although we evaluate MCR with a separate implementation, we note that in essence, MCR is equivalent to checking whether the correct choice in ranked above all wrong choices in the Full setting.

The findings could be summarized as follows:

**In-depth Understanding remains hard** From Table 6.3, we can easily find that models struggle with Winogrande (90) and CSTS (75) at around chance level. Although some models perform good on Winogrande in the Full setting, we find that this performance might be more of a mirage. For example, Contriever can

| | αNLI | HellaSwag | PIQA | SIQA | WinoG | ARC-C | GeoMean-6 | cSTS-E | cSTS-H | GeoMean |
|---|---|---|---|---|---|---|---|---|---|---|
| *Without Instruction* | | | | | | | | | | |
| random baseline | 50.00 | 25.00 | 50.00 | 33.33 | 50.00 | 25.00 | 37.09 | 50.00 | 50.00 | 39.97 |
| contriever | 51.37 | 30.66 | 57.51 | 38.54 | 49.80 | 21.59 | 39.42 | 47.29 | 43.11 | 40.78 |
| mpnet | 55.35 | 34.98 | 58.71 | 41.50 | 51.30 | 27.90 | 43.48 | 49.65 | 40.92 | 43.87 |
| minilm | 49.61 | 32.20 | 57.34 | 39.41 | 51.30 | 24.06 | 40.57 | 47.38 | 45.20 | 41.92 |
| dragon | 51.57 | 31.40 | 58.54 | 39.30 | 50.75 | 24.15 | 40.73 | 45.29 | 44.24 | 41.70 |
| Instructor-base | 55.94 | 34.44 | 61.15 | 42.73 | 49.49 | 29.18 | 44.01 | 48.52 | 51.66 | 45.45 |
| Instructor-large | 57.18 | 38.48 | 63.71 | 46.78 | 48.62 | 31.48 | 46.44 | 47.47 | 45.90 | 46.50 |
| Instructor-XL | 57.44 | 39.80 | 65.45 | 43.86 | 50.12 | 31.48 | 46.69 | 49.83 | 47.56 | 47.18 |
| BGE-small | 52.87 | 32.56 | 58.11 | 41.20 | 47.83 | 23.29 | 40.77 | 44.42 | 50.26 | 42.30 |
| BGE-base | 53.59 | 33.58 | 59.30 | 42.37 | 51.22 | 25.09 | 42.40 | 48.08 | 50.87 | 44.06 |
| BGE-large | 54.05 | 34.89 | 59.79 | 42.73 | 50.20 | 26.19 | 43.02 | 49.30 | 52.62 | 44.87 |
| BGE-m3 | 52.74 | 33.23 | 56.64 | 45.45 | 48.78 | 24.83 | 41.97 | 43.72 | 45.55 | 42.62 |
| E5-Mistral | 62.01 | 42.40 | 72.69 | 50.72 | 52.64 | 48.38 | 53.96 | 55.58 | 46.16 | 53.11 |
| GritLM | 62.27 | 51.18 | 72.85 | 51.33 | 53.12 | 39.68 | 54.12 | 47.29 | 46.16 | 52.16 |
| Cohere-Embed-v3 | 56.40 | 34.59 | 59.85 | 50.00 | 49.01 | 25.60 | 44.07 | 46.77 | 45.03 | 44.52 |
| OpenAI-3-small | 55.48 | 36.21 | 63.22 | 44.78 | 50.20 | 31.57 | 45.62 | 49.04 | 46.60 | 46.16 |
| OpenAI-3-large | 57.64 | 42.50 | 70.62 | 47.59 | 50.59 | 47.70 | 52.04 | 48.52 | 47.29 | 50.98 |
| BGE-rerank-base | 52.15 | 31.86 | 54.62 | 40.58 | 49.49 | 21.93 | 39.84 | 47.38 | 48.52 | 41.73 |
| BGE-rerank-large | 55.94 | 35.38 | 59.19 | 44.47 | 50.12 | 26.79 | 43.73 | 48.08 | 49.74 | 44.97 |
| TART-Full | 57.77 | 36.01 | 63.22 | 48.98 | 54.62 | 34.04 | 47.83 | 45.55 | 48.87 | 47.67 |
| Cohere-rerank | 57.25 | 37.46 | 64.04 | 42.48 | 50.28 | 40.10 | 47.69 | 40.23 | 38.13 | 45.40 |
| + finetuned | 58.62 | 50.57 | 64.47 | 57.32 | 48.93 | 43.43 | 53.44 | 55.67 | 74.52 | 55.99 |
| *With Instruction* | | | | | | | | | | |
| random baseline | 50.00 | 25.00 | 50.00 | 33.33 | 50.00 | 25.00 | 37.09 | 50.00 | 50.00 | 39.97 |
| contriever | 49.80 | 30.87 | 55.66 | 38.95 | 50.28 | 22.53 | 39.46 | 49.74 | 44.76 | 41.26 |
| mpnet | 54.11 | 31.54 | 56.26 | 41.25 | 51.46 | 27.39 | 42.12 | 54.36 | 40.05 | 43.21 |
| minilm | 50.26 | 31.50 | 56.86 | 39.30 | 49.57 | 23.98 | 40.17 | 51.40 | 45.81 | 42.12 |
| dragon | 51.57 | 31.26 | 57.24 | 39.20 | 51.54 | 24.91 | 40.85 | 49.04 | 44.50 | 42.24 |
| Instructor-base | 55.48 | 33.82 | 60.39 | 42.63 | 51.38 | 29.44 | 44.05 | 48.60 | 50.70 | 45.39 |
| Instructor-large | 57.77 | 37.28 | 63.66 | 45.39 | 49.96 | 31.66 | 46.29 | 48.78 | 44.76 | 46.40 |
| Instructor-XL | 57.05 | 39.09 | 64.53 | 44.22 | 50.67 | 31.31 | 46.49 | 48.34 | 46.07 | 46.66 |
| BGE-small | 52.15 | 32.25 | 56.69 | 42.02 | 49.57 | 22.53 | 40.59 | 48.95 | 52.09 | 42.87 |
| BGE-base | 53.39 | 33.04 | 57.34 | 42.78 | 50.75 | 24.32 | 41.81 | 52.44 | 48.52 | 43.82 |
| BGE-large | 54.37 | 35.01 | 59.30 | 43.19 | 49.09 | 26.02 | 42.90 | 56.28 | 51.66 | 45.42 |
| BGE-m3 | 54.31 | 32.65 | 56.53 | 45.50 | 48.78 | 24.83 | 42.04 | 46.16 | 47.03 | 43.14 |
| E5-Mistral | 64.30 | 42.91 | 73.78 | 48.82 | 53.67 | 44.71 | 53.65 | 53.58 | 46.95 | 52.75 |
| GritLM | 66.58 | 52.16 | 74.59 | 56.04 | 55.25 | 53.50 | 59.17 | 47.03 | 43.72 | 55.36 |
| Cohere-Embed-v3 | 56.53 | 34.67 | 59.09 | 48.57 | 50.04 | 25.17 | 43.82 | 51.92 | 48.25 | 45.31 |
| OpenAI-3-small | 54.90 | 34.02 | 58.92 | 45.14 | 49.49 | 30.55 | 44.25 | 52.36 | 48.60 | 45.72 |
| OpenAI-3-large | 58.42 | 40.30 | 64.96 | 50.00 | 51.85 | 45.99 | 51.30 | 55.24 | 46.60 | 51.16 |
| BGE-rerank-base | 52.35 | 32.04 | 55.50 | 39.66 | 49.41 | 22.18 | 39.92 | 47.99 | 53.23 | 42.34 |
| BGE-rerank-large | 53.92 | 33.73 | 56.96 | 44.63 | 50.12 | 26.96 | 42.92 | 50.17 | 51.66 | 44.79 |
| TART-Full | 57.96 | 31.47 | 63.11 | 52.25 | 55.56 | 38.23 | 48.35 | 46.77 | 55.06 | 48.94 |
| Cohere-rerank | 55.61 | 35.91 | 61.32 | 42.37 | 50.51 | 40.10 | 46.80 | 41.19 | 44.85 | 45.82 |
| + finetuned | 58.29 | 51.03 | 64.36 | 57.73 | 49.64 | 43.86 | 53.73 | 60.65 | 75.39 | 56.91 |

Table 6.3: MCR performance

achieve an nDCG@10 of 0.471 on Winogrande without instruction in the Full setting (Table 6.2), outperforming even the OpenAI-Embedding-3-large model. However, its performance on MCR is stuck at chance level. We find that, the two candidate answers in Winogrande are both words that exist in the queries (e.g., entities), and it requires only finding these two words and ranked them at the top to get a good performance in the Full setting, making this task fall back on be a name entity matching task for the Full setting, without the models needing to develop a fine-grained understanding in distinguishing them.

Multiple-choice retrieval settings serve as a proxy measurement of how well the correct reasoning can be found through representation matching, given a few selected

possible answers already. From the MCR experiments, we can draw the conclusion that **current dual-dense retrievers fail to serve as late interaction reasoners**.

**Generalization Gap** A special case in the models is Instructor, which is known to have seen the formats of the first 6 tasks through finetuning on Super-NI (119), enabling it to achieve a better performance on these tasks compared to traditional S-DR models. Yet, its performance on the two settings of CSTS that we construct is stuck at chance level. Without solid control experiments, we hypothesize that this pattern stems from the subpar generalizablity of encoder models. Although the pattern can currently be empirically supported by the better performance achieved by similarly instruction-tuned decoder models including E5-Mistral, GritLM and OpenAI models, the generalization gap between encoders and decoders has not yet been systematically investigated on information retrieval, and we position this as a valuable research topic.

**Unsupervised Model is strong and robust to instructions.** Surprisingly, Contriever, as an unsupervised model only trained on different spans of text without labeled document pairs, serves as a strong baseline for all datasets. We speculate this pattern is due to that the formats of question-answer pair in reasoning datasets look alike the way that Contriever is trained on. For instance, a question and its answer in HellaSwag essentially compose a consecutive document, while being alike a span pair that Contriever is trained on. For this reason as well, Contriever is extremely robust to instructions, because a prefix instruction does not make the following question and answer looking less like a consecutive span in the document.

## 6.4 Behavioral Analysis

In this section, we present more in-depth understanding of retrievers' behaviors through the tasks that have multiple settings.

### 6.4.1 Behavioral Analysis through the lens of Temporal Reasoning

With the 7 sub-settings we construct for TempReason (Table 6.4), we find that it provides a multi-faceted breakdown of retrievers' behaviors, providing understanding of their **parameterized knowledge**, **reading comprehension abilities**, and **robustness against irrelevant context**, and thus we provide an analysis of it in this sub-section. TempReason-L1 (TR-1) in the original dataset concerns evaluating the built-in time perception of models, using time arithmetic questions. We evaluate this task with only one setting - the standalone setting (denoted as **Pure**). Query and groundtruth document in TR-1 looks like "When is 7 months after Nov, 1998?" and "June, 1999".

For TempReason-L2 (TR-2) and TempReason-L3 (TR-3), we respectively construct three settings: **Pure**, **Fact**, and **Context+Fact**. These two levels of tasks evaluate question answering situated in time, providing us the understanding of the three aspects outlined above.

**Parameterized Knowledge** *Pure setting* here evaluates the built-in knowledge stored in retrievers. The question here can look like "who is the president of United States in 2003?" (note that questions in TempReason are in a similar format but are much harder and niche than this), and the retrievers thus need to encode "George W. Bush" to be among having the most similar embedding. We can see the Pure setting evaluates factual knowledge parameterized in retrievers without references.

**Reading Comprehension** For the *Fact setting*, we concatenate the facts (a paragraph describing the answer to the question in different times - such as the history of US presidential tenures) after the question as query. Concretely, the query can look like: `Question: Who was the president of the United States in 2003? Facts: The US president from 2001 to 2009 was George W. Bush. Barack Obama's tenure began on January 20, 2009, and ended on January 20, 2017. Donald Trump began his presidency in 2017, and his tenure ended in 2021.` and ideally, `George W. Bush` has an embedding close to the above long passage.

Note that this above example is for illustrating the concept with the same format as TempReason, and is not from the TempReason dataset, as TempReason is made of much less well-known people and facts, and thus more difficult. In the setting, we are essentially evaluating retrievers' reading comprehension abilities. Intuitively, the "fact" is laid out in the query and the embedding of the query is expected to be guided towards that of the groundtruth document, by the facts. Another way to describe this setting is that we let the retrievers play the role of LLM in a RAG system in this setting.

**Robustness against Irrelevant Contexts** Lastly, for the Context+Fact setting, we further concatenate the context (the background knowledge of the entity in the question that is going to be asked) after the facts. On top of the Fact setting, the Context+Fact setting further reflects a model's robustness towards irrelevant information (since the answer can be extracted only from facts, but not from contexts), and encoding capabilities for texts of varied length ranges (31). However, there potentially is unfair comparison in here because of the max sequence length each model is capable of encoding. For instance, OpenAI embedidng models are able to encode sequence up to 8192 tokens and are faced with more noise to be robust against. On the other hand, models with max sequence length of 512 need to deal with less noise. Therefore, the Context-Fact setting here is only for analysis purposes, and we do not average the scores from this setting into the average TR scores in the main table. For E5-Mistral and GritLM, we set its max sequence length to 8192 here to match with OpenAI models, for attaining a comparable understanding on the Context+Fact setting.

### 6.4.2 Behavioral Analysis through the lens of Code Retrieval

Another area that we can get a more fine-grained understanding about retrievers' behaviors is through code retrieval, given the extensive settings we explored in the construction of the final pooled evaluation dataset.

| **Models →** | | TR-1 | TR-2 | | | TR-3 | | | avg. | avg. (fair) |
|---|---|---|---|---|---|---|---|---|---|---|
| **Test Dataset ↓** | | pure | pure | fact | context+fact | pure | fact | context+fact | | |
| Contriever | w/o Inst. | 0.019 | 0.011 | 0.227 | 0.207 | 0.078 | 0.206 | 0.194 | 0.135 | 0.108 |
| | w/ Inst. | 0.018 | 0.009 | 0.220 | 0.205 | 0.071 | 0.208 | 0.193 | 0.132 | 0.105 |
| all-mpnet-base-v2 | w/o Inst. | 0.018 | 0.012 | 0.112 | 0.095 | 0.056 | 0.094 | 0.078 | 0.066 | 0.058 |
| | w/ Inst. | 0.015 | 0.010 | 0.073 | 0.076 | 0.052 | 0.071 | 0.067 | 0.052 | 0.044 |
| all-MiniLM-L6-v2 | w/o Inst. | 0.015 | 0.005 | 0.176 | 0.100 | 0.063 | 0.141 | 0.087 | 0.084 | 0.080 |
| | w/ Inst. | 0.010 | 0.005 | 0.165 | 0.095 | 0.063 | 0.138 | 0.087 | 0.081 | 0.076 |
| Dragon+ | w/o Inst. | 0.018 | 0.006 | 0.175 | 0.110 | 0.080 | 0.157 | 0.099 | 0.092 | 0.087 |
| | w/ Inst. | 0.015 | 0.006 | 0.161 | 0.110 | 0.075 | 0.148 | 0.095 | 0.087 | 0.081 |
| bge-m3 | w/o Inst. | 0.010 | 0.007 | 0.332 | 0.223 | 0.053 | 0.301 | 0.192 | 0.160 | 0.140 |
| | w/ Inst. | 0.008 | 0.006 | 0.350 | 0.263 | 0.070 | 0.325 | 0.230 | 0.179 | 0.152 |
| Instructor-base | w/o Inst. | 0.007 | 0.006 | 0.109 | 0.066 | 0.055 | 0.120 | 0.076 | 0.063 | 0.059 |
| | w/ Inst. | 0.007 | 0.005 | 0.091 | 0.064 | 0.055 | 0.100 | 0.073 | 0.056 | 0.052 |
| Instructor-large | w/o Inst. | 0.007 | 0.011 | 0.123 | 0.081 | 0.069 | 0.133 | 0.084 | 0.073 | 0.068 |
| | w/ Inst. | 0.007 | 0.011 | 0.104 | 0.070 | 0.072 | 0.117 | 0.079 | 0.066 | 0.062 |
| Instructor-XL | w/o Inst. | 0.006 | 0.013 | 0.171 | 0.099 | 0.074 | 0.156 | 0.093 | 0.088 | 0.084 |
| | w/ Inst. | 0.008 | 0.013 | 0.144 | 0.089 | 0.077 | 0.137 | 0.088 | 0.079 | 0.076 |
| bge-small | w/o Inst. | 0.014 | 0.010 | 0.176 | 0.070 | 0.048 | 0.139 | 0.059 | 0.074 | 0.077 |
| | w/ Inst. | 0.013 | 0.011 | 0.167 | 0.085 | 0.046 | 0.128 | 0.074 | 0.075 | 0.073 |
| bge-base | w/o Inst. | 0.011 | 0.013 | 0.172 | 0.099 | 0.052 | 0.134 | 0.082 | 0.080 | 0.076 |
| | w/ Inst. | 0.008 | 0.013 | 0.166 | 0.111 | 0.051 | 0.127 | 0.092 | 0.081 | 0.073 |
| bge-large | w/o Inst. | 0.015 | 0.024 | 0.242 | 0.177 | 0.067 | 0.206 | 0.146 | 0.125 | 0.111 |
| | w/ Inst. | 0.012 | 0.021 | 0.212 | 0.158 | 0.060 | 0.176 | 0.126 | 0.109 | 0.096 |
| E5-Mistral | w/o Inst. | 0.030 | 0.093 | 0.356 | 0.247 | 0.144 | 0.304 | 0.213 | 0.198 | 0.185 |
| | w/ Inst. | 0.033 | 0.092 | 0.369 | 0.281 | 0.143 | 0.302 | 0.231 | 0.207 | 0.188 |
| GritLM | w/o Inst. | 0.025 | 0.090 | 0.482 | 0.289 | 0.125 | 0.340 | 0.225 | 0.225 | 0.212 |
| | w/ Inst. | 0.072 | 0.112 | 0.576 | 0.335 | 0.141 | 0.439 | 0.261 | 0.277 | 0.268 |
| Cohere-Embed-v3 | w/o Inst. | 0.015 | 0.019 | 0.359 | 0.197 | 0.085 | 0.275 | 0.162 | 0.159 | 0.151 |
| | w/ Inst. | 0.014 | 0.024 | 0.405 | 0.249 | 0.075 | 0.339 | 0.213 | 0.188 | 0.171 |
| OpenAI-ada-2 | w/o Inst. | 0.017 | 0.026 | 0.199 | 0.163 | 0.076 | 0.180 | 0.148 | 0.116 | 0.100 |
| | w/ Inst. | 0.014 | 0.024 | 0.194 | 0.161 | 0.073 | 0.176 | 0.147 | 0.113 | 0.096 |
| OpenAI-3-small | w/o Inst. | 0.023 | 0.028 | 0.257 | 0.165 | 0.098 | 0.221 | 0.152 | 0.135 | 0.125 |
| | w/ Inst. | 0.023 | 0.032 | 0.263 | 0.165 | 0.100 | 0.227 | 0.152 | 0.137 | 0.129 |
| OpenAI-3-large | w/o Inst. | 0.021 | 0.103 | 0.286 | 0.187 | 0.153 | 0.255 | 0.165 | 0.167 | 0.164 |
| | w/ Inst. | 0.021 | 0.110 | 0.398 | 0.272 | 0.155 | 0.370 | 0.233 | 0.223 | 0.211 |

Table 6.4: TempReason all sub-task Full-Retrieval nDCG@10 Results. For TempReason-L2 and TempReason-L3, we construct 3 settings: Pure, Fact, and Context+Fact. The pure setting reflects the standalone knowledge parameterized into retrieval models. The Fact setting reflects the reading comprehension abilities of retrieval models. And the Context+Fact setting is an indicator of retrievers' reading comprehension abilities subtracted by their vulnerability against irrelevant contexts.

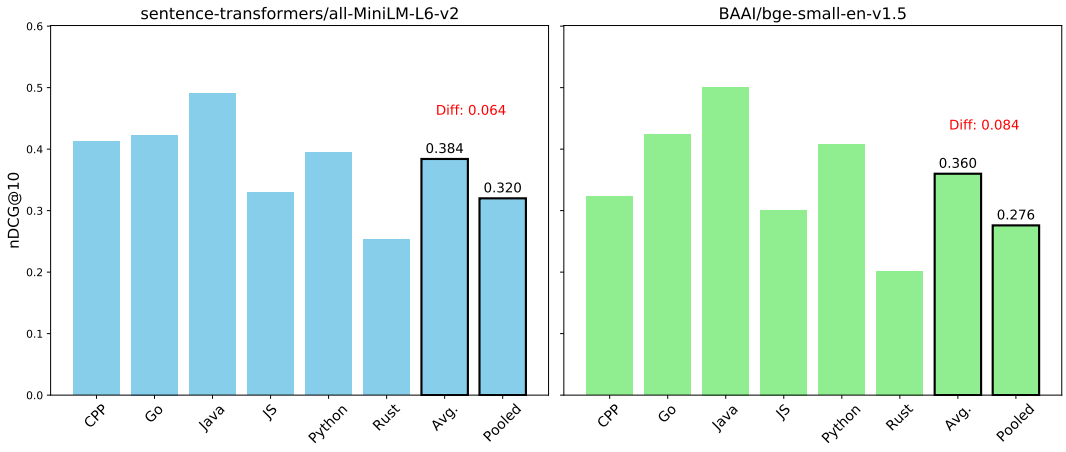


Figure 6.4: A glimpse of distraction from hard negatives of same content across languages.

**Hard Negatives across Programming Languages** The translated nature of HumanEvalPack (102) grants us the convenience to inspect models' familiarity with the same content of different programming languages, and when pooling them together, to what extent do the hard negatives come from the failure of distinguishing across languages through instructions (e.g., for entries that require "retrieve a Python snippet", a Javascript snippet of same content is instead ranked above the groundtruth Python snippet).

Here, we inspect the results of 7 settings, respectively with only queries from each of the 6 languages, and the pooled queries from all languages. For each setting, we enlarge the corpus with the same 200k documents sampled from CodeSearchNet and 100k documents from TinyCode, aligning with our final setting. However, we see a performance gap between averaging individual settings and the pooled setting, meaning that for a lot of queries, the "semi-groundtruth" is placed over the groundtruth, indicating the model's failure of perceiving about programming languages through instructions.

Figure 6.4 depicts the pattern. If the queries are not distracted by code of same content from different languages (i.e., same content from different languages is ranked above the groundtruth), the average performance of the single-language settings should actually match with the performance of the pooled setting, because in the pooled setting, the nDCG@10 is essentially nDCG@10 performance averaged across

different languages.

| Settings | | HumanEvalPack | | TinyCode |
|---|---|---|---|---|
| | format → | Original | Instruct | - |
| Contriever | w/o Inst. | 0.051 | 0.203 | 0.056 |
| | w/ Inst. | 0.048 | 0.228 | 0.044 |
| all-mpnet-base-v2 | w/o Inst. | 0.419 | 0.897 | 0.142 |
| | w/ Inst. | 0.408 | 0.867 | 0.123 |
| Instructor-XL | w/o Inst. | 0.396 | 0.920 | 0.209 |
| | w/ Inst. | 0.374 | 0.915 | 0.240 |

Table 6.5: Left: Performance difference between Original and Instruct setting of HumanEvalPack. Right: Performance of TinyCode, providing a glimpse of model behaviors on mixture of natural language and code.

**Entity Shortcut** Another property we present is the Entity Shortcut. In Table 6.5 (left), we present the results of two settings we construct for evaluating HumanEvalPack. For the original setting, we simply follow the continuation setting, where the query is the import, function name and the docstring of the function, and the aim is to retrieve the rest of the code. We construct another setting, the "instruct" setting, where we would like to retrieve the complete code (the concatenation of the query and the groundtruth in the "original" setting) with a self-contained instruction.

With the Instruct setting, we see a huge jump of nDCG@10 from the original setting (e.g., 0.374 → 0.915 for Instructor-XL), indicating that the easiness of this setting for the model. An example for this setting is given in Appendix 6.8, demonstrating the entity matching shortcut. Although this setting forms a more self-contained scenario that would happen in real-life question answering, the function declaration presents in both the query and the document, enabling the retrieval to rely solely on the function declaration matching. We speculate this pattern to be highly relevant to the "entourage effect" towards dominant tokens (function declaration in this case) after contrastive learning found in (28). Due to this reason, we opt for the original format in the final evaluation.

**Natural v.s. Symbolic Language** We further conduct an experiment by constructing the queries and corpus with only TinyCode, a synthetic dataset of code question answering. This dataset provides the natural mixture of natural language and code. We sample 15k queries and 150k documents including the groundtruth

to form the corpus. The result is shown in Table 6.5 (right). Although hard to construct controlled experiments, it is seen that models generally perform less well on the mixture of natural language and code.

## 6.5 Conclusion

In this chapter, we underscored the critical need of evaluating retrievers' next-level language understanding abilities, moving beyond the conventional scope of semantic textual similarity and topical matching. We introduced the RAR framework, a novel problem that casts reasoning tasks as retrieval tasks. Through extensive evaluation, we revealed retrievers' behaviors in understanding reasoning-level language and depicted the behavioral discrepancies between retrievers and LLMs, suggesting promising directions in future research in retrieval-augmented generation. We envision that representation models excelling in RAR-b are well-suited to enhance LLMs within RAG systems, positioning RAR-b as a vital benchmark for tracking advancements in the field.

We also see how findings in earlier chapters provide motivations and guide the benchmark design and experiments in this chapter. For instance, inspired by the encoded length information in embeddings in Chapter 5, we intentionally sample code snippets within the same range of the groundtruth documents when constructing the code corpus, making it more difficult. On the other hand, the understanding of isotropy motivates us to prevent fine-tuning bi-encoders on one single reasoning task, which might break the generalization of models overall.

In this chapter, we also see the great potential of decoder-based representation models, and how representation learning can be re-imagined as an alignment of models' representation capabilities to their generative capabilities. In later chapters (Chapter 8), we also explore whether this finding transfers to multi-modality, and whether grounding in more modalities raises the upper bound of this re-imagined "alignment". Last, this chapter also inspires future work directions, such as reasoning-aware representation models trained using RL, which we detail Chapter 9.

## 6.6 Dataset Format

| Dataset | Instruction | Query | Doc |
|---|---|---|---|
| $\alpha$NLI | Given the following start and end of a story, retrieve a possible reason that leads to the end. | Start: Ron started his new job as a landscaper today. End: Ron is immediately fired for insubordination. | Ron ignores his bosses's orders and called him an idiot. |
| HellaSwag | Given the following unfinished context, retrieve the most plausible ending to finish it. | A man is sitting on a roof. He | starts pulling up roofing on a roof. |
| PIQA | Given the following goal, retrieve a possible solution. | How do I ready a guinea pig cage for it's new occupants? | Provide the guinea pig with a cage full of a few inches of bedding made of ripped paper strips, you will also need to supply it with a water bottle and a food dish. |
| SocialiQA | Given the following context and question, retrieve the correct answer. | Context: Tracy didn't go home that evening and resisted Riley's attacks. Question: What does Tracy need to do before this? | Find somewhere to go |
| Quail | Given the following context and question, retrieve the correct answer. | Context: Candy watched the bearded man drive his silver BMW into the convenience store parking lot and pull around to the side, near the back corner of the building. There were plenty of open slots in the front, so she figured the guy was there for something other than a bag of chips and a coke. **{Omitted for brevity}** | |
| ARC-Challenge | Retrieve the answer to the question. | An astronomer observes that a planet rotates faster after a meteorite impact. Which is the most likely effect of this increase in rotation?" | Planetary days will become shorter. |
| Winogrande | Given the following sentence, retrieve an appropriate answer to fill in the missing underscored part. | Sentence: Sarah was a much better surgeon than Maria so _ always got the easier cases.. | Maria |

| | | | |
|---|---|---|---|
| TempReason-L1 | Given the following question about time, retrieve the correct answer. | What is the time 6 year and 4 month after Nov, 1185 | Mar, 1192 |
| TempReason-L2 pure | Given the following question, retrieve the correct answer. | Which employer did Jaroslav Pelikan work for in Jan, 1948? | Valparaiso University |
| TempReason-L2 fact | Given the following question and facts, retrieve the correct answer. | Question: Which employer did Jaroslav Pelikan work for in Jan, 1948? Facts: Jaroslav Pelikan works for Concordia Seminary from Jan, 1949 to Jan, 1953. \n Jaroslav Pelikan works for University of Chicago from Jan, 1953 to Jan, 1962. \n Jaroslav Pelikan works for Valparaiso University from Jan, 1946 to Jan, 1949. \n Jaroslav Pelikan works for Yale University from Jan, 1962 to Jan, 1962. | Valparaiso University |
| TempReason-L2 context | Given the following question, facts and contexts, retrieve the correct answer. | Question: Which employer did Jaroslav Pelikan work for in Jan, 1948? Facts: Jaroslav Pelikan works for Concordia Seminary from Jan, 1949 to Jan, 1953. \n Jaroslav Pelikan works for University of Chicago from Jan, 1953 to Jan, 1962. \n Jaroslav Pelikan works for Valparaiso University from Jan, 1946 to Jan, 1949. \n Jaroslav Pelikan works for Yale University from Jan, 1962 to Jan, 1962. Context: Jaroslav PelikanJaroslav Jan Pelikan Jr. (December 17, 1923 \u2013 May 13, 2006) was an American scholar of the history of Christianity, Christian theology, and medieval intellectual history at Yale University.Jaroslav Jan Pelikan Jr. was born on December 17, 1923, in Akron, Ohio, to a Slovak father Jaroslav Jan Pelikan Sr. and Slovak mother Anna Buzekova Pelikan from \u0160id in Serbia. His father was pastor of .... **{context is too long and omitted for brevity}** | Valparaiso University |

| | | | |
|---|---|---|---|
| TempReason-L3 pure | Given the following question, retrieve the correct answer. | Which employer did Jaroslav Pelikan work for before Concordia Seminary? | Valparaiso University |
| TempReason-L3 fact | Given the following question and facts, retrieve the correct answer. | Question: Which employer did Jaroslav Pelikan work for before Concordia Seminary? Facts: Jaroslav Pelikan works for Yale University from Jan, 1962 to Jan, 1962. \nJaroslav Pelikan works for University of Chicago from Jan, 1953 to Jan, 1962. \nJaroslav Pelikan works for Valparaiso University from Jan, 1946 to Jan, 1949. \nJaroslav Pelikan works for Concordia Seminary from Jan, 1949 to Jan, 1953. | Valparaiso University |
| TempReason-L3 context | Given the following question, facts and contexts, retrieve the correct answer. | Question: Which employer did Jaroslav Pelikan work for before Concordia Seminary? Facts: Jaroslav Pelikan works for Yale University from Jan, 1962 to Jan, 1962. \nJaroslav Pelikan works for University of Chicago from Jan, 1953 to Jan, 1962. \nJaroslav Pelikan works for Valparaiso University from Jan, 1946 to Jan, 1949. \nJaroslav Pelikan works for Concordia Seminary from Jan, 1949 to Jan, 1953. Context: Jaroslav PelikanJaroslav Jan Pelikan Jr. (December 17, 1923 \u2013 May 13, 2006) was an American scholar of the history of Christianity, Christian theology, and medieval intellectual history at Yale University.Jaroslav Jan Pelikan Jr. was born on December 17, 1923, in Akron, Ohio, to a Slovak father Jaroslav Jan Pelikan Sr. and Slovak mother Anna Buzekova Pelikan from \u0160id in Serbia. His father was pastor of .... **{context is too long and omitted for brevity}** | Valparaiso University |

| | | | |
|---|---|---|---|
| SpartQA | Given the following spatial reasoning question, retrieve the right answer. | We have three blocks. Lets call them A, B and C. Block B is below A. Block A is below C. Block A contains a medium yellow square. Block B has two medium blue squares. Medium blue square number one is touching the bottom edge of this block. Medium blue square number two is below a medium yellow square. Medium blue square number one is below the square which is below the medium yellow square. It is below the medium yellow square. Block C contains one medium black square. What is below the black thing? a medium yellow square that is in block A or a medium yellow square that is in block B? | both of them |
| Math-pooled | Retrieve the answer for the following math problem. | **{Omitted for brevity}** | **{Omitted for brevity}** |
| HumanEvalPack-MBPP | Retrieve the answer for the following coding problem. | Finish the following code based on the docstring: **{Omitted for brevity}** | **{Omitted for brevity}** |
| CSTS-easy | Retrieve an aspect and a sentence which are similar to the following sentence. | 3 people wearing festive costumes and feathers dancing in a parade . | In terms of "The number of persons" retrieve a sentence similar to the following. Three uniquely dressed women dancing in a parade . |
| CSTS-hard | Retrieve a condition under which the following two sentences are similar. | Sentence1: 3 people wearing festive costumes and feathers dancing in a parade .; Sentence2: Three uniquely dressed women dancing in a parade . | The number of persons. |

## 6.7 Potential Sparse Annotation Problem of C-STS

CSTS concerns aspect-aware similarity perception between sentence pairs. For each unique sentence pair in C-STS, two conditions are given, yielding two annotated similarity scores ranging from 1-5. Based on the nature of the dataset, we are

only able to construct it into multiple-choice setting, evaluating the retrieval of the condition out of the 2 that provides a higher similarity, but can not construct it into Full-dataset retrieval setting because of the potential sparse annotation problem (14) - a condition, based on which two sentences are more similar, than based on the other condition, does not make this condition the most suitable condition out of the corpus.

## 6.8 Illustration of the Entity Matching Shortcut

In Section 6.4.2, we presented the high performance of the Instruct setting of HumanEvalPack, which is largely due to the entity matching shortcut. We provide an example of a typical query-groudtruth pair in this setting as follows:

Query in "Instruct" setting:

```
Write a Python function 'has_close_elements(numbers: List[float], threshold:
float) -> bool' to solve the following problem:
Check if in given list of numbers, are any two numbers closer to each other than
given threshold.
»> has_close_elements([1.0, 2.0, 3.0], 0.5)
False
»> has_close_elements([1.0, 2.8, 3.0, 4.0, 5.0, 2.0], 0.3)
True
```

Groundtruth in "Instruct" setting:

```
from typing import List



def has_close_elements(numbers: List[float], threshold: float) -> bool:
    for idx, elem in enumerate(numbers):
        for idx2, elem2 in enumerate(numbers):
            if idx != idx2:
                distance = abs(elem - elem2)
                if distance < threshold:
                    return True

    return False
```

With function declaration presenting in both query and groundtruth document, the document can easily be retrieved with the overlapping part perceived by the retrievers.

## 6.9 Full setting Recall@10

Table 6.7 presents the recall@10 performance of the Full retrieval setting, providing a more straightforward measure of the average empirical probability that the groundtruth is retrieved as the top-10 documents.

| Model ↓ Dataset → | | αNLI | HellaSwag | PIQA | SIQA | Quail | ARC-C | WinoG | TR | SpartQA | Math | Code | Geo. Mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Contriever | w/o Inst. | 0.425 | 0.221 | 0.387 | 0.019 | 0.097 | 0.161 | 0.717 | 0.210 | 0.196 | 0.357 | 0.142 | 0.196 |
| | w/ Inst. | 0.371 | 0.271 | 0.335 | 0.016 | 0.096 | 0.147 | 0.463 | 0.202 | 0.184 | 0.258 | 0.110 | 0.171 |
| all-mpnet-base-v2 | w/o Inst. | 0.310 | 0.396 | 0.439 | 0.041 | 0.092 | 0.207 | 0.386 | 0.112 | 0.006 | 0.786 | 0.698 | 0.178 |
| | w/ Inst. | 0.039 | 0.207 | 0.426 | 0.021 | 0.081 | 0.179 | 0.195 | 0.087 | 0.026 | 0.765 | 0.654 | 0.133 |
| all-MiniLM-L6-v2 | w/o Inst. | 0.393 | 0.373 | 0.392 | 0.024 | 0.076 | 0.170 | 0.770 | 0.151 | 0.038 | 0.749 | 0.588 | 0.210 |
| | w/ Inst. | 0.228 | 0.321 | 0.383 | 0.029 | 0.076 | 0.162 | 0.401 | 0.143 | 0.014 | 0.700 | 0.573 | 0.169 |
| Dragon+ | w/o Inst. | 0.425 | 0.421 | 0.412 | 0.028 | 0.081 | 0.166 | 0.948 | 0.167 | 0.182 | 0.517 | 0.251 | 0.232 |
| | w/ Inst. | 0.355 | 0.363 | 0.390 | 0.025 | 0.079 | 0.154 | 0.890 | 0.156 | 0.201 | 0.423 | 0.197 | 0.210 |
| bge-m3 | w/o Inst. | 0.350 | 0.392 | 0.354 | 0.079 | 0.136 | 0.158 | 0.692 | 0.202 | 0.148 | 0.764 | 0.546 | 0.276 |
| | w/ Inst. | 0.352 | 0.391 | 0.297 | 0.076 | 0.128 | 0.158 | 0.607 | 0.215 | 0.136 | 0.722 | 0.560 | 0.265 |
| Instructor-base | w/o Inst. | 0.349 | 0.403 | 0.424 | 0.040 | 0.069 | 0.181 | 0.538 | 0.113 | 0.074 | 0.671 | 0.570 | 0.217 |
| | w/ Inst. | 0.298 | 0.364 | 0.392 | 0.042 | 0.071 | 0.177 | 0.311 | 0.101 | 0.120 | 0.665 | 0.542 | 0.207 |
| Instructor-large | w/o Inst. | 0.354 | 0.437 | 0.497 | 0.059 | 0.097 | 0.221 | 0.510 | 0.136 | 0.032 | 0.788 | 0.728 | 0.236 |
| | w/ Inst. | 0.335 | 0.402 | 0.442 | 0.056 | 0.083 | 0.192 | 0.425 | 0.124 | 0.066 | 0.746 | 0.696 | 0.230 |
| Instructor-XL | w/o Inst. | 0.455 | 0.477 | 0.527 | 0.061 | 0.111 | 0.247 | 0.873 | 0.157 | 0.008 | 0.678 | 0.658 | 0.231 |
| | w/ Inst. | 0.400 | 0.449 | 0.470 | 0.075 | 0.102 | 0.204 | 0.594 | 0.143 | 0.045 | 0.655 | 0.659 | 0.248 |
| bge-small | w/o Inst. | 0.175 | 0.390 | 0.360 | 0.016 | 0.033 | 0.165 | 0.228 | 0.144 | 0.078 | 0.543 | 0.573 | 0.160 |
| | w/ Inst. | 0.024 | 0.357 | 0.320 | 0.018 | 0.039 | 0.142 | 0.122 | 0.138 | 0.058 | 0.548 | 0.567 | 0.121 |
| bge-base | w/o Inst. | 0.170 | 0.403 | 0.389 | 0.022 | 0.028 | 0.174 | 0.332 | 0.146 | 0.070 | 0.565 | 0.615 | 0.170 |
| | w/ Inst. | 0.072 | 0.364 | 0.353 | 0.007 | 0.025 | 0.161 | 0.242 | 0.141 | 0.058 | 0.540 | 0.619 | 0.130 |
| bge-large | w/o Inst. | 0.195 | 0.428 | 0.418 | 0.024 | 0.035 | 0.177 | 0.419 | 0.203 | 0.061 | 0.666 | 0.637 | 0.190 |
| | w/ Inst. | 0.022 | 0.392 | 0.363 | 0.012 | 0.052 | 0.161 | 0.211 | 0.180 | 0.048 | 0.587 | 0.608 | 0.131 |
| E5-Mistral | w/o Inst. | 0.268 | 0.472 | 0.478 | 0.088 | 0.128 | 0.316 | 0.696 | 0.327 | 0.185 | 0.848 | 0.954 | 0.341 |
| | w/ Inst. | 0.377 | 0.508 | 0.561 | 0.090 | 0.144 | 0.276 | 0.678 | 0.333 | 0.196 | 0.816 | 0.945 | 0.360 |
| GritLM | w/o Inst. | 0.410 | 0.501 | 0.511 | 0.091 | 0.159 | 0.271 | 0.852 | 0.374 | 0.030 | 0.888 | 0.952 | 0.318 |
| | w/ Inst. | 0.464 | 0.542 | 0.618 | 0.117 | 0.207 | 0.411 | 0.840 | 0.434 | 0.189 | 0.881 | 0.960 | 0.429 |
| Cohere-Embed-v3 | w/o Inst. | 0.223 | 0.400 | 0.423 | 0.073 | 0.082 | 0.180 | 0.886 | 0.275 | 0.080 | 0.790 | 0.739 | 0.266 |
| | w/ Inst. | 0.272 | 0.441 | 0.417 | 0.082 | 0.140 | 0.182 | 0.939 | 0.298 | 0.074 | 0.787 | 0.737 | 0.291 |
| OpenAI-ada-2 | w/o Inst. | 0.357 | 0.441 | 0.465 | 0.051 | 0.110 | 0.238 | 0.342 | 0.188 | 0.087 | 0.809 | 0.914 | 0.262 |
| | w/ Inst. | 0.178 | 0.381 | 0.378 | 0.046 | 0.108 | 0.213 | 0.199 | 0.181 | 0.094 | 0.756 | 0.903 | 0.221 |
| OpenAI-3-small | w/o Inst. | 0.419 | 0.464 | 0.498 | 0.048 | 0.113 | 0.259 | 0.543 | 0.237 | 0.137 | 0.783 | 0.880 | 0.298 |
| | w/ Inst. | 0.311 | 0.415 | 0.460 | 0.048 | 0.120 | 0.235 | 0.441 | 0.242 | 0.079 | 0.718 | 0.879 | 0.264 |
| OpenAI-3-large | w/o Inst. | 0.504 | 0.498 | 0.596 | 0.054 | 0.189 | 0.388 | 0.511 | 0.306 | 0.167 | 0.940 | 0.966 | 0.362 |
| | w/ Inst. | 0.467 | 0.468 | 0.559 | 0.085 | 0.233 | 0.358 | 0.577 | 0.363 | 0.160 | 0.921 | 0.964 | 0.383 |

Table 6.7: Full-dataset Retrieval (recall@10 performance)

# CHAPTER 7

# Visual Text Representation

In parallel to the exploration of text-only representation models and methods, we see a few disadvantages of text-only models, including their sensitivity to typos due to tokenization, and their high data volume requirement to attain robust performance across languages. This chapter studies the modeling of sentence and document embeddings in **the pixel space** using vision encoders. The core questions to be answered in this chapter is: Can pixel-based models match token-based SRL on semantics and IR, and how do they behave cross-lingually?

Pretrained language models are long known to be subpar in capturing sentence and document-level semantics. Though heavily investigated, transferring perturbation-based methods from unsupervised visual representation learning to NLP remains an unsolved problem. This is largely due to the discreteness of subword units brought by tokenization of language models, limiting small perturbations of inputs to form semantics-preserved positive pairs. In this work, we conceptualize the learning of sentence-level textual semantics as a visual representation learning process. Drawing from cognitive and linguistic sciences, we first introduce an unsupervised visual sentence representation learning framework, employing visually-grounded text perturbations like typos and word order shuffling, resonating with human cognitive

patterns, and enabling perturbation to texts to be perceived as continuous. Our approach is further enhanced by large-scale unsupervised topical alignment training and natural language inference supervision, achieving comparable performance in semantic textual similarity (STS), information retrieval (IR) and reasoning tasks to existing state-of-the-art NLP methods. Additionally, we unveil our method's inherent cross-lingual transferability and a unique leapfrogging pattern across languages during iterative cross-lingual transfer. The framework represents a novel representation learning method devoid of language models for understanding sentence and document semantics, marking a stride closer to human-like textual comprehension.

## 7.1 Introduction

Vanilla language models are long known to have subpar sentence-level representation (18; 35), even worse than averaging static word embeddings (19), i.e., sentence representations attained by pooling from sub-word embeddings encoded by language models do not closely reflect the relative semantics of sentences. Encouraged by the remarkable success of visual representation learning facilitated by unsupervised contrastive learning (21; 22), efforts in NLP are made to leverage unsupervised contrastive learning to recover sentence-level encoding abilities from the models (3; 47; 127; 128; 129).

However, translating the advancements in visual representation learning to learning sentence-level textual semantics presents unique challenges: a single augmentation (128; 129) might alter the meaning of a sentence, posing problems of the validity of the augmented sentence as a positive pair. Such attempts are primarily bottlenecked by the discreteness of subword units brought by tokenization (130), impeding the creation of continuous unsupervised semantic pairs that have preserved semantics through small perturbations to inputs.

Therefore, the most recognized unsupervised sentence representation learning method in NLP applies two dropout masks to the identical input to attain two representations, as positive pairs in contrastive learning (3). We argue that using identical inputs confines the method of (3) to essentially merely a way to improve

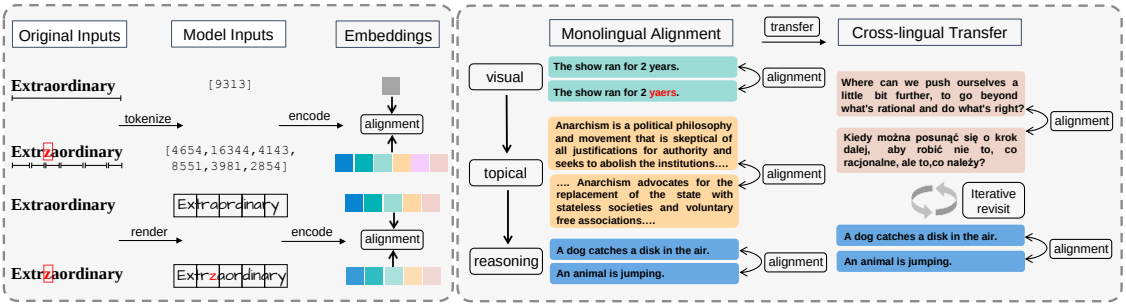


Figure 7.1: **Left:** Perceptual difference between tokenization-based language models and vision models, with the example of the word "extraordinar" with one single typo injected. **Right:** Our progressive pixel sentence representation learning framework.

uniformity (60) by distancing negative examples that are not identical to an instance itself, lacking the capability to provide signals towards non-verbatim examples, which are crucial for capturing semantically similar sentences.

Figure 7.1 encapsulates the difference between tokenization-based language models and vision models, on the perception of text. Using `bert-base-uncased` tokenizer (16), the word `extraordinary` is a standalone token [9313], while with one single typo `z` injected, `extrzaordinary` is tokenized into [4654, 16344, 4143, 8551, 3981, 2854], causing a large perceptual shift for the model. This mechanism has hindered traditional language models in recognizing that these minor textual perturbations do not fundamentally alter the underlying semantics. By contrast, the inherent continuity in visual models grants them less perceptual variance for textual perturbations. In parallel, we recognize that human understanding of text is not only visually grounded, but also tolerant of irregularities such as typos, varied word orders, and distorted text presentations (131; 132; 133; 134). Addressing these disadvantages, recent work proposed to model texts in the vision space, leading to work like PIXEL (38), which trained a foundation vision model to encode texts with reconstruction objectives to reconstruct masked rendered texts; and more recently, PIXAR and PTP (40; 41) with autoregressive losses. However, no works have studied using vision models to attain strong sentence and document-level textual semantics such that they can perform on-par with text models under STS, information retrieval (IR) or reasoning-as-retrieval (RAR)-style evaluation protocols used by text models.

Motivated by these observations, we propose a novel *pixel sentence representation*

*learning* framework that **redefines the learning of sentence and document-level textual semantics as a visual representation learning process**, taking the perceptually continuous advantages of vision models and closely mirroring the human cognitive processes. The approach diverges from traditional tokenization-based language models, allowing the models to effectively leverage the rich multi-modal semantic signals inherent in text, providing an alternative avenue for achieving a more natural understanding of textual semantics. Importantly, we also introduce the superiority of modeling sentence semantics in the pixel space to extrapolate and generalize to semantics of un-trained languages. The main contributions of this work are:

- We present and validate the potential of learning sentence and document-level semantics as a visual representation learning process, and design a progressive alignment scheme to facilitate the framework, providing $\sim$56 points sentence semantics gain over PIXEL base.
- Inspired by cognitive and linguistic sciences, we utilize typos and word-order shuffling as visually-grounded unsupervised augmentation methods, overcoming the challenges of applying perturbation augmentation methods in NLP due to discrete tokenization.
- We uncover a surprising leapfrogging pattern in pixel-based language models through iteratively training on OOD cross-lingual pairs and revisiting English NLI, showcasing an epiphany-like advancement in semantic understanding by "taking hints" across languages.
- We train and open-source **Pixel Linguist**, the first pixel model family that achieves on-par performance to conventional LMs in sentence-level semantics, providing the research community with an alternative avenue for achieving a more natural and intuitive understanding of textual semantics.

## 7.2 On the Behavioral Gap between Language Models and Pixel Models

As we leverage pure vision models to learn sentence and document-level text representation, it is essential to examine 1) the motivation behind this approach, and 2) the status quo of available techniques to facilitate the idea. In this section, we achieve this by conducting three experiments to study the behavioral differences between vanilla tokenization-based LMs and their pixel counterparts.

### 7.2.1 Preliminary

**Pixel Sentence and Document-level Representation Learning** We define pixel sentence and document-level representation learning as the process of understanding sentence and document-level text using vision models. The representations of sentences or documents can be used to reflect relative semantic relationships with one another, ideally with simple similarity matching without further projection, which approximates relative semantics drawn from the real-world distribution. Formally, given real-world pairwise data $\boldsymbol{i}, \boldsymbol{j}$ sampled from $\boldsymbol{p}_{\text{data}}$, we aim to minimize between the model's similarity perception of $\boldsymbol{i}, \boldsymbol{j}$, and their ground-truth relative semantics $\boldsymbol{s_{ij}}$.

$$\min_{\boldsymbol{f} \in \boldsymbol{\mathcal{F}}} \mathop{\mathbb{E}}_{\substack{\text{i.i.d.} \\ (\boldsymbol{i},\boldsymbol{j}) \sim \boldsymbol{p}_{\text{data}}}} \left[ \left( \boldsymbol{f}(\boldsymbol{x}_i)^\top \boldsymbol{f}(\boldsymbol{x}_j) - \boldsymbol{s}_{ij} \right)^2 \right] + \boldsymbol{\lambda} \sum_{i=1}^{N} \| \boldsymbol{f}(\boldsymbol{x}_i) \|_p^p , \tag{7.1}$$

where $\boldsymbol{x_i}$ is a text, $\boldsymbol{f} \in \boldsymbol{\mathcal{F}}$ is a vision model, making $\boldsymbol{f}(\boldsymbol{x_i})$ a visual representation of text.

This distinguishes our work from other lines of work, including Image Representation Learning with vision encoders (e.g., MOCO, SimCLR (21; 22)), Sentence Representation Learning with text encoders (e.g., SBERT, SimCSE (3; 18)), Image-Text Representation Learning with multi-modal encoders (e.g., CLIP (23)), and Image-Text Representation Learning with only vision encoders (e.g., CLIPPO (39)). The only work that fully aligns with our scope (text understanding with vision encoders, without non-text image signals) is PIXEL (38), and more recently PIXAR and PTP (40; 41). However, models like PIXEL are general-purpose vanilla models

Table 7.1: Representation Distance Shift, and Sentence-level Semantics Shift characterized by STS-b test performance.

| Model → | BERT | | PIXEL | |
|---|---|---|---|---|
| Perturbation ↓ | Rep. Shift | Semantics | Rep. Shift | Semantics |
| *Charcter-level* | | | | |
| Insertion | 0.049 | - 10.8↓ | 0.049 | 2.6↑ |
| Deletion | 0.040 | - 7.8↓ | 0.048 | - 0.4↓ |
| Substitution | 0.049 | - 10.3↓ | 0.038 | 0.0↔ |
| Neighbor Swap | 0.047 | - 11.6↓ | 0.014 | 0.3↑ |
| *Word-level* | | | | |
| Random Shuffle | 0.115 | - 19.1↓ | 0.087 | 4.6↑ |
| Condition Shuffle | 0.079 | - 17.6↓ | 0.048 | 2.1↑ |

like BERT (16), requiring fine-tuning to adapt to further downstream tasks. As we will show, the vanilla sentence and document-level representation provided by this backbone largely falls behind its NLP vanilla counterparts. In the following subsections, we present *three key observations* from our evaluations to highlight the behavioral differences between pixel models and their language model counterparts, providing insights that motivate our approach.

### 7.2.2 Observation 1: Robustness to Text Perturbations

We measure the behaviors of vanilla tokenization-based language models (16) and a pixel-based model (38) under text perturbations, drawing inspiration from human cognitive patterns.

We perform each perturbation outlined in Table 7.1 on all `sentence 1` in STS-b (13), and measure the embeddings' cosine distance shifted from the original emebeddings (*Rep. Shift*), and the degradation of relative semantics performance when evaluating the attacked sentences 1 with original sentences 2 (*Semantics*) on STS-b. Detailed descriptions of these perturbations are given in Section 7.3.1.

Not surprisingly, due to its tokenization dependency, BERT degrades under character-level attacks. However, the greatest degradation occurs when word order is shuffled randomly, showing the non-trivial contribution of positional embeddings in BERT. Conversely, PIXEL shows less semantic sensitivity, and even surprisingly attains semantics gain on STS-b in 5 out of 6 perturbed methods evaluated.

In conclusion, pixel-based language models exhibit significantly less sensitivity to

Table 7.2: Anisotropy Estimates (↓; the lower the better) of 10 languages.

| | en | de | nl | es | fr |
|---|---|---|---|---|---|
| BERT | **0.763** | 0.895 | 0.895 | 0.901 | 0.893 |
| PIXEL | 0.833 | **0.877** | **0.867** | **0.894** | **0.879** |
| | it | pt | pl | ru | zh |
| BERT | 0.900 | 0.891 | 0.914 | 0.942 | 0.909 |
| PIXEL | **0.884** | **0.874** | **0.866** | **0.917** | **0.880** |

visually-grounded textual perturbations (shown by character-level semantic shifts) than tokenization-based language models. They are also less sensitive to positions of words (word-level semantic shifts). This behavior of pixel models has granted us the natural convenience of using perturbed examples in constructing unsupervised contrastive learning pairs - as they are already perceived similar before training, and thus not detrimental to the models as positive pairs.

### 7.2.3 Observation 2: Potential for Zero-shot Cross-lingual Transferability

Pixel-based language models are tokenization-free and are intuitively ideal for cross-lingual transfer learning. We adopt a representation degeneration perspective (49; 50) to understand the zero-shot superiority of pixel language models in out-of-distribution (OOD) generalization. We measure the representation distribution of each language of the vanilla models using the multilingual STS-b (13; 135), which spans 10 languages from 4 language families.

The results in Table 7.2 reveal key insights. We encode sentence-level embeddings from the test set of each language with mean-pooling, and estimate the anisotropy by calculating the empirical mean of pairwise cosine similarity among these embeddings (49).

While BERT presents a slightly more isotropic pattern in its in-distribution language (en), all OOD languages (i.e., not seen during pre-training) suffer from severe representation degeneration. The advantage of PIXEL is immediately pronounced in OOD languages, with isotropy levels surpassing BERT. The robustness of PIXEL in maintaining consistent representation distribution across diverse languages, suggests

Table 7.3: Sentence semantics (↑; higher the better) of static word embeddings, BERT and its pixel-based counterpart.

| Model | GloVe | BERT | PIXEL |
|---|---|---|---|
| cls | - | **26.40** | 21.20 |
| mean | **58.02** | 52.59 | 16.28 |

that its semantic understanding at the sentence level is not solely reliant on language-specific features. Instead, PIXEL appears to leverage a more universal, shape-based approach to semantic cognition, suggesting a natural cognitive alignment with humans.

As we further explore (Section 7.5.2), when facilitated by contrastive learning, this alignment promises an amazingly strong bonding effect across languages, and provides a synergistic enhancement on unseen languages, evident in the model's zero-shot semantics understanding abilities.

### 7.2.4 Dilemma: Unsatisfactory Semantics (yet)

It has long been recognized that vanilla LMs are worse at capturing sentence-level semantics than simply averaging static word embeddings (18; 19), with a plethora of research dedicated to overcoming this (3; 35).

By measuring STS-b (13) performance, we show that, when pretrained on the same corpus with similar model architectures, the vanilla pixel counterpart (38) of BERT presents a sentence-level semantics even significantly inferior to the already subpar performance of BERT (16; 18).

Despite their robustness to text perturbations and potential in zero-shot cross-lingual transferability, which we have shown previously, we reveal the dilemma that vanilla pixel models lag behind their LM counterparts in capturing sentence-level semantics. Therefore, our main contribution is to fill this gap and get the best of both worlds.

### 7.2.5 Summary

Through the above three proof-of-concept experiments, we show that PIXEL has good robustness to perturbations and OOD-language isotropy compared to its tokenization counterpart BERT. However, the sentence-level semantics is much worse than BERT. The advantages and the dilemma motivate our progressive visual-topical-reasoning alignment framework introduced in this chapter, which aims to leverage visual text models' inherent advantages and enhance their sentence-level semantics, taking the best of both worlds.

## 7.3 Methods

As demonstrated in Section 7.2, pixel-based language models are less sensitive to perturbations and present a less severe representation degeneration in OOD languages. In this section, we present a novel pixel sentence representation learning framework designed upon these insights (Fig 7.1, right).

In the framework, a text $\boldsymbol{t}$ is rendered as an image using a rendering operator $\boldsymbol{R(t)}$, which is then encoded by a vision model $\boldsymbol{f(i)}$, i.e., $\boldsymbol{f(R(t))}$. Finally, we attain the a pooled embedding representing the input text using a pooling operator (e.g., mean pooling), $\boldsymbol{P(e)}$. The full encoding pipeline is denoted as $\boldsymbol{P(f(R(t)))}$. The parameters of the vision model $\boldsymbol{f(i)}$ is trained using its output embeddings as inputs to InfoNCE loss.

We propose 1) a progressive visual alignment (unsup.) - topical alignment (unsup.) - reasoning alignment (sup.) scheme for *monolingual learning*, and 2) an iterative cycle training scheme of revisiting OOD language pairs and English natural language inference pairs, for *cross-lingual transfer*.

### 7.3.1 Unsupervised Visual & Topical Alignment

Our unsupervised methods consist of a bag of visually-grounded language augmentation techniques for *visual alignment*, and a random-span sampling method for *topical alignment*. In the analysis section 7.6.2, we also show that traditional visual augmentation methods do not work for the task.

**Visually-grounded Language Augmentation** Our visually-grounded textual augmentation methods comprise of two classes, operating on word orders and typos, heavily inspired by linguistic cognitive studies (131; 132; 133; 134). The two methodologies and their intuitions are as follows.

**1) Word order shuffling** is an augmentation method inspired by the phenomenon that shuffled word orders do not affect making sense of the semantics much (131). For instance, in the sentence "This a is computer vision research paper", we as humans are able to make sense of the semantics, acting just as a bag-of-word model (65).

Here, we design two operations, *random shuffling*, and *conditional shuffling*. a) For random shuffling, we shuffle all the words in the same document, allowing words to go across different sub-sentences. b) For conditional shuffling, we impose that the first and the last word of a sub-sentence must stay in the original position, while allowing the rest of the words to shuffle within the sub-sentence.

**2) Typos** serve as another source of augmentation which humans have tolerance towards. Leveraging resources from taxonomy of adversarial text attack in NLP (136; 137), we incorporate 4 shape-based character-level typo attacks–*Character Insertion*, *Character Deletion*, *Character Substitution*, and *Character Neighbor Swap*– as augmentation methods. *Character Deletion* randomly deletes a character in a sample (e.g., `sentence` $\rightarrow$ `setence`). *Character Substitution* randomly replaces a character by either a character with a similar visual appearance (e.g, `a` and `e`), or a character that is next to the original character on the keyboard (e.g, `a` and `s`). *Character Neighbor Swap* randomly swaps the order of two adjacent characters (e.g., `random` $\rightarrow$ `radnom`).

The superiority of word and character-level perturbation, when combined with vision encoders, compared to combining with text encoders, is the **continuity**. Were to process the perturbed texts with language models, a small perturbation would even lead to being tokenized into extremely different sub-words, depriving the useful signals provided to the language models (Recalling Figure 7.1, left). We discuss other augmentation methods in Section 7.10.

Note that we posit such perturbation augmentations as **semantics-preserving** operations. We acknowledge they may still occasionally break semantics, but we

consider them acceptable compared to word-level augmentations used in previous tokenization-based unsupervised methods, such as random word deletion which is very likely to break semantics.

**Topical Alignment** Following Visual Alignment, we facilitate Topical alignment with random span sampling. We sample spans from the same document to form different views of the document. This method is inspired by unsupervised information retrieval learning (29; 138). In this work, we conduct two independent span sampling to a document, allowing overlapping spans from the same document. Intuitively, the overlapping part encourages model's perception about lexical matching, providing with vision models a bridge to pick up the part from the two sequences that have the same shape, as an anchor point to extrapolate the semantics of the remaining parts.

### 7.3.2 Supervised Reasoning Alignment

For supervised learning, we leverage paired language data. We use Natural Language Inference (NLI) datasets in the main experiments, including SNLI (24) and MNLI (25).

We then heavily explore cross-lingual zero-shot transfer, by collecting high-quality sentence pairs from parallel datasets, including Global Voice, MUSE, News Commentary, Tatoeba, Talks, WikiMatrix (139; 140; 141; 142; 143). We use English as an anchor and transfer NLI abilities to other languages using these paired language data as a bridge. We conduct iterative transfer detailed in the following section.

### 7.3.3 Learning Process

We use the standard InfoNCE loss (20) and make it symmetric as shown in Eq. 7.2, in line with CLIP (23). We find that a symmetric loss provides extra signals to the learning, resulting in faster and stabler convergence. The loss of a batch $\boldsymbol{\mathcal{L}}$ is defined as:

$$-\sum_{\boldsymbol{N}}(\log \frac{\exp(\boldsymbol{s_i} \cdot \boldsymbol{s_i^+})/\tau)}{\sum_{j=0}^{N} \exp(\boldsymbol{s_i} \cdot \boldsymbol{s_j})/\tau)} + \log \frac{\exp(\boldsymbol{s_i^+} \cdot \boldsymbol{s_i})/\tau)}{\sum_{j=0}^{N} \exp(\boldsymbol{s_i^+} \cdot \boldsymbol{s_j})/\tau)}), \tag{7.2}$$

with a batch of N sentences, each $\boldsymbol{s_i}$ normalized. Taking typo-perturbed texts as examples, the model is essentially trained to find, out of all the perturbed examples

in the batch, the one corresponding to the original text; and in original examples, the one corresponding to a perturbed text.

**Monolingual Progressive Learning** We first investigate monolingual learning in English. Motivated by our anisotropy estimates in Section 7.2.3 and the established understanding of the uniformity promise of contrastive learning (60), we employ a curriculum progressive scheme in displaying the training sets to the model. We start by presenting the easiest examples to the model, facilitating the learning of 1) an isotropic representation space; 2) robustness towards shape-based perburbations. The intuition here is that if we directly present the supervised training sets to the pretrained model, it has to learn isotropy, shape perception, and reasoning abilities simultaneously with a short training, hindering exploitation of the supervision signals. Therefore, our training follows a Visual Alignment - Topical Alignment - Reasoning Alignment progression.

**Cross-lingual Iterative Transfer** For Cross-lingual Transfer, we first conduct a small-scale experiment to understand the mutual transferability across the languages, by constructing and training on each bilingual pairs from multilingual STS-b, and evaluate on the rest (see Section 7.12 for details).

With these insights, we conduct 3 larger-scale transfers, by constructing parallel datasets of 10, 18, and 59 languages. We construct a mixed corpus $\boldsymbol{\mathcal{M}}$ by concatenating English NLI corpus $\boldsymbol{\mathcal{A}}$ and parallel language corpus $\boldsymbol{\mathcal{P}}$. Notably, each language is paired with English in $\boldsymbol{\mathcal{P}}$, because transferring English NLI ability to other languages intuitively requires using English as the anchor. Empirically, we find it surprisingly useful to iteratively go back-and-forth between $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{A}}$. And after each round that the model is trained on $\boldsymbol{\mathcal{P}}$, it learns better on $\boldsymbol{\mathcal{A}}$, shown by a general STS-b improvement on all languages, until convergence (Section 7.5.2).

We discuss the advantages of pixel-based methods for cross-lingual transferabilty as follows: 1) As pixel models do not involve tokenization, words from different languages are not tokenized into distinct tokens, but the visual shape similarity of their characters/subwords can be used to infer semantics. 2. In our experiments, we find a "cross-lingual leapfrogging" pattern unique to pixel models, where going

between English and other languages iteratively gradually enhances both at each iteration, showing that transferable visual hints across languages are leveraged to infer previously unknown words.

## 7.4 Experimental Settings

**Datasets** For *Visual Alignment*, we use the Wikipedia 1M dataset (3). We use the TextAttack framework (137) to construct character-level text-perturbed positive pairs and our own implementation for word-level perturbations. For *Topical Alignment*, we use a larger Wikipedia dump, in line with (29), as we speculate topical alignment to be more difficult, with a larger ratio of lexical mismatch between positive pairs (we construct ∼6.5M pairs leveraging the English Wikipedia dump from March 2022). For *Reasoning Alignment*, we use the concatenation of SNLI (24) and MNLI (25). This concatenated NLI collection is commonly referred to as all-NLI (∼314k entailment pairs). We only use entailment pairs as positive pairs, without leveraging contradiction as hard negative signals.

For cross-lingual main experiments, we first use the Ted parallel datasets introduced in (142) to construct $\boldsymbol{\mathcal{P}}_{10}$ and $\boldsymbol{\mathcal{P}}_{18}$, with ∼30-40k pairs per language. With the insights, we provide a XL-scale training with the collection of Global Voice, MUSE, News Commentary, Tatoeba, Talks, WikiMatrix (139; 140; 141; 142; 143), resulting in 59 languages, pushing the limit of cross-lingual performance.

**Evaluation** Our extensive evaluation spans across 7 semantic textual similarity (STS) tasks, 5 information retrieval (IR) tasks, and 6 reasoning tasks. For STS tasks, we include STS 2012 - STS 2016, STS-b and SICK-R (8; 10; 11; 12; 13; 72; 144). For IR tasks, we include Natural Questions, HotpotQA, Scidocs, Scifact, and NFcorpus (145; 146; 147; 148; 149) from BEIR (14). For reasoning tasks, we include PIQA, SIQA, Winogrande, AlphaNLI, Hellaswag, and TempReason-L1 (90; 92; 93; 94; 95; 96) from RAR-b (37), a novel benchmark that evaluates reasoning-level language understanding abilities of embedding models through re-framing reasoning as retrieval.

**Implementation Details** We initialize our models with PIXEL (38), which uses a ViT-MAE (150) backbone and objective, and is pre-trained on the same corpus as BERT (16), with the training set rendered into images. We use PangoCairo (151) to render texts into images on-the-fly. In most experiments, we train the models for 1 epoch, with a learning rate of 3e-6 in visual alignment step and 3e-5 otherwise, a temperature $\boldsymbol{\tau}$ of 0.05, and a max sequence (patch) length of 64. We use a batch size of 768 for visual and topical alignment step, and 128 for reasoning alignment. Mean-pooling is used in all experiments as [cls] token of the base model is under-trained (Section 7.15). We normalize the final embeddings before loss computation and in inference time, and find it highly beneficial to model convergence, which is consistent with prior work (22; 60; 152). We let the model go through all training data in early stages (visual and topical alignment), and take the best checkpoint on STS-b when finally optimizing it on the reasoning alignment step (see Section 7.11 for empirical evidence).

## 7.5 Results

In this section, we first present the results of monolingual learning facilitated by our visual-topical-reasoning alignment progression, where we characterize the importance of shape-based perturbation pretraining. We then present results of cross-lingual learning further facilitated by an iterative transfer from English to other languages, where a surprising "linguistic leapfrogging" pattern is revealed.

**Main Empirical Results** Table 7.4 shows that Pixel Linguist brings huge performance gain to the original PIXEL models on STS tasks, achieving state-of-the-art performance compared to methods training PIXEL's language model counterpart BERT, including BERT-flow (6), BERT-whitening (5), and Sentence BERT (18). Table 7.5 presents Pixel Linguist's performance on information retrieval tasks using the BEIR benchmark. We ablate the contribution brought by each of our progressive training step, collectively outperforming SimCSE (3). Lastly, Table 7.6 shows that with proper fine-tuning on a small reasoning training set, Pixel Linguist is able to achieve SOTA performance on RAR-b, outperforming state-of-the-art LM-based

Table 7.4: Performance on STS tasks (Spearman's correlation). ♠: Our recipes.

| | STS average | Gain over vanilla | SICK-R | STS12 | STS13 | STS14 | STS15 | STS16 | STS-B |
|---|---|---|---|---|---|---|---|---|---|
| PIXEL-base$_{\text{mean-pooled}}$ | 19.97 | - | 25.28 | 31.73 | 19.81 | 15.30 | 13.57 | 17.80 | 16.28 |
| + allNLI (♠) | 74.28 | 54.31 | 70.94 | 71.23 | 74.25 | 70.75 | 80.45 | 74.82 | 77.50 |
| + SimCSE unsup (3) + allNLI | 73.57↓ | 52.99↓ | 70.52↓ | 71.17↓ | 73.69↓ | 70.65↓ | 80.34↓ | 75.05↓ | 77.17↓ |
| + character typo + allNLI (♠) | 74.64 | 54.67 | 70.71 | 71.71 | 75.29 | 71.55 | 80.46 | **75.05** | 77.71 |
| + word con. shuffle + allNLI (♠) | 74.77 | 54.80 | 70.67 | 71.76 | 75.73 | 71.43 | 80.72 | 75.05 | 78.02 |
| Pixel-Linguist-all (♠) | **75.62** | **55.65** | **72.00** | **75.81** | **76.06** | **71.65** | **81.21** | 74.12 | **78.49** |
| BERT-base-cased $_{\text{mean-pooled}}$ (153) | 56.59 | - | 60.06 | 38.49 | 62.74 | 53.30 | 64.56 | 64.43 | 52.58 |
| BERT-base-uncased$_{\text{mean-pooled}}$ (153) | 52.64 | - | 58.65 | 30.87 | 59.90 | 47.73 | 60.29 | 63.73 | 47.29 |
| + BERT-flow (6) | 66.55 | 13.92 | 64.47 | 58.4 | 67.1 | 60.85 | 75.16 | 71.22 | 68.66 |
| + BERT-whitening (5) | 66.28 | 13.65 | 63.73 | 57.83 | 66.9 | 60.9 | 75.08 | 71.31 | 68.24 |
| + Sentence BERT (18) | 74.88 | 22.24 | 73.06 | 70.97 | 76.53 | 73.20 | 79.09 | 74.30 | 76.98 |

embedding models that are trained on massive paired datasets, even outperforming or matching OpenAI-Embedding-3. Lastly, Pixel Linguist shows great cross-lingual transferability, bringing enhanced performance at every iterative step in our designed iterative cross-lingual training framework.

Table 7.5: Information Retrieval Results (nDCG@10) on BEIR

| | Nfcorpus | Scidocs | Scifact | NQ | HotpotQA | Geomean | Average |
|---|---|---|---|---|---|---|---|
| *ours* | | | | | | | |
| + allNLI | 0.133 | 0.063 | 0.231 | 0.143 | 0.212 | 0.156 | 0.142 |
| + visual + allNLI | 0.148 | 0.069 | 0.237 | 0.148 | 0.243 | 0.169 | 0.154 |
| + visual + topical + allNLI | **0.154** | **0.071** | **0.265** | **0.175** | **0.257** | **0.184** | **0.167** |
| + visual + topical + MSMARCO | **0.219** | **0.104** | 0.373 | 0.236 | **0.375** | 0.261 | **0.237** |
| + visual + topical + (MSMARCO + allNLI) | 0.214 | 0.099 | **0.384** | **0.256** | 0.353 | **0.261** | 0.236 |
| SimCSE-BERT-supervised (3) | 0.124 | 0.075 | 0.296 | 0.161 | 0.229 | 0.177 | 0.159 |

Table 7.6: Reasoning results (nDCG@10) on RAR-b benchmark

| | alphanli | hellaswag | siqa | piqa | winogrande | TR1 | Geomean | Average |
|---|---|---|---|---|---|---|---|---|
| **Pixel-linguist-rar (ours)** | 0.230 | 0.158 | **0.051** | 0.159 | **0.690** | **0.021** | **0.128** | **0.218** |
| Contriever (154) | **0.318** | 0.144 | 0.013 | 0.246 | 0.471 | 0.019 | 0.105 | 0.202 |
| all-mpnet-base-v2 (18) | 0.224 | 0.263 | 0.024 | **0.290** | 0.207 | 0.018 | 0.107 | 0.171 |
| BGE-base (113) | 0.110 | **0.266** | 0.009 | 0.257 | 0.138 | 0.015 | 0.072 | 0.133 |
| OpenAI-embedding-3-large | 0.373 | 0.341 | 0.034 | 0.420 | 0.291 | 0.021 | 0.149 | 0.247 |

### 7.5.1 English-only Evaluation Results

**Semantic Textual Similarity** Table 7.4 presents the results of 7 STS tasks. We find a consistent gain by first pre-training on our text perturbation augmentations outlined in our Visual Alignment step, before optimizing on NLI datasets (the character typo & word con. shuffle settings in Table 7.4). The advantage is more pronounced by comparing to training the same base model with unsupervised-SimCSE (3) before

allNLI, which degrades model's performance in our case. While the unsupervised SimCSE primarily enforces a uniform representation distribution by pushing away negative pairs, chiming in with the unifomrity promise of contrastive loss (60), our perturbation methods provide additional shape-based signals to visually facilitate the alignment promise. We also present several most-recognized NLP baselines initialized with BERT, including SBERT (18), BERT-whitening (5), and BERT-Flow (6). We highlight that we achieve this performance with a vanilla checkpoint with subpar semantic capabilities, as the vanilla PIXEL is 36 absolute points behind vanilla BERT in sentence-level semantics (Section 7.2.4). We also evaluate our Pixel-Linguist-all model, which has undergone the full progressive recipe and cross-lingual transfer (Section 7.5.2 further reveals the linguistic leapfrogging pattern brought by cross-lingual training). Results on IR in the following paragraph and Section 7.6.1 ablate the benefits contributed by each step.

**Information Retrieval** Table 7.5 outlines results on 5 information retrieval tasks. The upper part of the table characterizes the importance of each step in our main recipes in contributing to a higher retrieval performance. With a much under-trained base model, we outperform supervised SimCSE (3) with BERT with the same allNLI supervision. In the middle part, we additionally show that, by adding the commonly-adopted MSMARCO training (67), the retrieval results are largely enhanced.

**Reasoning** We move beyond the established evaluation on STS and IR tasks, and understand the potential to achieve reasoning-level language encoding abilities with Pixel Linguist. We adopt the recent RAR-b framework (37) of re-framing reasoning tasks as retrieval tasks, i.e., whether embedding models can search the correct answer out of a corpus that is intentionally made large.

By sampling only around 166k data from training sets of the tasks and concatenating with allNLI (314k+166k=480k), we manage to achieve better performance compared to state-of-the-art LM-based embedding models of our size, which are trained on massive paired datasets, which are orders of magnitudes larger than ours (For example, all-mpnet-base-v2 is trained on 1B pairs).

Table 7.7: Cross-lingual Results across iterative training. $\mathcal{P}_{10}$, $\mathcal{P}_{18}$ and $\mathcal{P}_{\text{XL}}$ have 10, 18, and 59 languages including English, respectively. Numberdenotes best across all settings;numberdenotes second best. **Bold** denotes the best in its own setting.

| Eval (→) Train (↓) | | en | de | nl | es | fr | it | pt | pl | ru | zh | Germanic (en, de, nl) | Romance (es, fr, it, pt) | Slavic (pl, ru) | Sinitic (zh) | avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\mathcal{P}_{10}$ + $\mathcal{A}$ | **Iter.1** | 79.46 | 62.69 | 64.89 | 67.47 | 68.02 | 67.48 | 65.97 | 64.81 | 60.42 | 42.17 | 69.01 | 67.24 | 62.61 | 42.17 | 64.34 |
| | **Iter.2** | 79.85 | 64.61 | 66.09 | 68.48 | 69.48 | 68.97 | 67.02 | 65.97 | 62.04 | 47.32 | 70.19 | 68.49 | **64.00** | 47.32 | 65.98 |
| | **Iter.3** | 79.53 | 66.12 | 66.50 | 69.25 | 69.72 | 68.90 | 67.51 | 65.87 | 61.74 | 50.11 | **70.72** | **68.85** | 63.80 | **50.11** | **66.53** |
| $\mathcal{P}_{18}$ + $\mathcal{A}$ | **Iter1** | 80.04 | 61.92 | 65.83 | 66.64 | 68.07 | 65.75 | 65.01 | 62.94 | 59.58 | 46.58 | 69.26 | 66.37 | 61.26 | 46.58 | 64.24 |
| | **Iter2** | 80.09 | 63.49 | 66.28 | 66.87 | 68.84 | 65.93 | 65.52 | 65.05 | 62.24 | 49.10 | 69.95 | 66.79 | 63.65 | 49.10 | 65.34 |
| | **Iter3** | 79.96 | 64.59 | 66.30 | 67.46 | 69.39 | 65.68 | 65.84 | 65.56 | 62.00 | 50.57 | **70.28** | **67.09** | **63.78** | **50.57** | **65.74** |
| $\mathcal{P}_{\text{XL}}$ | **Iter3** | 78.79 | 68.15 | 67.60 | 70.31 | 70.84 | 68.92 | 69.72 | 66.39 | 66.76 | 52.92 | 71.51 | 69.95 | 66.57 | 52.92 | 68.04 |

### 7.5.2 Cross-lingual Transfer Results

Table 7.7 presents our findings on Cross-lingual Transfer, measured through performance on Multilingual STS-b. We conduct iterative training going back and forth parallel datasets and English NLI. The number reported for each iteration is after re-optimization on NLI again after "cross-lingual exploratio". We have three settings, $\mathcal{P}_{10}$, $\mathcal{P}_{18}$ and $\mathcal{P}_{\text{XL}}$, with 10, 18 and 59 languages respectively.

**Linguistic Leapfrogging** We observe a "leapfrogging" pattern in our cross-lingual learning process. In our monolingual training, the model's English STS-b performance is capped at around 78 regardless how much extra pre-training we do before fine-tuning it on English all-NLI, which we attributed largely to the limitations of the pretrained checkpoint as discussed earlier. However, after further training on English-other language pairs and revisiting all-NLI, the model's English STS-b performance surprisingly exceeds this plateau, achieving a performance surpassing 80. This clearly reveals that models have learned to infer on the semantics of English through finding helpful "visual hints" during training on other languages. We envision that the discovered leapfrogging pattern suggests that pixel-based language models may require less multilingual pretraining to achieve state-of-the-art performance previously attained by existing conventional LMs.

**Visualizing Language Alignment** In Figure 7.2, we visualize language alignment at the end of our iterative training. Initially, languages occupy distinctive subspaces in the vanilla model (Fig 7.2, left 1). After contrastive alignment training, all languages overlap in a larger shared space (Fig 7.2, left 2). This alignment facilitates "bonding" across languages: as shown in Fig 7.2, left 3, we visualize the mutual growth

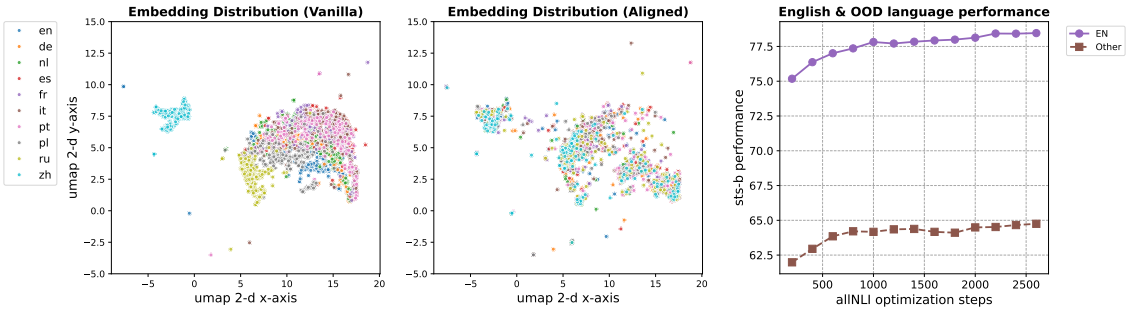


Figure 7.2: **Left 1-2:** Embedding Distribution of the vanilla model and model after three rounds of iterative alignment. **Left 3:** English and out-of-distribution (OOD) language performance during the final optimization of allNLI. After alignment, English and other languages exhibit a bonding effect.

of all language semantics in the final optimization round on English all-NLI. The well-aligned subspaces in parallel training ensure that when optimizing on English, all languages benefit and optimize together without direct optimization. Note that UMAP learns to preserve relative distance from the high-dimensional space in the low-dimensional space, and thus provides reliable results for us to interpret geometric relationship across language embeddings. We use 1000 examples per language to train this UMAP projector such that it is more robust, and transform and visualize only a subset here for better visibility.

The leapfrogging performance found in Table 7.7 and the overlapped cross-lingual embedding space shown in Figure 7.2 collectively suggest that the cross-lingual embedding space is bonded visually, providing strong transferability across languages by leveraging visual hints to infer semantics of previous unknown words by using learned semantics from other languages.

## 7.6 Analysis

In this analysis section, we first present an ablation of the benefits brought by each of our progressive training step; we then show that, traditional visual augmentation methods could not replace our visually-grounded textual augmentation for learning sentence representation. Lastly, we experiment with subword, character-level, byte-level language models, showing the versatility of our methods.

### 7.6.1 Ablation on Progressive Learning

Table 7.8: Monolingual Training Ablation

| Metrics→ | English | | OOD Languages (9) | |
|---|---|---|---|---|
| Model | STS-b (↑) | Ani. (↓) | STS-b (↑) | Ani. (↓) |
| $\text{PIXEL}_{\text{vanilla}}$ | 16.28 | 0.833 | 22.50 | 0.882 |
| + allNLI | 77.50 | 0.023 | 48.87 | 0.529 |
| + Character-level Typos | 52.14 | 0.190 | 47.33 | 0.444 |
| + allNLI | <u>77.71</u> | 0.021 | 49.20 | 0.508 |
| + Word Cond. Shuffling | 56.78 | 0.116 | 43.88 | 0.410 |
| + allNLI | **78.02** | 0.022 | 48.73 | 0.535 |
| + Word Rand. Shuffling | 60.61 | 0.169 | 49.69 | 0.450 |
| + allNLI | 77.40 | 0.022 | 48.55 | 0.508 |
| + Ensemble | 59.39 | 0.145 | <u>51.34</u> | 0.441 |
| + Ensemble + WikiSpan | 52.74 | 0.320 | 40.01 | 0.552 |
| + allNLI | 77.78 | 0.026 | **51.99** | 0.525 |
| $\text{BERT}_{\text{vanilla}}$ | 52.59 | 0.763 | 43.11 | 0.904 |
| + SBERT (18) | 76.98 | 0.174 | 44.69 | 0.712 |

Table 7.8 ablates each step in the progressive learning on STS-b. We find a consistent gain by first pre-training on our text perturbation augmentations in Visual Alignment step before optimizing on NLI.

We find that taking an ensemble of all unsupervised checkpoints improves OOD generalization, and is further enhanced by topical alignment (denoted WikiSpan). We find pretraining on WikiSpan essential to trigger exceptional OOD generalization after fine-tuning on allNLI, providing a good initialization for downstream cross-lingual transfer.

### 7.6.2 Do Standard Visual Augmentations Work?

With the performance gain of our visually-grounded text-based augmentation, it is a natural question whether standard visual augmentation methods could replace our methods. We experiment with four visual augmentation methods, cropping, horizontal flip, vertical flip, and Gaussian blurring (22) (Table 7.9) , and show that traditional methods lead to a severe semantic collapse, and it is not recoverable with NLI, largely falling behind our textual perturbations.

Table 7.9: Traditional Visual augmentation.

| Model | STS-b (↑) | Anisotropy (↓) |
|---|---|---|
| $\text{PIXEL}_{\text{vanilla}}$ | 16.28 | 0.833 |
| + Cropping | 4.07 | 0.187 |
| + allNLI | 77.11 | 0.216 |
| + Horizontal Flip | 5.97 | 0.218 |
| + allNLI | 77.19 | 0.211 |
| + Vertical Flip | 24.75 | 0.650 |
| + allNLI | 76.99 | 0.213 |
| + Gaussian Blurring | 23.59 | 0.560 |
| + allNLI | 76.98 | 0.220 |
| $\text{Pixel-Linguist}_{\texttt{en}}$-best | 16.28 (vanilla) → 78.02 (ours) | |
| $\text{Pixel-Linguist}_{\texttt{all}}$-best | 16.28 (vanilla) → 80.09 (ours) | |

Table 7.10: LMs with our methods.

| Model | BERT | mBERT | CANINE | ByT5 |
|---|---|---|---|---|
| Vanilla | 52.59 | 50.98 | 48.36 | 22.12 |
| + allNLI | 80.59 | 78.01 | 68.29 | 67.43 |
| + Character-level Typos | 62.16 | 61.88 | 50.71 | 47.01 |
| + allNLI | 80.93 | 78.14 | 67.84 | 67.43 |
| + Word Con. Shuffling | 71.99 | 68.61 | 36.91 | 39.65 |
| + allNLI | 81.24 | 78.72 | 68.25 | 68.63 |
| + Word Rand. Shuffling | 73.81 | 66.54 | 43.33 | 34.63 |
| + allNLI | 81.50 | 78.34 | 69.25 | 70.05 |
| + Text Pert. Mixture | 46.78 | 46.58 | 47.46 | 46.08 |
| + allNLI | 81.12 | 78.21 | 69.26 | 69.64 |
| $\text{Pixel-Linguist}_{\texttt{en}}$-best | 16.28 (vanilla) → 78.02 (ours) | | | |
| $\text{Pixel-Linguist}_{\texttt{all}}$-best | 16.28 (vanilla) → 80.09 (ours) | | | |

### 7.6.3 Do tokenization-based models work?

We experiment with whether our perturbation-based methods apply to language models using tokenization. We consider subword, character, and byte language models, including BERT (16), mBERT (16), Canine (155), and ByT5 (156).

As shown in Table 7.10 (see Section 7.14 for technical details), our unsupervised methods unexpectedly do work for conventional LMs, showing the versatility of our method. While we believe in pixel models, we encourage future work to understand this pattern.

## 7.7 Limitations

As discussed throughout the paper, the base model we leverage largely falls behind conventional LMs in sentence-level semantics. We see our visual and topical alignment steps as a continual pretraining process to make up the subpar performance of the

base model, which introduces certain computational overheads. Additionally, having revealed the cross-lingual leapfrogging phenomenon, it is intuitive that a cross-lingual pretrained checkpoint will further enhance the performance. However, we are not able to pretrain from scratch due to resource constraints. We envision the advancement of this research line pioneered by our method will greatly benefit from better-pretrained checkpoints.

Further, there is a computational overhead for rendering texts into images. Depending on the CPU used and batch size of encoding, we observe a 10%-12% computational overhead for the basic sequential rendering-encoding setting. However, we find that by rendering the next batch when encoding the current batch, we are able to drop the overhead to $\sim$0% (the only bottleneck now is rendering for the first batch). Moreover, we are not yet able to evaluate on very long documents due to the limited context size of an image (the number of words that are able to fit in an image), although we indeed evaluated on document-level texts through BEIR evaluation. Last, the augmentations are restricted to visually-grounded textual augmentations and do not explore other NLP-style augmentations such as synonym replacement etc.

## 7.8 Conclusion

We proposed a novel pixel sentence representation learning framework, facilitated by the visual-topical-reasoning alignment scheme. We demonstrated model's shape perception injected by our visually grounded textual perturbations. We uncovered a "leapfrogging" pattern, where learning across languages enhances the model's understanding on each language individually. These patterns validate that modeling textual semantics in the pixel space, utilizing shape-based information inherent in text, is a promising path for learning stronger and more human-like sentence encoders.

We also see how findings and knowledge we attained in earlier chapters contribute to the understanding of this chapter. For instance, how the measurement of isotropy we introduced in Chapter 4 provides an important lens to understand the perceptual

robustness and cross-lingual transferability potential, for visual text models. This chapter also partly answers the question in posed in Chapter 6 - whether grounding in more modalities can potentially improve representations. Findings in this chapter provide motivations for the following section, where we provide a more holistic evaluation of multimodal representations beyond just visual texts.

## 7.9 Computational Cost

We use A100 and 3090 GPUs for all experiments. The visual and topical alignment step experiments are all run on A100 because of the need for a large batch size for these two steps. All other experiments are run either on A100 or 3090 GPUs. The visual alignment step takes around 1.5 hours; the topical alignment step takes around 6 hours; and one reasoning alignment step in the main experiments takes around 0.5 hours. One round of largest cross-lingual transfer takes around 24 hours on 1 A100, and our main cross-lingual checkpoint iteratively goes back and forth cross-lingual transfer and English reasoning revisiting for 3 times, which takes $\sim$3.5 days including its English training.

## 7.10 Other augmentation techniques

Notably, to explore the limit of unsupervised language semantics acquisition solely dependent on visual learning, we only consider **visually-grounded text perturbation methods**. Therefore, even though there are many text augmentation methods, such as back translation (157), and synonym replacement (158), we do not consider them as "unsupervised vision methods".

## 7.11 Checkpoint Selection

Empirically, we find that checkpoints that display good semantic performance in earlier stages do not necessarily provide best potential in later supervised training, exhibiting certain early-phase overfitting. The pattern might also be highly relevant to the performance orthogonality between STS and retrieval tasks (31; 68; 74), with

the latter can be highly optimized through our topical alignment step while the former relies on the transferability provided by NLI data. Therefore, we let the models go through all training data in early stages, but take the best checkpoint on STS-b when finally optimizing it on all-NLI.

## 7.12 Cross-lingual Small-scale Experiments

Table 7.11: Bilingual small-scale transfer Results. It presents the zero-shot transferability by training small-scale on each bilingual pair ({en, 1 other language}).

| Eval (→) / Train (↓) | | en | de | nl | es | fr | it | pt | pl | ru | zh | Germanic (en, de, nl) | Romance (es, fr, it, pt) | Slavic (pl, ru) | Sinitic (zh) | avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\mathcal{A}$ + $\mathcal{S}_n$ | **de** | 77.40 | 58.45 | 59.44 | 58.53 | 60.23 | 58.95 | 58.63 | 50.31 | 41.35 | 23.51 | **65.09** | 59.09 | 45.83 | 23.51 | **54.68** |
| | **nl** | 77.63 | 57.45 | **59.65** | 55.55 | 59.21 | 57.49 | 56.81 | 49.26 | 38.39 | 23.62 | 64.91 | 57.27 | 43.82 | 23.62 | 53.51 |
| | **es** | 77.48 | 58.75 | 55.39 | **59.45** | 59.77 | 56.32 | 58.22 | 48.97 | 39.13 | 23.06 | 63.87 | 58.44 | 44.05 | 23.06 | 53.65 |
| | **fr** | 77.50 | **59.00** | 55.61 | 58.67 | **63.09** | 58.88 | 57.83 | 49.95 | 38.46 | **24.73** | 64.04 | **59.62** | 44.21 | **24.73** | 54.37 |
| | **it** | 77.61 | 58.35 | 55.85 | 57.46 | 59.96 | **60.39** | 56.26 | 49.42 | 38.34 | 22.85 | 63.93 | 58.52 | 43.88 | 22.85 | 53.65 |
| | **pt** | 77.41 | 58.85 | 55.47 | 58.26 | 59.13 | 58.07 | **61.03** | 48.72 | 36.02 | 23.70 | 63.91 | 59.12 | 42.37 | 23.70 | 53.67 |
| | **pl** | 76.98 | 57.61 | 55.17 | 54.81 | 58.40 | 58.34 | 56.74 | 50.95 | 37.30 | 20.50 | 63.25 | 57.07 | 44.13 | 20.50 | 52.68 |
| | **ru** | 75.87 | 58.30 | 55.73 | 54.87 | 57.69 | 57.57 | 56.07 | 49.97 | 43.12 | 21.75 | 63.30 | 56.55 | 46.54 | 21.75 | 53.10 |
| | **zh** | 75.42 | 57.05 | 55.05 | 55.49 | 57.29 | 57.10 | 55.02 | **51.50** | **43.24** | 19.42 | 62.50 | 56.23 | **47.37** | 19.42 | 52.66 |

Table 7.11 presents the zero-shot transferability by training small-scale on each bilingual pair, by pairng only 5k {en, 1 other language} from multilingual STS-b. This gives us a glimpse of the visually-grounded understanding that learning a certain language will bring to other languages.

Specifically, the datasets are created by concatenating English NLI dataset $\boldsymbol{\mathcal{A}}$ with STS-b bilingual pair $\boldsymbol{\mathcal{S}_n}$, where $\boldsymbol{n}$ is each of the language from {en, de, nl, es, fr, it, pt, pl, ru, zh}. The findings are summarized as follows: 1) In general, training on $\boldsymbol{\mathcal{S}_n}$ provides zero-shot generalization on languages from the same language family. For instance, training on $\boldsymbol{\mathcal{S}}_{\text{German}}$ (de) provides good STS-b performance on Dutch. And the same applies to languages among the Romance family.

2) The only outlier is $\boldsymbol{\mathcal{S}}_{\text{Chinese}}$ (zh), which surprisingly hurts its own semantic performance, but provides good transfer for Russian and Portuguese, which we hypothesize to be due to its shape's being too out-of-distribution in pre-training, and thus being extremely unstable when getting aligned using few paired examples.

## 7.13 Information Retrieval Results Further Analysis

We implement our evaluation of information retrieval with the BEIR framework (14). We wrap the pixel encoders into the framework.

Here, we provide extra analysis of our intermediate checkpoints that have not been optimized on allNLI on Natural Questions (145), considering it is general-domain task, it is large-scale, and commonly used to evaluate retriever models. The performance is quantified by nDCG@n, and recall@n, which respectively concern (**nDCG@n:**) how well the top $\boldsymbol{n}$ retrieved documents are ranked, and (**recall@n:**) if the ground-truth documents are even successfully retrieved into top $\boldsymbol{n}$.

| Method | nDCG@1 | nDCG@10 | recall@1 | recall@10 |
|---|---|---|---|---|
| + visual | 0.012 | 0.026 | 0.011 | 0.043 |
| + visual + wikispan | 0.002 | 0.003 | 0.002 | 0.004 |
| + allNLI | 0.076 | 0.143 | 0.068 | 0.224 |
| + visual + allNLI | <u>0.082</u> | <u>0.148</u> | <u>0.075</u> | <u>0.231</u> |
| + visual + wikispan + allNLI | **0.091** | **0.175** | **0.078** | **0.283** |

Table 7.12: Information Retrieval Results on Natural Question

The results are presented in Table 7.12. The findings are interesting: 1) without ending in allNLI, training wikispan after visual alignment actually degrades the model's performance. This is in contrast to findings of (29) in NLP, who find that training with Wikipedia spans provides strong IR performance. 2) However, the advantage of Wikispan is immediately pronounced when the model is further trained with allNLI, outperforming other training orders. Therefore, it can be concluded that training on Wikispan does enhance retrieval performance, but with pixel models, Wikispan acts more like a pretraining task, and its advantages need to be triggered with supervised data like allNLI, or supervised IR datasets.

## 7.14 Implementation of LM ablation

We use mean-pooling to attain sentence representation for all language model ablation experiments, aligning with pixel experiments. For ByT5 (156), we use the T5 encoder only, aligning with previous works in sentence-level semantics.

## 7.15 Pooling Mode

As mentioned in implementation details, we use mean-pooling (instead of `[cls]` pooling). This is because we find that the `[cls]` token in PIXEL, though present, is not leveraged in the model's pretraining objectives, and thus present also perfect anisotropy (Table 7.13), with two randomly-sampled sentences presenting a cosine similarity of 0.999. By contrast, BERT's NSP objective in pretraining more or less pushes `[cls]` embeddings of different sentences apart. Based on these behaviors, we choose to apply mean-pooling to PIXEL.

| Pooling→ | [CLS]-pooling | | Mean-pooling | |
|---|---|---|---|---|
| Language↓ | BERT | PIXEL | BERT | PIXEL |
| en | **0.926** | 0.999 | **0.763** | 0.833 |
| de | **0.971** | 0.999 | 0.895 | **0.877** |
| nl | **0.965** | 0.999 | 0.895 | **0.867** |
| es | **0.974** | 0.999 | 0.901 | **0.894** |
| fr | **0.970** | 0.999 | 0.893 | **0.879** |
| it | **0.970** | 0.999 | 0.900 | **0.884** |
| pt | **0.968** | 0.999 | 0.891 | **0.874** |
| pl | **0.975** | 0.999 | 0.914 | **0.866** |
| ru | **0.977** | 0.999 | 0.942 | **0.917** |
| zh | **0.963** | 0.999 | 0.909 | **0.880** |

Table 7.13: Anisotropy Estimates (↓ the better) with different pooling modes.

# CHAPTER 8

# Towards Holistic Multimodal Representation Evaluation through a Massive Benchmark

This chapter introduces MIEB (Massive Image Embedding Benchmark), a large-scale benchmark for evaluating image and image-text embeddings. In previous chapters, we have built a solid understanding of embeddings, and are also motivated to think whether grounding in more modalities raises the upper bound of representation-generation alignment. Building on these, this chapter aims to holistically evaluate multimodal embedding models across tasks, beyond visual text evaluation in Chapter 7. Through this chapter, the core research questions we aim to answer: how to holistically evaluate embedding models across tasks/languages and what trade-offs different model families exhibit.

Image representations are often evaluated through disjointed, task-specific protocols, leading to a fragmented understanding of model capabilities. For instance, it is unclear whether an image embedding model adept at clustering images is equally good at retrieving relevant images given a piece of text. We introduce the Massive Image Embedding Benchmark (MIEB) to evaluate the performance of image and image-text embedding models across the broadest spectrum to date. MIEB spans 38 languages across 130 individual tasks, which we group into 8 high-level categories.

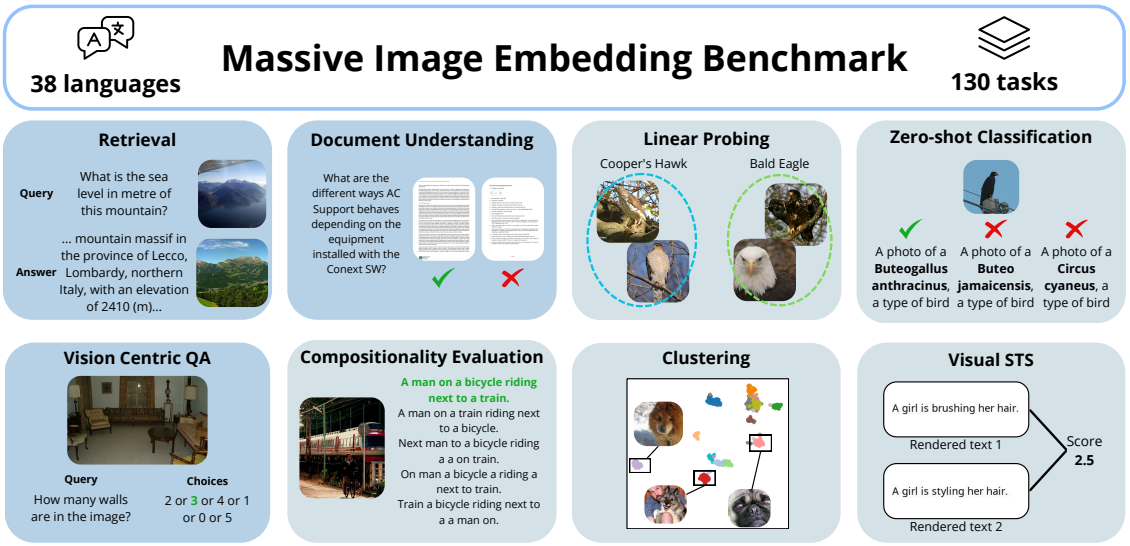


Figure 8.1: **Overview of MIEB task categories with examples.** See Table 8.1 for details about capabilities measured and other information.

We benchmark 50 models across our benchmark, finding that no single method dominates across all task categories. We reveal hidden capabilities in advanced vision models such as their accurate visual representation of texts, and their yet limited capabilities in interleaved encodings and matching images and texts in the presence of confounders. We also show that the performance of vision encoders on MIEB correlates highly with their performance when used in multimodal large language models. Our code, dataset, and leaderboard are publicly available at `https://github.com/embeddings-benchmark/mteb`.

## 8.1 Introduction

Image and text embeddings power a wide range of use cases, from search engines to recommendation systems (159; 160; 161). However, evaluation protocols for image and multimodal embedding models vary widely, ranging from image-text retrieval, zero-shot classification (23; 44), linear probing (23; 42), fine-tuning the models (21; 22), and using MLLM performance as proxies (1). These divergent protocols reveal the lack of standardized criteria for assessing image representations.

We introduce the Massive Image Embedding Benchmark (MIEB) to provide a unified comprehensive evaluation protocol to spur the field's advancement toward

universal image-text embedding models. We build on the standard for the evaluation of text embeddings, MTEB (74), extending its codebase and leaderboard for image and image-text embedding models. MIEB spans 130 tasks grouped into 8 task categories: Aligning with MTEB, we integrate **Clustering**, **Classification**, and **Retrieval**. Notably, we consider fine-grained aspects, such as *interleaved retrieval*, *multilingual retrieval*, *instruction-aware retrieval*. We additionally include **Compositionality Evaluation** and **Vision Centric Question Answering**, respectively assessing nuanced information encoded in embeddings and their capabilities in solving vision-centric QA tasks. We focus on tasks that require strong *visual understanding of texts*, for which we include **Visual STS**, the visual counterpart of semantic textual similarity in NLP, and **Document Understanding**, assessing the vision-only understanding of high-resolution documents with dense texts and complex layout, enabling evaluation that pushes forward the development of natural interleaved embeddings.

Our analysis across task categories shows that the performance of current image embedding models is fragmented, with no method dominating all task categories. We further study the predictability of the performance of visual encoders as part of Multimodal Large Language Models (MLLMs), via a large-scale correlation study. We find that the performance of vision encoders on MIEB strongly correlates with the performance of MLLMs that use the same vision encoder. For instance, the performance on our Visual STS tasks has over 99% correlation with the performance of an MLLM leveraging the same vision encoder on tasks like OCRBench and TextVQA. This provides a practical way to select vision encoders for MLLMs based on MIEB results.

## 8.2 The MIEB Benchmark

### 8.2.1 Overview

Existing image benchmarks are often task-specific (e.g., retrieval (162)) with fine-grained domains (e.g., landmarks (163), artworks (164)). These benchmarks lack a unified evaluation of MLLM-based embeddings, and also have not covered important capability aspects, such as visual text tasks and multilingual breadth. MIEB

| Task category | Example abilities assessed | # Tasks | # Languages | Modalities |
|---|---|---|---|---|
| **Retrieval** | cross-modal/-lingual matching | 45 | 38 | i-i; i-t; t-i; it-i; it-t; i-it; t-it; it-it; i-t |
| **Document Understanding (Retrieval)** | OCR abilities | 10 | 2 | t-i; i-t; it-t |
| **Linear Probing (Classification)** | information encoded | 22 | 1 | i-i; i-i |
| **Clustering** | embedding space consistency | 5 | 1 | i-i |
| **Zero-shot Classification** | cross-modal matching | 23 | 1 | i-t; i-t |
| **Compositionality Evaluation (PairClassification)** | reasoning with confounders | 7 | 1 | i-t; t-i |
| **Vision-centric QA (Retrieval)** | counting, object detection | 6 | 1 | it-t; it-i |
| **Visual STS** | OCR abilities | 9 | 12 | i-i |
| **MIEB** | all | 130 | 38 | all |
| **MIEB-lite** | all | 51 | 38 | all |

Table 8.1: **An overview of MIEB tasks.** In brackets behind task categories, we denote the task type implementation in the code, e.g., our document understanding tasks use our retrieval implementation. We denote the modalities involved in both sides of the evaluation (e.g., queries and documents in retrieval; images and labels in zero-shot classification) with i=image, t=text.

comprehensively fills in this gap. MIEB provides a unified framework to evaluate diverse abilities of embedding models. We categorize tasks based on a combination of the evaluation protocol (e.g., Clustering) and the abilities assessed (e.g., Document Understanding) to better align with user interests. Figure 8.1 and Table 8.1 summarize MIEB task categories. Beyond traditional tasks like linear probing, zero-shot classification, and image-text retrieval, we emphasize under-explored capabilities in image-text embedding models via: **1)** Visual representation of texts, covered by document understanding and visual STS; **2)** Vision-centric abilities, including spatial and depth relationships; **3)** Compositionality; **4)** Interleaved embedding; **5)** Multilinguality.

In addition to MIEB (130 tasks), we introduce MIEB-lite, a lightweight version of MIEB with 51 tasks to support efficient evaluation, by selecting representative tasks from task performance clusters, detailed in subsection 8.6.3. We refer to Appendix for all datasets, statistics, and evaluation metrics for MIEB and MIEB-lite, and section 8.4 for implementation details. Here, we discuss task categories and capabilities assessed.

**Retrieval** Retrieval evaluates if embeddings of two similar items (images or texts) have high similarity (165). We focus on three retrieval aspects: **1) Modality**: The combination of images and texts among queries and documents and whether they are interleaved; **2) Multilinguality**: Whether tasks cover mulitple languages, including texts in images; **3) Instructions** Some tasks may benefit from instructions on

| Task | Short Definition | Metric |
|---|---|---|
| Retrieval | Retrieve target items (text, image or interleaved) using a query (text, image or interleaved) | nDCG@10 |
| Document Understanding | Retrieve a target visual document given a text query | nDCG@5 |
| Linear Probe | Classify labels using a linear classifier trained on frozen representations | accuracy |
| Clustering | Cluster images of different labels into distinct clusters | nmi |
| Zero-shot Classification | Classify labels using Similarity of image and text embeddings | accuracy |
| Compositionality Evaluation | select the correct match to the query over hard negative options | accuracy |
| Vision-centric QA | select the correct answer to a vision-centric reasoning question | accuracy |
| Visual STS | judge similarity of visual text semantics using image embedding similarity, and compare with human labels | Spearman Correlation |

Table 8.2: Brief summary of task definition and metric.

what to retrieve, e.g., in VQA tasks questions in the text serve as example-specific instructions. We use nDCG@10 as the primary metric (14; 162), and recall@1/map@5 for some tasks to align with prior work or adjust for difficulty.

**Document understanding** There has been much interest in using image embeddings to understand entire documents with interleaved figures and tables (166). To address these needs, we create a separate document understanding category. It uses the same evaluation procedure as retrieval and nDCG@5 as the main metric.

**Linear probing** For linear probing, a linear model is trained on embedded images to predict associated class labels (23; 167). Linear probing allows evaluating knowledge encoded in embeddings, even if they are not spatially consistent as would be needed for good clustering performance. We opt for few-shot linear probing (43; 74) with a default of 16 shots per class on which we train a logistic regression classifier with a maximum of 100 iterations. This method is more efficient than probing on the entire dataset (23; 42; 168), making it suitable for large-scale benchmarks like ours.

In subsection 8.6.1, we ablate the performance trend of k-shot per class, showing that model ranking generally remains the same across different values of k. In text embeddings, this task is often called classification (74), so we adopt that term in our code.

**Zero-shot Classification** While generally using the same tasks as linear probing (e.g., ImageNet (169)), zero-shot Classification directly matches image embeddings to classes without training a separate classifier. We follow common practice and turn class labels into text prompts (e.g., for our ImageNet task, a text prompt could be "a photo of space shuttle"). This task is related to retrieval, specifically, a setting where we only care about the top-1 match. We measure accuracy following prior work (23). Models trained with non-representation losses, such as autoregressive models, often lack good off-the-shelf zero-shot performance, but may still perform well in linear probing (18).

**Compositionality Evaluation** Vision-language compositionality assesses whether the composition of a given set of elements aligns with an image and a text, such as relationships between objects, attributes, and spatial configurations. Commonly, it involves distinguishing a ground truth from hard negatives with perturbed inputs, e.g., word order shuffling in ARO benchmark (65). In our code implementation, we also refer to it as ImageTextPairClassification, as images and texts come in small pairs. The main metric we use for this task category is accuracy.

**Vision-centric question answering** Inspired by insights from MLLMs (1), we include vision centric question answering tasks, including object counting, spatial relationships, etc. We also include other challenging visual perception tasks, such as perceiving art styles. This task category can be seen as a form of retrieval where the corpus is a small set of query-specific options (see Figure 8.1), thus it uses our retrieval code implementation.

**Clustering** We use k-means clustering (with k set to the number of true labels) and Normalized Mutual Information (NMI) (170; 171) as the main metric to evaluate

if image embeddings group meaningfully in the embedding space according to the labels.

**Visual STS** Semantic textual similarity (STS) is an established task to evaluate text embeddings (9; 13). It measures the similarity of text embeddings compared to human annotations via Spearman correlation.

In MIEB, we conceptualize *"Visual STS"* (46) as an out-of-distribution task to assess *how good vision encoders are at understanding relative semantics of texts*. We implement it by rendering STS tasks into images to be embedded by models. We compute embedding similarity scores and compare with human annotations at the dataset level using Spearman correlation as the primary metric, following practices for STS evaluation (74). Leveraging this novel protocol, we reveal optical character recognition (OCR) of models like CLIP, which have largely gone unnoticed.

### 8.2.2 Design Considerations

**Generalization** We emphasize **zero-shot** evaluation where models are not fine-tuned for specific tasks; only their embeddings are used. A special case is linear probing, where 'frozen' embeddings are used to train a linear model. However, as the embedded information is not modified, we still consider it zero-shot.

**Usability** In line with MTEB (74), we prioritize: **1) Simplicity**: New models can be added and benchmarked in less than 5 lines of code by using our existing implementations or defining a new model wrapper that can produce image embeddings and text embeddings with the model checkpoint; **2) Extensibility**: New dataset can be added via a single file specifying the download location of a dataset in the correct format, its name, and other metadata; **3) Reproducibility**: The benchmark is fully reproducible by versioning at a model and dataset level; **4) Diversity**; MIEB covers 8 diverse task categories with many different individual tasks, assessing distinct abilities for comprehensive benchmarking and flexibility to explore specific capabilities.

## 8.3 Models

We evaluate three main model categories on MIEB. Note that the categories may overlap.

### 8.3.1 Vision-only Models

MOCO-v3 (168) builds upon MOCO-v1/2 with the ViT architecture and a random patch projection technique to enhance training stability. DINO-v2 (42) scales self-supervised learning to 142M images with similarity-based curation. Different from previous computer vision systems that are trained to predict a fixed set of predetermined object categories (e.g., "ImageNet models" (172)), these models are also referred to as **self-supervised** models.

### 8.3.2 CLIP Models

CLIP (Contrastive Language-Image Pre-training) (23) trains models simultaneously on text-image pairs. We evaluate many models across this line of research including CLIP, SigLIP (44), ALIGN (45), Jina-CLIP (173), DataComp-CLIP (174), OpenCLIP (43), and Eva-CLIP (175). These models are also sometimes referred to as **language-supervised** models (1; 23). We also evaluate VISTA (176), which fuses a ViT encoder (177) with a pretrained language model followed by CLIP-style training.

### 8.3.3 MLLM-based models

Embedding models increasingly leverage MLLMs. For open-source models, we benchmark E5-V (7) and VLM2Vec (178). E5-V uses pre-trained MLLMs followed by text-only contrastive fine-tuning with prompts like "summarize the above sentence with one word" and last-token pooling (73; 124), showing surprising generalization to images and interleaved encodings. VLM2Vec trains MLLM backbones on paired image-text datasets.

We also evaluate the Voyage API model (179). Recent multi-modal API embedding models optimize not only for standard image search, but also for business search

applications like figure and table understanding, making them strong candidates for tasks that require deep visual-text understanding in MIEB.

**Summary** To summarize, we have covered vision-only models, CLIP-style models, and MLLM-based models. In line with previous chapters, we note that sparse/hybrid retrievers are out of scope of our evaluation. Also, training data/size differences of models limit strict fairness across models, and instead serve as insights that we want to provide.

## 8.4 Implementation Details

For interleaved inputs in retrieval and other task categories, we follow the original implementation of each model if it is capable of taking in mixed-modality inputs (176), e.g., MLLM-based embedding models (7; 178). Else, we by default apply a simple sum operation on text and image embeddings (162) to attain interleaved embeddings, e.g., for CLIP-style models (23; 44; 174; 175).

## 8.5 Experimental Results

Table 8.3 presents the overall results for the top 20 models on MIEB (130 tasks) and MIEB-lite (51 tasks). We find that there is no universal embedding model with the best performance on all task categories.

MLLM-based models lead in overall performance on MIEB and MIEB-lite, most notably excelling in visual text understanding and multilingual tasks. However, they perform worse than CLIP-style models in linear probing and zero-shot classification, indicating a loss of precision in image representations. MLLM-based models struggle particularly with fine-grained classification tasks, such as bird species identification (see detailed results in Appendix).

Conversely, CLIP-style models are strong in traditional tasks like linear probing, zero-shot classification, and retrieval. Scaling model size, batch size, and dataset quality improves performance in clustering, classification, and retrieval, but not universally. These models struggle on interleaved retrieval, visual text representations,

| MIEB Full (130 tasks) | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Model Name (↓)** | **Model Type** | **Rtrv. (45)** | **Clus. (5)** | **ZS. (23)** | **LP. (22)** | **Cmp. (7)** | **VC. (6)** | **Doc. (10)** | **vSTS (en) (7)** | **Rtrv. (m) (3 (55))** | **vSTS (x&m) (2 (19))** | **Mean (en) (125)** | **Mean (m) (130)** |
| Voyage-multimodal-3 | MLLM | 38.8 | 82.4 | 58.2 | 71.3 | 43.5 | 48.6 | **71.1** | **81.8** | 58.9 | **70.4** | **62.0** | **62.5** |
| E5-V | MLLM | 34.0 | 70.0 | 50.0 | 74.5 | **46.3** | 51.9 | 62.7 | 79.3 | **66.6** | 46.3 | 58.6 | 58.2 |
| siglip-so400m-patch14-384 | Enc. | 40.8 | 82.1 | **70.8** | 84.6 | 40.4 | 46.3 | 56.4 | 68.0 | 40.2 | 41.4 | 61.2 | 57.1 |
| siglip-large-patch16-384 | Enc. | 39.9 | 79.9 | 68.0 | 83.7 | 39.7 | 45.4 | 53.3 | 69.5 | 51.1 | 39.8 | 59.9 | 57.0 |
| siglip-large-patch16-256 | Enc. | 38.8 | 82.1 | 67.7 | 82.5 | 40.8 | 44.9 | 39.4 | 67.4 | 49.8 | 38.1 | 57.9 | 55.2 |
| siglip-base-patch16-512 | Enc. | 38.1 | 74.7 | 64.1 | 80.9 | 37.5 | 53.2 | 52.1 | 67.7 | 43.2 | 38.1 | 58.5 | 54.9 |
| CLIP-ViT-bigG-14-laion2B | Enc. | **41.5** | 85.6 | 69.4 | 83.6 | 42.4 | 43.2 | 43.2 | 70.9 | 28.0 | 34.5 | 60.0 | 54.2 |
| siglip-base-patch16-384 | Enc. | 37.7 | 76.3 | 64.1 | 80.6 | 38.5 | 52.8 | 45.0 | 67.0 | 42.5 | 37.5 | 57.8 | 54.2 |
| EVA02-CLIP-bigE-14-plus | Enc. | 40.1 | **92.4** | 70.8 | **86.0** | 45.7 | 39.4 | 32.3 | 72.0 | 27.8 | 28.2 | 59.8 | 53.5 |
| CLIP-ViT-L-14-DataComp.XL | Enc. | 38.1 | 86.4 | 68.4 | 82.0 | 39.1 | 52.3 | 38.6 | 69.9 | 23.8 | 35.8 | 59.4 | 53.4 |
| siglip-base-patch16-256(m) | Enc. | 35.6 | 74.6 | 61.2 | 78.9 | 38.1 | 51.3 | 26.4 | 65.5 | 59.2 | 40.3 | 53.9 | 53.1 |
| CLIP-ViT-H-14-laion2B | Enc. | 39.7 | 83.9 | 67.5 | 82.5 | 42.0 | 45.8 | 40.4 | 65.5 | 25.5 | 33.9 | 58.4 | 52.7 |
| CLIP-ViT-g-14-laion2B | Enc. | 39.8 | 82.7 | 67.9 | 82.8 | 41.9 | 44.2 | 37.6 | 69.1 | 25.9 | 31.7 | 58.3 | 52.4 |
| EVA02-CLIP-bigE-14 | Enc. | 39.0 | 89.4 | 69.3 | 84.5 | 42.4 | 43.6 | 31.6 | 68.8 | 25.5 | 28.3 | 58.6 | 52.2 |
| siglip-base-patch16-256 | Enc. | 36.6 | 75.2 | 63.1 | 79.7 | 39.5 | 52.2 | 31.7 | 66.2 | 41.3 | 34.4 | 55.5 | 52.0 |
| siglip-base-patch16-224 | Enc. | 36.3 | 74.5 | 62.6 | 79.3 | 39.8 | 51.1 | 26.2 | 64.3 | 41.2 | 33.5 | 54.3 | 50.9 |
| CLIP-ViT-L-14-laion2B | Enc. | 38.0 | 83.5 | 65.8 | 81.2 | 40.8 | 45.9 | 36.3 | 65.8 | 23.0 | 26.0 | 57.2 | 50.6 |
| VLM2Vec-LoRA | MLLM | 27.7 | 72.6 | 46.3 | 62.0 | 34.6 | 62.0 | 49.7 | 72.6 | 34.9 | 42.2 | 53.4 | 50.5 |
| VLM2Vec-Full | MLLM | 27.6 | 70.7 | 46.3 | 62.0 | 35.4 | **62.1** | 49.8 | 72.6 | 35.0 | 42.2 | 53.3 | 50.4 |
| clip-vit-large-patch14 | Enc. | 33.7 | 76.4 | 62.1 | 80.1 | 44.8 | 44.1 | 38.0 | 64.5 | 20.2 | 35.1 | 55.4 | 49.9 |
| **MIEB-lite (51 tasks)** | | | | | | | | | | | | | |
| **Model Name (↓)** | **Model Type** | **Rtrv. (11)** | **Clus. (2)** | **ZS. (7)** | **LP. (8)** | **Cmp. (6)** | **VC. (5)** | **Doc. (6)** | **vSTS (en) (2)** | **Rtrv. (m) (2 (47))** | **vSTS (x&m) (2 (19))** | **Mean (en) (47)** | **Mean (m) (51)** |
| Voyage-multimodal-3 | MLLM | 33.2 | 76.6 | 48.6 | 69.3 | 35.8 | 50.0 | **63.5** | **84.2** | 49.0 | **70.4** | **57.7** | **58.1** |
| siglip-so400m-patch14-384 | Enc. | 32.4 | 75.9 | 73.8 | 78.8 | 32.8 | 48.0 | 46.9 | 69.6 | 35.4 | 41.4 | 57.3 | 53.5 |
| siglip-large-patch16-384 | Enc. | 31.9 | 75.2 | 71.3 | 77.7 | 32.1 | 46.8 | 44.9 | 69.6 | 43.5 | 39.8 | 56.2 | 53.3 |
| E5-V | MLLM | 26.9 | 51.7 | 36.2 | 70.6 | **39.4** | 52.6 | 56.0 | 81.2 | **58.3** | 46.3 | 51.8 | 51.9 |
| siglip-large-patch16-256 | Enc. | 31.0 | 76.5 | 70.3 | 76.3 | 33.4 | 46.5 | 31.9 | 67.6 | 42.6 | 38.1 | 54.2 | 51.4 |
| CLIP-ViT-bigG-14-laion2B | Enc. | 34.2 | 80.8 | 72.4 | 77.8 | 35.0 | 43.0 | 35.5 | 73.4 | 26.2 | 34.5 | 56.5 | 51.3 |
| siglip-base-patch16-512 | Enc. | 30.8 | 69.7 | 66.3 | 74.6 | 29.7 | 55.5 | 42.6 | 67.1 | 34.8 | 38.1 | 54.5 | 50.9 |
| EVA02-CLIP-bigE-14-plus | Enc. | **35.2** | **87.3** | **74.0** | **80.0** | 38.9 | 38.8 | 26.2 | 73.7 | 26.0 | 28.2 | 56.8 | 50.8 |
| siglip-base-patch16-384 | Enc. | 30.6 | 72.2 | 66.0 | 74.4 | 31.0 | 55.1 | 37.1 | 66.9 | 34.5 | 37.5 | 54.1 | 50.5 |
| CLIP-ViT-L-14-DataComp.XL | Enc. | 31.0 | 80.4 | 69.4 | 75.3 | 31.6 | 54.9 | 30.8 | 72.5 | 22.6 | 35.8 | 55.7 | 50.4 |
| CLIP-ViT-H-14-laion2B | Enc. | 32.8 | 79.3 | 69.4 | 76.8 | 34.8 | 46.8 | 33.7 | 68.3 | 23.9 | 33.9 | 55.2 | 50.0 |
| EVA02-CLIP-bigE-14 | Enc. | 34.3 | 86.7 | 73.0 | 78.3 | 35.1 | 44.4 | 25.1 | 69.9 | 23.9 | 28.3 | 55.9 | 49.9 |
| siglip-base-patch16-256(m) | Enc. | 28.2 | 68.2 | 63.2 | 73.4 | 30.7 | 53.3 | 22.9 | 63.7 | 52.9 | 40.3 | 50.4 | 49.7 |
| CLIP-ViT-g-14-laion2B | Enc. | 33.5 | 76.8 | 69.6 | 77.3 | 34.7 | 45.0 | 29.9 | 71.6 | 24.2 | 31.7 | 54.8 | 49.4 |
| siglip-base-patch16-256 | Enc. | 29.5 | 69.6 | 65.6 | 73.6 | 32.2 | 54.4 | 25.0 | 66.1 | 33.5 | 34.4 | 52.0 | 48.4 |
| CLIP-ViT-L-14-laion2B | Enc. | 31.1 | 76.4 | 67.8 | 75.9 | 33.6 | 46.9 | 28.7 | 68.7 | 21.4 | 26.0 | 53.6 | 47.6 |
| clip-vit-large-patch14 | Enc. | 26.7 | 71.3 | 63.8 | 74.5 | **39.4** | 44.9 | 29.4 | 69.4 | 19.8 | 35.1 | 52.4 | 47.4 |
| siglip-base-patch16-224 | Enc. | 29.3 | 68.4 | 65.0 | 73.5 | 32.5 | 53.0 | 20.9 | 64.2 | 33.6 | 33.5 | 50.8 | 47.4 |
| CLIP-ViT-B-16-DataComp.XL | Enc. | 28.3 | 73.6 | 61.9 | 73.2 | 31.4 | 56.9 | 22.7 | 69.7 | 19.9 | 28.5 | 52.2 | 46.6 |
| VLM2Vec-LoRA | MLLM | 21.0 | 66.3 | 32.1 | 64.8 | 29.4 | **65.3** | 42.7 | 70.9 | 24.8 | 42.2 | 49.1 | 46.0 |

Table 8.3: **MIEB results broken down by task categories for the top 20 models.** We provide averages of both English and multilingual tasks. Models are ranked by the Mean (m) column. Shortcuts are x=Crosslingual, m=Multilingual, en=English, and task categories from Figure 8.1. We refer to the leaderboard for the latest version: `https://hf.co/spaces/mteb/leaderboard`

and multilingual tasks unless specifically optimized (e.g., the multilingual variant of SigLIP).

The strong performance of MLLM-based embedding models and insights from their training recipes highlight a potential pathway for future universal embedding models. E5-V (7), a LLaVA-based model (180), achieves state-of-the-art open-source performance on document understanding, visual STS, multilingual retrieval, and compositionality, despite using a small batch size of 768 for text-only lightweight contrastive finetuning. This suggests its generative pretraining already leads to strong multimodal representations. However, it performs poorly on linear probing and zero-shot classification. Focusing on such tasks in a larger scale finetuning stage may lead to good universal performance.

**Main empirical findings** To summarize our main empirical findings, we find that (i) there is no single universal winner across all task categories; (ii) MLLM-based models excel on visual text, document understanding, multilingual retrieval, inheriting their base model capabilities; (iii) CLIP-style models lead on classification and some retrieval tasks. Future scaling efforts on MLLM-based models can benefit from considering best practices of MLLM and CLIP models to take the best of both worlds. (iv) Vision encoders performance on MIEB strongly correlates with MLLM performance on relevant downstream tasks. For instance, Visual STS strongly correlates with OCR/text-heavy MLLM benchmarks.

We analyze each category in the following sections and refer to the Appendix for full results.

### 8.5.1 Retrieval

The best overall performance is achieved by *CLIP-ViT-bigG-laion2B-39B-b160k* (43) and *siglip-so400m-patch14-384* (44). We find that MLLM-based models with their natural interleaved encoding abilities excel on sub-categories like VQA retrieval (retrieving correct answers given questions and images). For some tasks vision-only models can achieve the best performance, e.g., Dino-v2 (42) on CUB200. We refer to Appendix for full retrieval results.

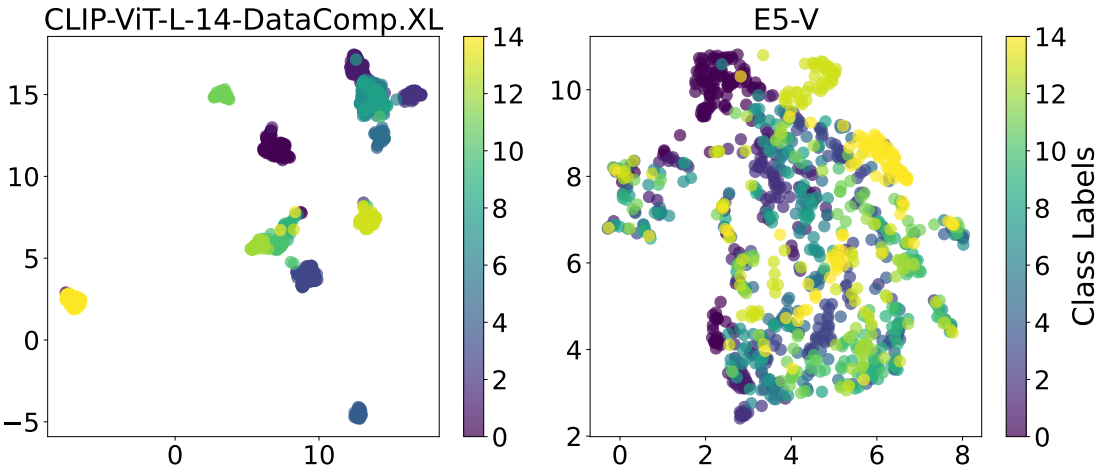


Figure 8.2: **UMAP Visualization of ImageNet Dog15.** Each class corresponds to one dog breed. CLIP clusters are more distinct.

### 8.5.2 Clustering

Similar to findings for Retrieval, MLLM-based models fall short on tasks with fine-grained categories (e.g., dog breeds in ImageNet-Dog15 (169)), indicating their limitations in encoding nuanced image features. Figure 8.2 is a UMAP visualization on ImageNet Dog15, where E5-V underperforms CLIP-style models, showing less separation between fine-grained labels. EVA-CLIP (175), DataComp-CLIP (174), and OpenCLIP checkpoints (43) dominate in most clustering tasks. Similar to patterns in classification shown in the next section, state-of-the-art MLLM-based models have poor performance distinguishing fine-grained classes. We refer to Appendix for full clustering results.

### 8.5.3 Zero-shot Classification

Similar to Retrieval and Clustering, Zero-shot Classification requires coherent image and text embedding subspaces, thus CLIP-style models still dominate. MLLM-based models like E5-V, Voyage, and VLM2Vec largely underperform in zero-shot classification tasks, most notably ones with fine-grained labels. While decoder-based generative models show inherent generalizability in embedding tasks (7; 36; 37; 118; 181), it is likely still necessary to learn robust fine-grained nuances through contrasting multimodality finetuning paired with validated training recipes like large

batch sizes and diverse datasets (23; 43; 174; 175).

### 8.5.4 Linear Probing

Average performance on linear probing is generally the highest among all our task categories, signaling that it is closer to saturation. However, with relatively low overall average scores on MIEB, there is still significant room to improve on the benchmark. In subsection 8.6.1, we investigate label granularity and ablate the number of shots in linear probing, validating the robustness of our design choice of 16-shot for few-shot linear probing (section 8.2).

### 8.5.5 Multilingual Retrieval

Our multilingual retrieval tasks span 38 languages with 55 subtasks (182; 183). We present the full results in Appendix and summarize the key findings here in Table 8.4.

E5-V (7) achieves state-of-the-art performance on multilingual retrieval, highlighting the inherent strong multilingual abilities of LLaVA-Next (184), which E5-V initializes from. E5-V was fine-tuned contrastively using LoRA (185), which only lightly modifies the underlying models, thus leaving most knowledge (such as about different languages) intact. The multilingual version of SigLIP (44), *siglip-base-patch16-256-multilingual*, attains the second best performance. VISTA (176) models also perform strongly despite their relatively small sizes, showing notable consistency across languages. This cross-lingual robustness likely stems from its frozen backbone text model BGE-M3, which was trained to produce high-quality multilingual textual embeddings (112; 113).

Overall, these findings highlight that a strong text encoder trained across various languages is critical to good multilingual performance.

### 8.5.6 Visual STS

For Visual STS (see Appendix for full results), E5-V (7) achieves the best performance. This is likely because it was trained on the allNLI collection (SNLI (186) + MNLI (187)), which is commonly used to train text representation models for

| Model Name | xFlickr&CO | | XM3600 | | WIT | | avg. | |
|---|---|---|---|---|---|---|---|---|
| | avg. | var. | avg. | var. | avg. | var. | avg. | var. |
| E5-V | **90.8** | **0.1** | **74.8** | 3.5 | **57.3** | 0.6 | **74.3** | 1.4 |
| SigLIP | 80.4 | 1.2 | 65.6 | 5.3 | 54.4 | 1.3 | 66.8 | 2.6 |
| VISTA (m3) | 65.3 | 0.2 | 48.5 | **2.0** | 49.3 | **0.4** | 54.4 | **0.9** |
| VLM2Vec | 63.8 | 3.8 | 27.0 | 4.7 | 31.7 | 2.5 | 40.8 | 3.6 |
| Open-CLIP | 35.9 | 9.3 | 20.5 | 6.0 | 37.8 | 6.5 | 31.4 | 7.3 |
| EVA02-CLIP | 35.6 | 9.4 | 20.1 | 6.0 | 37.4 | 6.4 | 31.0 | 7.2 |

Table 8.4: **Performance of models on multilingual retrieval tasks across 38 languages.** We compute the average performance across languages (avg) and the respective variance (var). We take the best variant from each top-6 model family.

| | **12** | **13** | **14** | **15** | **16** | **17** | **b** | **avg.** |
|---|---|---|---|---|---|---|---|---|
| STS* | 80.0 | 89.9 | 85.7 | 89.1 | 85.9 | 87.9 | 83.5 | 86.0 |
| v-STS (ours) | 73.2 | 78.2 | 74.9 | 84.2 | 79.5 | 85.8 | 79.4 | 79.3 |

Table 8.5: **E5-V performance on regular STS and our Visual STS.** *: numbers from (7). Columns are STS12-17 and STS-b.

STS tasks (18). As our Visual STS simply renders existing STS tasks as images (section 8.2), if a model is perfect in optical character recognition (OCR), its Visual STS performance would match its STS performance. Table 8.5 shows that this is almost the case, with some room left for improving the text recognition capabilities of E5-V.

(1) show that textually-supervised models like CLIP are inherently good visual text readers, while purely visually-supervised models are not. Our results support this finding: EVA-CLIP, DataComp-CLIP (OpenCLIP variants trained on DataComp (174)), SigLIP, and CLIP achieve strong performance with EVA-CLIP-bigE-14-plus achieving an average English performance of 71.99, whereas Dino-v2 and Moco-v3 perform near random (Spearman correlation of 12.98 and 14.31).

Note that the design of Visual STS enables it to reflect two capabilities: visual text recognition and semantic encoding. The model needs to have good OCR capabilities (recognizing texts in images), and also the ability to understand the text semantics.

### 8.5.7 Document Understanding

As shown in subsection 8.5.6, E5-V has strong OCR performance. This translates to strong performance on our Document Understanding tasks, where it is the best open-source model (avg. nDCG@5 of 62.69 on 10 Vidore tasks). Voyage-multimodal-3 has better performance but is closed-source.

OpenCLIP (43) and DataComp-CLIP (174) variants provide insights into the positive impact of scaling model sizes and datasets to document understanding capabilities. The performance of OpenCLIP scales from 36.26 for its 430M parameter model (Vit-L) to 40.41 for its 990M parameter model (ViT-H); both having seen the same number of training examples. Data quality also matters with DataComp-CLIP achieving 38.64 with a ViT-L trained on only 13B seen examples, while the above OpenCLIP models use 32B examples.

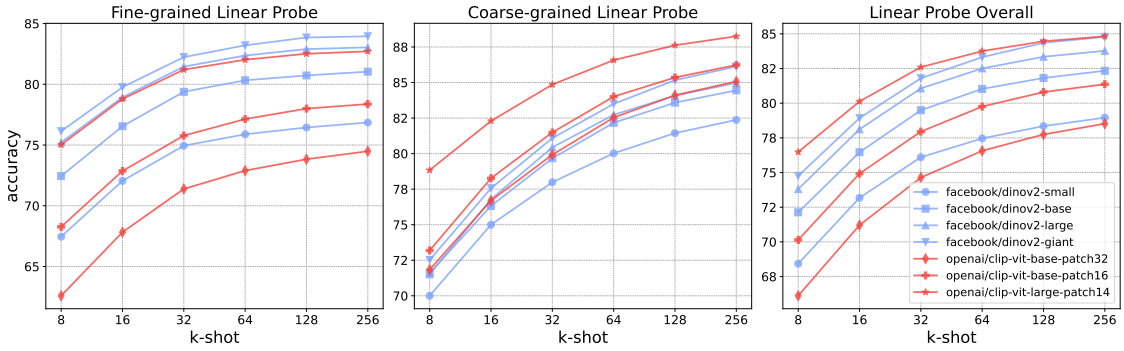


Figure 8.3: **Linear probing performance across different shots k.** We select representative models from our vision-only and CLIP categories (section 8.3). See subsection 8.6.1 for details on fine-grained and coarse-grained tasks.

### 8.5.8 Compositionality Evaluation

Together with Retrieval, Compositionality Evaluation is where models have the lowest scores. Especially, WinoGround (188) is extremely challenging (see Appendix) due to its image and textual confounders. We hypothesize that future models that better incorporate reasoning capabilities and test-time scaling techniques (189; 190) may achieve better results on compositionality tasks.

### 8.5.9 Vision-centric QA

BLIP models (191; 192) surprisingly contribute to two of the top 5 models in vision-centric QA despite their absence for other task categories. This highlights that including images in the contrastive finetuning stage can be beneficial, opposite to their exclusion in (7).

## 8.6 Discussions

### 8.6.1 K-shot Linear Probing

We opt for k-shot linear probing instead of full-dataset linear probing as the default setting in MIEB (section 8.2) to make the evaluation cheaper given the large size of the benchmark. In Figure 8.3, we ablate this design by training k-shot classifiers with k in {8,16,32,64,128,256}. We find that different values of k preserve the same model rank on both **fine-grained classification** (Birdsnap, Caltech101, CIFAR100, Country211, FGVCAircraft, Food101, Imagenet1k, OxfordFlowers, OxfordPets, RESISC45, StanfordCars, SUN397, UCF101) and **coarse-grained classification** (CIFAR10, DTD, EuroSAT, FER2013, GTSRB, MNIST, PatchCamelyon, STL10) tasks. As a result, we choose a modest 16-shot evaluation by default.

### 8.6.2 On the predictability of MLLM performance

MLLM evaluation has been proposed as a robust method to assess visual representations (1), where the performance of an MLLM provides information about the strength of its visual encoder. However, this evaluation paradigm is much more computationally intensive than benchmarking only the vision encoder, given the large sizes of MLLMs and the large hyperparameter search space (data size, LLM choice, instruction-tuning details, etc.). Thus, it remains impractical as a general benchmarking method.

We explore the opposite: Can MLLM performance be predicted from the vision encoder (193)? To do so, we calculate correlations between vision encoder performance on MIEB tasks and their MLLM counterparts across 16 benchmarks using results

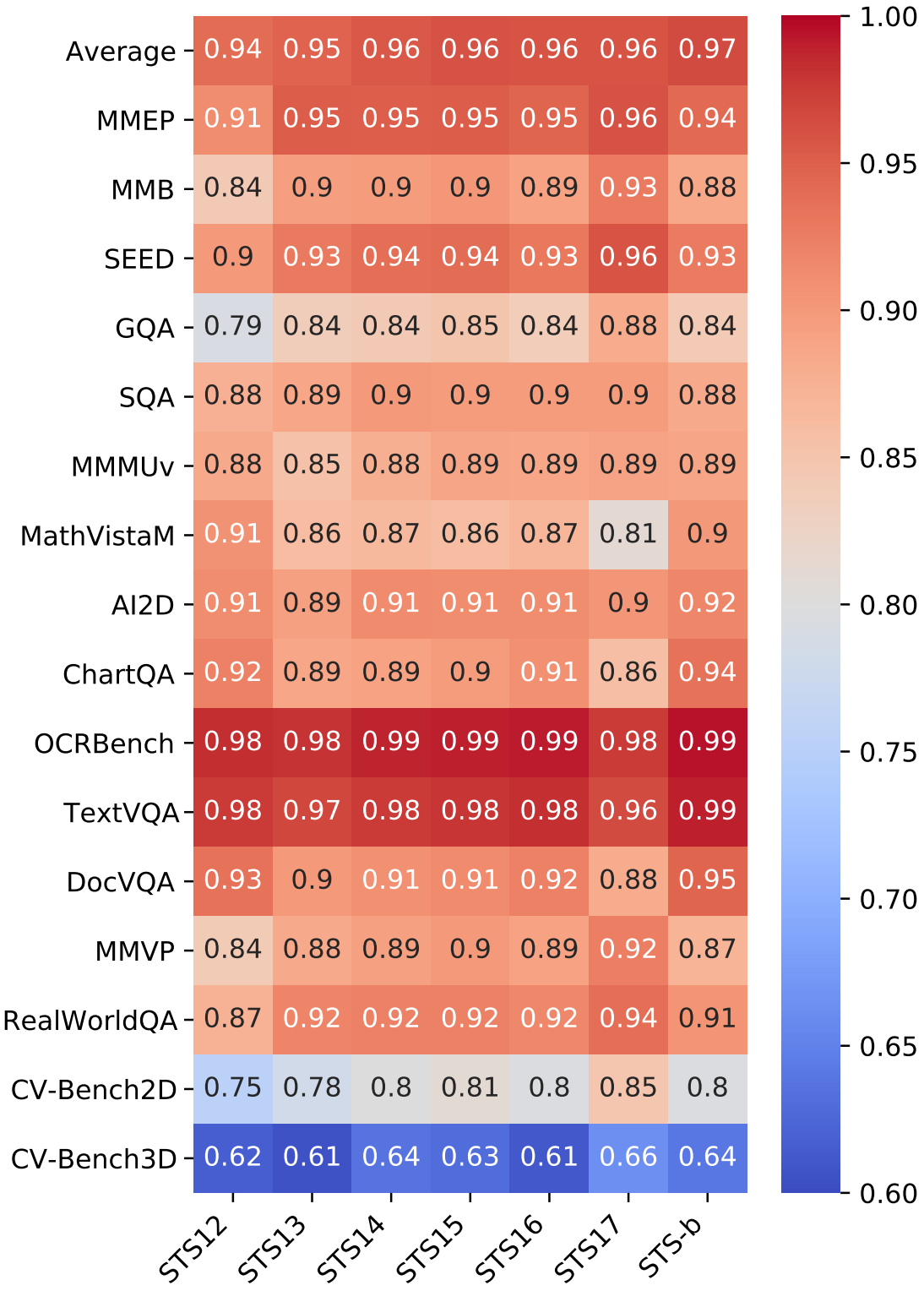


Figure 8.4: **Correlations between performance on generative MLLM benchmarks from Tong et al.(1) (y-axis) and our Visual STS (x-axis).** High correlation means that our Visual STS tasks can predict generative performance.

from (1). Figure 8.4 shows these correlations using our Visual STS protocol as an example (46). Given the common need for visual text interpretation in MLLM tasks, vision encoders' performance on Visual STS has a strong correlation with the performance of their MLLM counterparts. The pattern is most pronounced for the 4 OCR and Chart tasks in (1), and least pronounced for CV-bench 3D, which relies little on visual text understanding. This highlights the utility of MIEB for selecting MLLM vision encoders.

### 8.6.3 MIEB-lite: A lightweight Benchmark

Computationally efficient benchmarks are more usable (194). While MIEB avoids training MLLMs, evaluating 130 tasks remains resource-intensive. While a more comprehensive coverage allows for more nuanced analysis, many tasks have high correlations (e.g., Visual STS in Figure 8.4). To enable lightweight evaluation, we build MIEB-lite by iteratively removing redundant tasks while preserving task category coverage and inter-task correlation.

We first compute pairwise task correlations using model performance, then iteratively remove tasks with average correlations above 0.5 (11 tasks) and 0.45 (32 tasks). Key patterns emerged: 1) Established tasks (e.g., CLIP benchmark linear probing (23)) had high redundancy, possibly due to dataset exposure in pretraining; 2) Easy OCR tasks correlated unexpectedly with non-OCR tasks, though Visual STS and VIDORE remained distinct; 3) Novel tasks (e.g., ARO benchmark, M-BEIR protocols) had low correlations.

To capture nuanced task relationships, we cluster tasks via UMAP+HDBSCAN (195; 196) using correlation vectors, yielding 17 interpretable clusters (e.g., 'fine-grained zero-shot', 'language-centric', 'easy OCR', 'VQA', 'low resolution tasks', etc). The outlier cluster (-1 label) spanned all categories, serving as a foundation for balanced selection.

**MIEB-lite has 51 tasks** by combining the above two approaches and excluding large-scale tasks (e.g., EDIS and GLD-v2 take 60-80 GPU hours for 7B models). MIEB-lite reduces computation while maintaining category balance and diagnostic power: 1) Table 8.6 compares model runtime on MIEB and MIEB-lite showing a

| Model Name | # Params (M) | Runtime (NVIDIA H100 GPU hours) | | |
|---|---|---|---|---|
| | | MIEB | MIEB-lite | Reduction % |
| E5-V | 8360 | 264.0 | 46.4 | 82.4% ↓ |
| CLIP (base-patch32) | 151 | 16.6 | 4.5 | **72.9%** ↓ |

Table 8.6: **MIEB vs. MIEB-lite runtime comparison.**

reduction of **82.4%** for E5-V, an 8B model. 2) We find that the overall average performance of 38 models on MIEB and MIEB-lite has a Spearman correlation of 0.992 and a Pearson correlation of 0.986. See Appendix for all results on MIEB-lite tasks.

To summarize, we combined average correlation removal (threshold of 0.5 correlation with other tasks.) and clustering to select tasks. The resulted compact MIEB-lite benchmark has 51 tasks and still covers the 8 capability categories, maintaining the diagnostic power (model ranking has a Spearman correlation of 0.992 with the full benchmark). We recommend running on MIEB-lite for efficient benchmarking and running the full benchmark to get fine-grained understanding for tasks of different domains.

## 8.7 Related Work

**Benchmarks** Prior efforts toward universal image embedding benchmarks focus on narrow scopes. The CLIP Benchmark (23) evaluates semantic similarity via classification and retrieval, while UnED (197) and M-BEIR (162) expand retrieval evaluation to multi-domain and mixed-modality settings. However, three critical gaps persist: **(1) Limited task diversity**: Existing benchmarks overlook tasks like multi-modal composition (65), social media understanding (198), and multilingual evaluation (183), restricting cross-domain insights. **(2) Neglect visual text tasks**: While understanding text in images is key to many MLLM use cases (166), benchmarks for OCR (199) and visual document retrieval remain sparse. **(3) Under-explored instruction tuning**: Though instruction-tuned embeddings show promise for generalization (200; 201), their evaluation beyond retrieval is limited. MIEB addresses these gaps via unified protocols spanning 130 tasks, consolidating prior

benchmarks into a holistic framework.

**Protocol limitations** Prior work relies heavily on linear probing and retrieval (21; 23), which struggle to assess generalization to complex tasks. While fine-tuning (22) adapts embeddings to specific tasks, it incurs high computational costs and risks overfitting. MIEB evaluates frozen embeddings through a broader suite of protocols including retrieval, linear probing, zero-shot classification, and novel additions like pair-wise classification and clustering, providing a more flexible and comprehensive assessment.

Building on the gap presented in previous work, MIEB presents the first benchmark to comprehensively cover under-evaluated capabilities including visual text, document understanding, multilingual retrieval, and MLLM-based embeddings at this scale. It also shows strong correlation between vision representation with MLLM performance through a large-scale correlation study between Visual STS and downstream MLLM OCR/text tasks.

## 8.8 Limitations

Despite its comprehensiveness, MIEB comes with several limitations. First, MIEB serves as a benchmark for models, not architectures or training methods themselves. It is important to note the model heterogeneity, i.e., models are of different parameter sizes and are trained on different data, sometimes of different orders of magnitude. Second, we mainly provide benchmarking for dense representation, but lack support and evaluation for sparse or hybrid retriever which are popular solutions in pure text applications; Third, the current version of MIEB does not support other modalities, such as video and long-horizon multi-image reasoning), making them natural extensions for future work.

## 8.9 Conclusion

We introduce the Massive Image Embedding Benchmark (MIEB), which consists of 8 task categories with 130 individual tasks covering 38 languages. We benchmark 50

models on MIEB, providing baselines and insights for future research. Our findings highlight the importance of evaluating vision embeddings beyond classification and retrieval, and their role in facilitating multimodal generative models.

Through this chapter, we also answer questions left in previous chapters. For instance, we see how the representation-generation alignment hypothesis (Chapter 6) also holds for multimodal models. This chapter also provides a more extensive study of visual texts which support and extend the findings in Chapter 7. In the next chapter, we will summarize the findings we have so far and present vision for future work.

## 8.10 Additional Information about MIEB

### 8.10.1 Tasks overview

This appendix provides detailed information on all tasks within MIEB, including size, language, metrics, and other relevant details. Note that we present the categories based on Abstask implementations here. We recommend refer to Table 8.1 for the taxonomy based on capabilities assessed.

Table 8.7 shows all information related to retrieval tasks. Table 8.8 presents data related to clustering, standard image classification, zero-shot classification, and multi-label image classification tasks. Lastly, Table 8.9 covers information for visual STS, text-based multiple choice, and image-text pair classification tasks.

For all the paper citation hyperlinks in this Appendix, we refer to the main paper for corresponding citation list.

### 8.10.2 Per Task Category Results

#### Clustering

Table 8.10 presents clustering results of clustering tasks.

#### Vision-centric QA

Table 8.11 presents results of all Vision-centric QA tasks.

#### Multilingual Retrieval

Table 8.12 presents all multilingual retrieval task results, which include 54 subtask results from the 3 multilingual retrieval tasks.

#### Visual STS

Table 8.13 presents English-only STS results across 7 STS tasks. Table 8.14 presents cross-lingual STS results across 11 language pairs. Table 8.15 presents multilingual STS results across 10 languages.

#### Document Understanding

Table 8.16 presents document understanding results.

#### Linear Probe

Table 8.17 and Table 8.18 respectively present linear probing results for coarse-grained and fine-grained classification tasks.

#### Zeroshot Classification

Table 8.19 and Table 8.20 respectively present zero-shot classification results for coarse-grained and fine-grained classification tasks.

#### Compositionality

Table 8.21 presents results of compositionality tasks.

#### Retrieval

Table 8.22 presents results of retrieval tasks.

### 8.10.3 Overall Results & First MIEB Leaderboard

Based on the per-task category results, we provide an overall ranking in Table 8.23, aggregating all results. Note that we currently exclude all models that are not able to evaluate on all tasks in the overall table, including vision-only models like Dino-2

and Moco-v3 that are not able to test on image-text tasks, yielding 36 models in **the first MIEB leaderboard**. Note that for models that are not in the overall table, we refer readers to per task category tables for details.

### 8.10.4 Models

All models used in evaluations are listed in Table 8.24.

| Type (# tasks) | Task | MIEB-lite | # Queries | # Documents | # Qrels | Avg. # Choices | Supported Languages | Queries per Language (multi) | Metric |
|---|---|---|---|---|---|---|---|---|---|
| | BLINKIT2IRetrieval (202) | | 285 | 570 | 285 | - | en | - | Recall@1 |
| | BLINKIT2TRetrieval (202) | | 1073 | 26 | 1073 | - | en | - | Recall@1 |
| | CIRRIT2IRetrieval (203) | ✓ | 4170 | 21551 | 4216 | - | en | - | NDCG@10 |
| | CUB200I2IRetrieval (204) | ✓ | 5794 | 5794 | 163756 | - | - | - | Recall@1 |
| | EDIST2ITRetrieval (205) | | 3241 | 1047067 | 8341 | - | en | - | NDCG@10 |
| | Fashion200kI2TRetrieval (206) | ✓ | 4889 | 61707 | 4889 | - | en | - | NDCG@10 |
| | Fashion200kT2IRetrieval (206) | | 1719 | 201824 | 4847 | - | en | - | NDCG@10 |
| | FashionIQIT2IRetrieval (207) | | 6003 | 74381 | 6014 | - | en | - | NDCG@10 |
| | Flickr30kI2TRetrieval (208) | | 31014 | 155070 | 155070 | - | en | - | NDCG@10 |
| | Flickr30kT2IRetrieval (208) | | 31014 | 155070 | 155070 | - | en | - | NDCG@10 |
| | FORBI2IRetrieval (209) | | 13250 | 53984 | 13250 | - | - | - | Recall@1 |
| | GLDv2I2IRetrieval (163) | | 1129 | 761757 | 15138 | - | - | - | NDCG@10 |
| | GLDv2I2TRetrieval (163) | | 1972 | 674 | 1939 | - | en | - | NDCG@10 |
| | HatefulMemesI2TRetrieval (210) | ✓ | 829 | 8045 | 829 | - | en | - | NDCG@10 |
| | HatefulMemesT2IRetrieval (210) | | 829 | 8045 | 829 | - | en | - | NDCG@10 |
| | InfoSeekIT2ITRetrieval (211) | | 17593 | 481782 | 131376 | - | en | - | NDCG@10 |
| | InfoSeekIT2TRetrieval (211) | ✓ | 11323 | 611651 | 73869 | - | en | - | NDCG@10 |
| | MemotionT2IRetrieval (212) | | 700 | 6988 | 700 | - | en | - | NDCG@10 |
| | METI2IRetrieval (164) | | 87942 | 260655 | 172713 | - | - | - | Recall@1 |
| | MSCOCOI2TRetrieval (213) | | 5000 | 24809 | 24989 | - | en | - | NDCG@10 |
| | MSCOCOT2IRetrieval (213) | | 24809 | 5000 | 24989 | - | en | - | NDCG@10 |
| | NIGHTSI2IRetrieval (214) | ✓ | 2120 | 40038 | 2120 | - | en | - | NDCG@10 |
| | OVENIT2ITRetrieval (215) | | 14741 | 335135 | 261258 | - | en | - | NDCG@10 |
| | OVENIT2TRetrieval (215) | ✓ | 50004 | 676667 | 492654 | - | en | - | NDCG@10 |
| | ROxfordEasyI2IRetrieval (216) | | 70 | 4993 | 345657 | - | - | - | map@5 |
| | ROxfordMediumI2IRetrieval (216) | | 70 | 4993 | 345657 | - | - | - | map@5 |
| | ROxfordHardI2IRetrieval (216) | | 70 | 4993 | 345657 | - | - | - | map@5 |
| Any2AnyRetrieval | RP2kI2IRetrieval (217) | ✓ | 39457 | 39457 | 4409419 | - | - | - | Recall@1 |
| | RParisEasyI2IRetrieval (216) | | 70 | 6322 | 435387 | - | - | - | map@5 |
| | RParisMediumI2IRetrieval (216) | | 70 | 6322 | 435387 | - | - | - | map@5 |
| | RParisHardI2IRetrieval (216) | | 70 | 6322 | 435387 | - | - | - | map@5 |
| | SciMMIRI2TRetrieval (218) | | 16263 | 16263 | 16263 | - | en | - | NDCG@10 |
| | SciMMIRT2IRetrieval (218) | | 16263 | 16263 | 16263 | - | en | - | NDCG@10 |
| | SketchyI2IRetrieval (164) | | 452886 | 25000 | 90577200 | - | en | - | Recall@1 |
| | SOPI2IRetrieval (219) | | 120053 | 120053 | 840927 | - | - | - | Recall@1 |
| | StanfordCarsI2IRetrieval (220) | | 8041 | 8041 | 325570 | - | - | - | Recall@1 |
| | TUBerlinT2IRetrieval (221) | | 250 | 20000 | 20000 | - | en | - | NDCG@10 |
| | VidoreArxivQARetrieval (166) | | 500 | 500 | 500 | - | en | - | NDCG@5 |
| | VidoreDocVQARetrieval (166) | ✓ | 500/451 | 500 | 500 | - | en | - | NDCG@5 |
| | VidoreInfoVQARetrieval (166) | ✓ | 500/494 | 500 | 500 | - | en | - | NDCG@5 |
| | VidoreTabfquadRetrieval (166) | ✓ | 280 | 70 | 280 | - | fr | - | NDCG@5 |
| | VidoreTatdqaRetrieval (166) | ✓ | 1646 | 277 | 1663 | - | en | - | NDCG@5 |
| | VidoreShiftProjectRetrieval (166) | ✓ | 100 | 1000 | 1000 | - | fr | - | NDCG@5 |
| | VidoreSyntheticDocQAAIRetrieval (166) | ✓ | 100 | 968 | 1000 | - | en | - | NDCG@5 |
| | VidoreSyntheticDocQAEnergyRetrieval (166) | | 100 | 977 | 1000 | - | en | - | NDCG@5 |
| | VidoreSyntheticDocQAGovernmentReportsRetrieval (166) | | 100 | 972 | 1000 | - | en | - | NDCG@5 |
| | VidoreSyntheticDocQAHealthcareIndustryRetrieval (166) | | 100 | 965 | 1000 | - | en | - | NDCG@5 |
| | VisualNewsI2TRetrieval (222) | ✓ | 20000 | 537568 | 20000 | - | en | - | NDCG@10 |
| | VisualNewsT2IRetrieval (222) | | 19995 | 542246 | 20000 | - | en | - | NDCG@10 |
| | VizWizIT2TRetrieval (223) | | 4319 | 2091 | 4319 | - | en | - | NDCG@10 |
| | VQA2IT2TRetrieval (224) | ✓ | 214354 | 21597 | 214354 | - | en | - | NDCG@10 |
| | WebQAT2ITRetrieval (225) | ✓ | 2511 | 403196 | 3627 | - | en | - | NDCG@10 |
| | WebQAT2TRetrieval (225) | | 2455 | 544457 | 5002 | - | en | - | NDCG@10 |
| | WITT2IRetrieval (183) | ✓ | 9790 | 8553 | 8291 | - | ar, bg, da, el, et, id, ko, ja, tr, vi, en | 792, 806, 814, 541, 780, 854, 842, 889, 681, 869, 685 | NDCG@10 |
| | XFlickr30kCoT2IRetrieval (183) | | 16000 | 16000 | 16000 | - | de, en, es, id, ja, ru, tr, zh | 2000 each | NDCG@10 |
| | XM3600T2IRetrieval (182) | ✓ | 129600 | 259200 | 259200 | - | ar, bn, cs, da, de, el, en, es, fa, fi, fil, fr, hi, hr, hu, id, it, he, ja, ko, mi, nl, no, pl, pt, quz, ro, ru, sv, sw, te, th, tr, uk, vi, zh | 3600 each | NDCG@10 |

Table 8.7: **Datasets overview and metadata for *Any2AnyRetrieval* task.**

| Type | Task | MIEB-lite | # Samples Train | # Samples Test | # Labels | Metric |
|---|---|---|---|---|---|---|
| ImageClassification | Birdsnap (226) | | 2674 | 1851 | 500 | Accuracy |
| | Caltech101 (227) | | 3060 | 6084 | 101 | |
| | CIFAR10 (228) | | 50000 | 10000 | 10 | |
| | CIFAR100 (228) | | 50000 | 10000 | 100 | |
| | Country211 (23) | ✓ | 28000 | 21100 | 211 | |
| | DTD (229) | ✓ | 3760 | 1880 | 47 | |
| | EuroSAT (230) | ✓ | 16200 | 5400 | 10 | |
| | FER2013 (231) | | 28709 | 7178 | 7 | |
| | FGVCAircraft (232) | | - | 3333 | - | |
| | Food101Classification (233) | | 75750 | 25300 | 101 | |
| | GTSRB (234) | ✓ | 26640 | 12630 | 43 | |
| | Imagenet1k (169) | | 45200 | 37200 | 744 | |
| | MNIST (235) | | 60000 | 10000 | 10 | |
| | OxfordFlowersClassification (236) | | 7169 | 1020 | 102 | |
| | OxfordPets (237) | ✓ | 3680 | 3669 | 37 | |
| | PatchCamelyon (238) | ✓ | 262144 | 32768 | 2 | |
| | RESISC45 (239) | ✓ | 18900 | 6300 | 45 | |
| | StanfordCars (220) | | 8144 | 8041 | 196 | |
| | STL10 (240) | | 5000 | 8000 | 10 | |
| | SUN397 (241) | ✓ | 76127 | 21750 | 397 | |
| | UCF101 (242) | | 1786096 | 697222 | 101 | |
| ImageMultiLabelClassification* | VOC2007 (243) | | - | 4952 | $\in [\mathbf{1-5}]$ | Accuracy |
| ImageClustering | CIFAR10Clustering (228) | | - | 10000 | 10 | NMI |
| | CIFAR100Clustering (228) | | - | 10000 | 100 | |
| | ImageNetDog15Clustering (169) | ✓ | - | 1076 | 15 | |
| | ImageNet10Clustering (169) | | - | 13000 | 10 | |
| | TinyImageNetClustering (244) | ✓ | - | 10000 | 200 | |
| ZeroShotClassification | BirdsnapZeroShot (226) | | 2674 | 1851 | 500 | Accuracy |
| | Caltech101ZeroShot (227) | | 3060 | 6084 | 101 | |
| | CIFAR10ZeroShot (228) | | 50000 | 10000 | 10 | |
| | CIFAR100ZeroShot (228) | ✓ | 50000 | 10000 | 100 | |
| | CLEVRZeroShot (245) | | 51600 | 15000 | 6 | |
| | CLEVRCountZeroShot (245) | | 51600 | 15000 | 8 | |
| | Country211ZeroShot (23) | ✓ | 28000 | 21100 | 211 | |
| | DTDZeroShot (229) | | 3760 | 1880 | 47 | |
| | EuroSATZeroShot (230) | | 16200 | 5400 | 10 | |
| | FER2013ZeroShot (231) | ✓ | 28709 | 7178 | 7 | |
| | FGVCAircraftZeroShot (232) | ✓ | - | 3333 | - | |
| | Food101ZeroShot (233) | ✓ | 75750 | 25300 | 101 | |
| | GTSRBZeroShot (234) | | 26640 | 12630 | 43 | |
| | Imagenet1kZeroShot (169) | | 45200 | 37200 | 744 | |
| | MNISTZeroShot (235) | | 60000 | 10000 | 10 | |
| | OxfordPetsZeroShot (237) | ✓ | 3680 | 3669 | 37 | |
| | PatchCamelyonZeroShot (238) | | 262144 | 32768 | 2 | |
| | RenderedSST2 (23) | | 6920 | 1821 | 2 | |
| | RESISC45ZeroShot (239) | | 18900 | 6300 | 45 | |
| | SciMMIR (218) | | 498279 | 16263 | 5 | |
| | StanfordCarsZeroShot (220) | ✓ | 8144 | 8041 | 196 | |
| | STL10ZeroShot (240) | | 5000 | 8000 | 10 | |
| | SUN397ZeroShot (241) | | 76127 | 21750 | 397 | |
| | UCF101ZeroShot (242) | | 1786096 | 697222 | 101 | |

Table 8.8: **Datasets overview and metadata for *ImageClassification*, *ImageMultiLabelClassification*, *ImageClustering* and *ZeroShotClassification* tasks.** * For *ImageMultiLabelClassification*, the number of labels per sample is between the given interval. Further, we again note that with the large scales of training set in classification datasets, we adopt the few-shot linear probe paradigm in the evaluation.

| Type | Task | MIEB-lite | # Samples Test | # Choices | Supported Languages | # Samples per language | Metric |
|---|---|---|---|---|---|---|---|
| Any2AnyMultiChoice | CVBenchCount (1) | ✓ | 788 | [4-6] | en | - | Accuracy |
| | CVBenchRelation (1) | ✓ | 650 | 2 | en | - | |
| | CVBenchDepth (1) | ✓ | 600 | 2 | en | - | |
| | CVBenchDistance (1) | ✓ | 600 | 2 | en | - | |
| | BLINKIT2IMultiChoice (202) | ✓ | 402 | 2 | en | - | |
| | BLINKIT2TMultiChoice (202) | | 1073 | [2-4] | en | - | |
| ImageTextPairClassification* | AROCocoOrder (246) | ✓ | 25010 | 5 | - | - | Text Accuracy |
| | AROFlickrOrder (246) | ✓ | 5000 | 5 | - | - | |
| | AROVisualAttribution (246) | ✓ | 28748 | 2 | - | - | |
| | AROVisualRelation (246) | ✓ | 23937 | 2 | - | - | |
| | SugarCrepe (247) | | 7511 | 2 | - | - | |
| | Winoground (188) | ✓ | 400 | 2 | - | - | Accuracy |
| | ImageCoDe (248) | ✓ | 25322 | 10 | - | - | |
| VisualSTS | STS12VisualSTS (46) | | 5342 | - | en | - | Cosine Spearman |
| | STS13VisualSTS (46) | ✓ | 1500 | - | en | - | |
| | STS14VisualSTS (46) | | 3750 | - | en | - | |
| | STS15VisualSTS (46) | ✓ | 3000 | - | en | - | |
| | STS16VisualSTS (46) | | 1186 | - | en | - | |
| | STS17MultilingualVisualSTS (46) | ✓ | 5346 | - | ar-ar, en-ar, en-de, en-en, en-tr, es-en, es-es, fr-en, it-en, ko-ko, nl-en | 250 each, except ko-ko with 2.85k | |
| | STSBenchmarkMultilingualVisualSTS (46) | ✓ | 86280 | - | en, de, es, fr, it, nl, pl, pt,ru, zh | 8628 each | |

Table 8.9: **Datasets overview and metadata for *Any2AnyMutipleChoice*, *ImageTextPairClassification* and *Visual STS* tasks. *** For *ImageTextPair-Classification*, only 1 caption is correct over all the available ones for a sample.

| model name | CIFAR10 | CIFAR100 | ImageNet10 | ImageNetDog15 | TinyImageNet | Avg. |
|---|---|---|---|---|---|---|
| EVA02-CLIP-bigE-14-plus | 98.65 | 89.51 | 99.09 | 91.08 | 83.57 | 92.38 |
| EVA02-CLIP-bigE-14 | 90.30 | 89.03 | 94.32 | 89.85 | 83.58 | 89.42 |
| EVA02-CLIP-L-14 | 97.83 | 86.14 | 94.37 | 83.57 | 79.44 | 88.27 |
| laion/CLIP-ViT-L-14-DataComp.XL-s13B-b90K | 93.65 | 84.26 | 93.39 | 82.60 | 78.28 | 86.44 |
| laion/CLIP-ViT-bigG-14-laion2B-39B-b160k | 87.66 | 79.97 | 98.75 | 86.09 | 75.49 | 85.59 |
| laion/CLIP-ViT-H-14-laion2B-s32B-b79K | 88.10 | 78.69 | 93.93 | 85.93 | 72.67 | 83.86 |
| nomic-ai/nomic-embed-vision-v1.5 | 87.39 | 81.16 | 95.80 | 81.19 | 72.65 | 83.64 |
| laion/CLIP-ViT-L-14-laion2B-s32B-b82K | 93.62 | 77.43 | 93.74 | 81.09 | 71.62 | 83.50 |
| facebook/dinov2-large | 79.90 | 79.93 | 92.23 | 86.20 | 77.22 | 83.10 |
| facebook/dinov2-base | 82.62 | 77.20 | 93.93 | 85.67 | 74.31 | 82.74 |
| laion/CLIP-ViT-g-14-laion2B-s34B-b88K | 87.63 | 78.13 | 94.38 | 81.46 | 72.10 | 82.74 |
| EVA02-CLIP-B-16 | 89.23 | 83.51 | 89.22 | 74.82 | 75.96 | 82.55 |
| voyage-multimodal-3 | 86.22 | 75.15 | 97.58 | 83.82 | 69.29 | 82.41 |
| google/siglip-large-patch16-256 | 83.61 | 76.23 | 97.87 | 86.40 | 66.52 | 82.13 |
| google/siglip-so400m-patch14-384 | 83.79 | 76.67 | 98.19 | 83.57 | 68.31 | 82.11 |
| laion/CLIP-ViT-B-16-DataComp.XL-s13B-b90K | 89.93 | 78.21 | 93.42 | 74.98 | 72.13 | 81.73 |
| facebook/dinov2-giant | 76.84 | 75.77 | 91.84 | 92.63 | 70.13 | 81.44 |
| facebook/dinov2-small | 79.25 | 72.62 | 91.23 | 87.27 | 70.65 | 80.20 |
| google/siglip-large-patch16-384 | 81.61 | 74.43 | 93.28 | 84.17 | 66.21 | 79.94 |
| laion/CLIP-ViT-B-32-laion2B-s34B-b79K | 83.76 | 70.92 | 95.22 | 76.31 | 63.76 | 77.99 |
| Salesforce/blip-itm-large-coco | 84.95 | 72.62 | 98.29 | 67.56 | 64.24 | 77.53 |
| laion/CLIP-ViT-B-32-DataComp.XL-s13B-b90K | 79.81 | 74.85 | 91.54 | 73.68 | 66.98 | 77.37 |
| Salesforce/blip-itm-large-flickr | 87.94 | 70.67 | 94.34 | 68.36 | 60.86 | 76.43 |
| openai/clip-vit-large-patch14 | 80.87 | 64.54 | 94.00 | 72.83 | 69.82 | 76.41 |
| google/siglip-base-patch16-384 | 71.62 | 67.78 | 97.63 | 86.16 | 58.15 | 76.27 |
| BAAI/bge-visualized-base | 82.16 | 77.80 | 98.33 | 49.37 | 73.28 | 76.19 |
| google/siglip-base-patch16-256 | 76.82 | 67.58 | 92.57 | 80.47 | 58.73 | 75.24 |
| google/siglip-base-patch16-512 | 73.95 | 66.56 | 93.32 | 79.61 | 59.82 | 74.65 |
| google/siglip-base-patch16-256-multilingual | 75.94 | 67.89 | 92.63 | 80.44 | 55.88 | 74.56 |
| google/siglip-base-patch16-224 | 76.11 | 67.01 | 92.61 | 78.18 | 58.58 | 74.50 |
| blip2-pretrain | 96.67 | 81.46 | 97.77 | 20.27 | 73.86 | 74.01 |
| nyu-visionx/moco-v3-vit-b | 74.69 | 63.99 | 90.30 | 80.77 | 59.53 | 73.86 |
| Salesforce/blip-image-captioning-large | 77.64 | 68.45 | 93.27 | 67.38 | 61.74 | 73.70 |
| BAAI/bge-visualized-m3 | 81.41 | 73.89 | 97.74 | 43.07 | 71.72 | 73.57 |
| TIGER-Lab/VLM2Vec-LoRA | 72.89 | 60.56 | 97.03 | 71.48 | 61.22 | 72.64 |
| nyu-visionx/moco-v3-vit-l | 71.65 | 60.60 | 86.41 | 80.70 | 59.14 | 71.70 |
| TIGER-Lab/VLM2Vec-Full | 69.43 | 60.72 | 92.64 | 69.29 | 61.51 | 70.72 |
| Salesforce/blip-itm-base-coco | 70.83 | 60.44 | 93.19 | 70.31 | 58.19 | 70.59 |
| royokong/e5-v | 82.58 | 70.43 | 93.85 | 36.73 | 66.64 | 70.05 |
| jinaai/jina-clip-v1 | 74.12 | 64.84 | 96.69 | 52.66 | 61.47 | 69.95 |
| openai/clip-vit-base-patch16 | 69.25 | 59.35 | 92.58 | 63.25 | 62.90 | 69.47 |
| openai/clip-vit-base-patch32 | 73.85 | 58.07 | 93.14 | 54.12 | 60.34 | 67.90 |
| blip2-finetune-coco | 90.37 | 75.81 | 93.12 | 8.97 | 70.92 | 67.84 |
| Salesforce/blip-itm-base-flickr | 63.94 | 58.89 | 92.46 | 66.00 | 55.07 | 67.27 |
| Salesforce/blip-image-captioning-base | 64.18 | 53.81 | 90.94 | 58.78 | 47.76 | 63.09 |
| kakaobrain/align-base | 54.13 | 50.68 | 84.21 | 58.88 | 50.03 | 59.59 |

Table 8.10: **Clustering Results.**

| model name | CVBenchCount | CVBenchDepth | CVBenchDistance | CVBenchRelation | BLINKIT2IMultiChoice | BLINKIT2TMultiChoice | Avg. |
|---|---|---|---|---|---|---|---|
| TIGER-Lab/VLM2Vec-Full | 62.18 | 62.17 | 58.00 | 71.69 | 72.39 | 46.28 | 62.12 |
| TIGER-Lab/VLM2Vec-LoRA | 62.56 | 62.50 | 58.17 | 71.08 | 72.39 | 45.40 | 62.02 |
| laion/CLIP-ViT-B-16-DataComp.XL-s13B-b90K | 61.93 | 52.50 | 46.00 | 49.23 | 74.63 | 41.74 | 54.34 |
| google/siglip-base-patch16-512 | 55.20 | 53.67 | 42.83 | 51.38 | 74.38 | 41.74 | 53.20 |
| blip2-pretrain | 46.95 | 57.67 | 50.17 | 47.69 | 74.38 | 41.99 | 53.14 |
| google/siglip-base-patch16-384 | 53.43 | 52.17 | 42.17 | 51.69 | 75.87 | 41.49 | 52.80 |
| blip2-finetune-coco | 44.54 | 59.67 | 52.33 | 48.77 | 71.39 | 39.60 | 52.72 |
| BAAI/bge-visualized-base | 50.25 | 49.00 | 56.33 | 48.15 | 73.63 | 37.20 | 52.43 |
| Salesforce/blip-itm-base-flickr | 60.66 | 44.67 | 50.33 | 53.08 | 66.92 | 38.46 | 52.35 |
| laion/CLIP-ViT-L-14-DataComp.XL-s13B-b90K | 43.27 | 55.83 | 46.50 | 55.54 | 73.13 | 39.72 | 52.33 |
| google/siglip-base-patch16-256 | 54.44 | 52.00 | 40.67 | 51.08 | 73.63 | 41.24 | 52.18 |
| royokong/e5-v | 39.21 | 48.50 | 43.83 | 59.69 | 71.89 | 48.30 | 51.90 |
| google/siglip-base-patch16-256-multilingual | 34.64 | 54.00 | 49.00 | 53.85 | 75.12 | 40.86 | 51.25 |
| Salesforce/blip-itm-large-coco | 45.30 | 50.00 | 49.67 | 48.77 | 74.38 | 38.46 | 51.10 |
| google/siglip-base-patch16-224 | 43.91 | 51.50 | 42.67 | 51.54 | 75.37 | 41.36 | 51.06 |
| Salesforce/blip-image-captioning-large | 14.72 | 63.33 | 59.67 | 46.92 | 70.40 | 39.61 | 49.11 |
| voyage-multimodal-3 | 26.40 | 53.17 | 47.50 | 53.54 | 69.65 | 41.11 | 48.56 |
| Salesforce/blip-itm-base-coco | 26.65 | 45.17 | 45.50 | 52.92 | 76.12 | 37.20 | 47.26 |
| Salesforce/blip-itm-large-flickr | 25.25 | 46.83 | 52.00 | 53.23 | 68.41 | 36.32 | 47.01 |
| openai/clip-vit-base-patch16 | 20.81 | 51.67 | 46.17 | 49.85 | 71.64 | 41.36 | 46.92 |
| nomic-ai/nomic-embed-vision-v1.5 | 21.83 | 45.33 | 50.33 | 48.62 | 75.37 | 38.84 | 46.72 |
| google/siglip-so400m-patch14-384 | 21.70 | 48.33 | 40.00 | 53.38 | 76.37 | 37.70 | 46.25 |
| laion/CLIP-ViT-B-32-DataComp.XL-s13B-b90K | 23.86 | 49.17 | 43.67 | 47.38 | 72.64 | 39.60 | 46.05 |
| laion/CLIP-ViT-L-14-laion2B-s32B-b82K | 8.25 | 49.17 | 47.50 | 55.08 | 74.38 | 40.73 | 45.85 |
| laion/CLIP-ViT-H-14-laion2B-s32B-b79K | 19.80 | 48.67 | 40.17 | 50.92 | 74.63 | 40.60 | 45.80 |
| kakaobrain/align-base | 47.59 | 43.17 | 50.83 | 47.08 | 46.77 | 38.71 | 45.69 |
| jinaai/jina-clip-v1 | 14.85 | 49.33 | 47.00 | 50.77 | 74.88 | 35.44 | 45.38 |
| google/siglip-large-patch16-384 | 8.76 | 54.67 | 45.83 | 50.92 | 73.63 | 38.34 | 45.36 |
| EVA02-CLIP-B-16 | 36.80 | 53.33 | 53.00 | 49.54 | 40.55 | 38.84 | 45.34 |
| google/siglip-large-patch16-256 | 8.88 | 56.17 | 46.17 | 48.15 | 73.13 | 36.70 | 44.87 |
| laion/CLIP-ViT-g-14-laion2B-s34B-b88K | 10.15 | 47.00 | 41.33 | 50.15 | 76.12 | 40.23 | 44.16 |
| openai/clip-vit-large-patch14 | 2.66 | 52.67 | 46.83 | 50.92 | 71.14 | 40.35 | 44.10 |
| BAAI/bge-visualized-m3 | 7.61 | 45.33 | 49.33 | 50.62 | 73.88 | 36.32 | 43.85 |
| EVA02-CLIP-bigE-14 | 30.46 | 48.83 | 48.17 | 49.85 | 44.53 | 39.60 | 43.57 |
| Salesforce/blip-image-captioning-base | 10.15 | 51.50 | 55.33 | 52.62 | 58.24 | 32.83 | 43.44 |
| laion/CLIP-ViT-bigG-14-laion2B-39B-b160k | 4.19 | 47.17 | 42.17 | 48.15 | 73.13 | 44.14 | 43.16 |
| laion/CLIP-ViT-B-32-laion2B-s34B-b79K | 0.38 | 50.00 | 40.83 | 49.69 | 73.38 | 43.51 | 42.97 |
| openai/clip-vit-base-patch32 | 6.60 | 45.33 | 46.00 | 48.46 | 70.15 | 39.85 | 42.73 |
| EVA02-CLIP-bigE-14-plus | 10.15 | 43.83 | 40.50 | 47.38 | 51.99 | 42.75 | 39.43 |
| EVA02-CLIP-L-14 | 1.02 | 49.50 | 53.50 | 45.69 | 45.27 | 41.24 | 39.37 |

Table 8.11: **Vision-centric QA Results.**

| | XFde | XFen | XFes | XFid | XFja | XFru | XFtr | XFzh | XMar | XMbn | XMcs | XMda | XMde | XMel | XMen | XMes | XMfa | XMfi | XMfil | XMfr | XMhe | XMhi | XMhr | XMhu | XMid | XMit | XMja | XMko |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 80.69 | 86.64 | 86.46 | 80.29 | 80.68 | 87.57 | 78.56 | 84.46 | 68.64 | 58.92 | 66.15 | 68.38 | 83.51 | 69.68 | 69.18 | 74.79 | 73.14 | 67.60 | 54.28 | 81.81 | 69.73 | 52.76 | 65.34 | 74.49 | 81.30 | 79.39 | 80.47 | 75.97 |
| | 75.08 | 80.62 | 83.07 | 72.20 | 44.93 | 83.99 | 70.66 | 63.87 | 61.58 | 33.28 | 62.88 | 68.54 | 79.82 | 56.24 | 68.12 | 74.53 | 67.73 | 59.52 | 32.80 | 77.54 | 66.72 | 30.13 | 63.49 | 69.17 | 75.81 | 76.74 | 57.93 | 63.80 |
| | 80.48 | 89.91 | 85.83 | 72.94 | 79.25 | 85.23 | 48.56 | 86.47 | 49.83 | 18.55 | 51.42 | 55.39 | 85.63 | 24.60 | 74.70 | 75.66 | 36.97 | 22.21 | 17.98 | 81.86 | 39.42 | 12.73 | 44.67 | 22.41 | 76.52 | 80.93 | 77.71 | 48.62 |
| | 77.81 | 86.80 | 87.38 | 66.55 | 28.71 | 78.83 | 64.72 | 40.02 | 51.10 | 2.16 | 51.57 | 55.92 | 80.51 | 36.10 | 71.18 | 75.46 | 33.74 | 29.05 | 20.53 | 78.83 | 31.82 | 6.04 | 44.35 | 45.81 | 69.63 | 78.32 | 25.88 | 40.91 |
| | 76.42 | 85.10 | 85.88 | 63.88 | 25.51 | 77.66 | 62.41 | 37.78 | 50.51 | 1.97 | 51.26 | 56.41 | 80.39 | 35.81 | 71.15 | 75.40 | 32.93 | 28.84 | 19.78 | 78.36 | 30.97 | 6.11 | 44.05 | 45.34 | 68.28 | 77.60 | 25.35 | 39.66 |
| | 54.91 | 62.20 | 59.89 | 54.09 | 49.88 | 57.49 | 50.96 | 58.77 | 38.72 | 28.33 | 42.53 | 49.63 | 50.94 | 39.21 | 48.42 | 47.35 | 44.37 | 47.20 | 27.05 | 52.21 | 41.54 | 22.99 | 46.11 | 46.54 | 52.16 | 47.83 | 48.20 | 43.14 |
| | 73.71 | 85.48 | 84.42 | 53.48 | 25.71 | 70.61 | 51.24 | 34.90 | 37.42 | 0.24 | 38.09 | 44.15 | 75.38 | 22.62 | 71.08 | 73.52 | 13.73 | 17.69 | 14.74 | 76.04 | 13.31 | 1.47 | 28.54 | 29.73 | 57.93 | 72.63 | 21.57 | 28.86 |
| | 72.16 | 85.03 | 83.45 | 52.41 | 24.34 | 69.53 | 49.21 | 33.62 | 36.73 | 0.22 | 37.72 | 43.69 | 74.82 | 21.99 | 70.80 | 73.35 | 13.56 | 17.50 | 14.72 | 75.30 | 13.41 | 1.58 | 28.27 | 29.73 | 57.77 | 72.10 | 21.53 | 29.07 |
| | 69.74 | 83.25 | 81.33 | 49.70 | 23.09 | 66.51 | 47.58 | 31.08 | 36.55 | 0.16 | 37.22 | 43.03 | 73.32 | 21.34 | 70.31 | 72.04 | 13.79 | 17.46 | 14.55 | 74.45 | 13.46 | 1.54 | 27.90 | 29.12 | 56.27 | 70.55 | 21.30 | 28.23 |
| | 70.22 | 83.51 | 81.56 | 50.80 | 21.88 | 67.11 | 47.19 | 31.52 | 35.83 | 0.15 | 37.11 | 42.70 | 73.40 | 21.09 | 70.41 | 72.53 | 13.19 | 17.23 | 14.42 | 74.47 | 12.79 | 1.38 | 28.02 | 28.99 | 55.98 | 70.91 | 20.21 | 27.41 |
| | 72.34 | 87.51 | 83.89 | 48.10 | 6.93 | 43.82 | 45.05 | 9.76 | 14.76 | 0.17 | 37.25 | 45.86 | 71.98 | 4.67 | 72.26 | 73.89 | 5.25 | 17.81 | 14.14 | 76.51 | 4.09 | 0.46 | 28.71 | 26.20 | 52.30 | 72.58 | 5.22 | 7.49 |
| | 68.88 | 80.62 | 76.85 | 28.97 | 48.30 | 52.62 | 25.10 | 60.02 | 20.37 | 0.47 | 16.68 | 29.49 | 62.12 | 3.99 | 63.88 | 47.05 | 1.69 | 9.94 | 9.29 | 56.43 | 14.16 | 7.98 | 10.73 | 9.57 | 20.20 | 43.26 | 44.76 | 17.76 |
| | 68.92 | 80.48 | 76.79 | 28.83 | 48.12 | 52.76 | 25.03 | 60.06 | 20.46 | 0.48 | 16.63 | 29.45 | 62.13 | 4.02 | 63.80 | 46.98 | 1.68 | 9.92 | 9.33 | 56.50 | 14.18 | 8.01 | 10.68 | 9.57 | 20.19 | 43.20 | 44.68 | 17.78 |
| | 51.99 | 84.22 | 66.43 | 20.95 | 5.75 | 8.36 | 10.36 | 5.47 | 0.71 | 0.11 | 9.19 | 23.05 | 58.18 | 0.59 | 71.21 | 59.06 | 0.32 | 5.92 | 9.33 | 71.04 | 0.36 | 0.16 | 9.83 | 8.28 | 24.44 | 56.04 | 8.86 | 0.31 |
| | 51.93 | 84.61 | 66.74 | 21.42 | 5.79 | 7.74 | 9.19 | 4.95 | 0.66 | 0.12 | 8.92 | 21.82 | 57.47 | 0.55 | 71.08 | 59.08 | 0.30 | 5.58 | 9.13 | 71.57 | 0.30 | 0.13 | 9.17 | 8.33 | 24.27 | 55.20 | 8.74 | 0.29 |
| | 47.25 | 82.75 | 62.80 | 18.16 | 6.03 | 3.68 | 8.87 | 5.24 | 0.71 | 0.12 | 7.11 | 18.96 | 51.28 | 0.48 | 70.09 | 54.98 | 0.30 | 5.08 | 8.71 | 67.51 | 0.28 | 0.11 | 6.73 | 6.82 | 23.02 | 49.58 | 7.06 | 0.31 |
| | 44.66 | 84.34 | 63.04 | 18.80 | 4.72 | 3.37 | 7.64 | 3.76 | 0.71 | 0.13 | 6.58 | 17.34 | 48.62 | 0.49 | 70.81 | 54.27 | 0.30 | 4.14 | 8.60 | 66.44 | 0.27 | 0.12 | 7.11 | 6.69 | 21.92 | 48.02 | 7.04 | 0.31 |
| | 45.83 | 82.91 | 62.84 | 19.07 | 4.65 | 3.22 | 8.61 | 4.18 | 0.76 | 0.14 | 6.65 | 17.78 | 48.53 | 0.48 | 70.53 | 53.57 | 0.29 | 4.44 | 8.62 | 65.48 | 0.27 | 0.15 | 7.13 | 6.73 | 22.13 | 47.52 | 6.99 | 0.38 |
| | 41.92 | 76.01 | 39.37 | 10.08 | 8.24 | 13.57 | 6.11 | 1.83 | 1.53 | 0.18 | 10.60 | 25.42 | 49.68 | 5.63 | 69.41 | 41.16 | 0.92 | 6.97 | 8.56 | 62.52 | 1.77 | 0.58 | 8.88 | 8.55 | 15.06 | 41.68 | 11.35 | 1.33 |
| 0K | 33.73 | 78.71 | 49.86 | 26.88 | 3.11 | 1.30 | 9.21 | 5.61 | 0.48 | 0.14 | 6.25 | 16.84 | 39.35 | 0.48 | 67.79 | 42.80 | 0.26 | 4.76 | 8.91 | 53.89 | 0.31 | 0.12 | 6.86 | 6.55 | 30.88 | 36.62 | 4.21 | 0.47 |
| | 37.58 | 81.31 | 57.42 | 21.63 | 6.52 | 1.22 | 6.26 | 2.19 | 0.60 | 0.15 | 5.01 | 13.76 | 38.17 | 0.43 | 67.06 | 47.28 | 0.25 | 4.83 | 10.62 | 55.20 | 0.29 | 0.13 | 5.93 | 5.97 | 20.91 | 35.48 | 7.11 | 0.27 |
| | 38.51 | 82.04 | 54.70 | 17.20 | 4.31 | 3.08 | 7.49 | 2.90 | 0.61 | 0.13 | 5.53 | 15.06 | 39.07 | 0.41 | 69.07 | 46.18 | 0.28 | 4.16 | 7.82 | 58.24 | 0.28 | 0.13 | 5.86 | 6.01 | 17.77 | 38.45 | 5.36 | 0.28 |
| 0K | 33.42 | 75.77 | 44.01 | 24.54 | 4.09 | 1.42 | 9.93 | 5.84 | 0.52 | 0.10 | 6.29 | 15.78 | 34.28 | 0.49 | 66.62 | 38.09 | 0.29 | 4.29 | 8.65 | 47.95 | 0.29 | 0.12 | 6.66 | 5.91 | 27.19 | 31.55 | 4.66 | 0.50 |
| 0K | 29.72 | 72.35 | 41.38 | 21.50 | 3.02 | 1.47 | 8.85 | 5.64 | 0.55 | 0.12 | 6.08 | 15.68 | 31.99 | 0.46 | 65.25 | 37.51 | 0.27 | 4.22 | 8.32 | 46.25 | 0.28 | 0.16 | 6.46 | 5.68 | 26.88 | 29.91 | 3.54 | 0.35 |
| | 34.17 | 75.15 | 47.83 | 11.94 | 3.87 | 2.23 | 6.01 | 2.50 | 0.63 | 0.13 | 5.38 | 13.56 | 32.69 | 0.56 | 66.66 | 38.77 | 0.25 | 4.18 | 7.60 | 51.89 | 0.30 | 0.14 | 5.17 | 5.31 | 15.21 | 30.66 | 4.58 | 0.27 |
| | 29.94 | 78.25 | 46.79 | 14.35 | 4.76 | 0.99 | 5.45 | 1.38 | 0.53 | 0.12 | 4.69 | 11.75 | 29.79 | 0.34 | 65.72 | 41.09 | 0.28 | 3.87 | 9.64 | 47.98 | 0.27 | 0.14 | 4.57 | 4.99 | 16.53 | 27.39 | 5.32 | 0.27 |
| | 26.93 | 71.43 | 42.27 | 14.16 | 7.14 | 1.10 | 4.47 | 1.97 | 0.64 | 0.12 | 3.60 | 8.51 | 26.09 | 0.39 | 59.73 | 35.26 | 0.28 | 3.21 | 7.68 | 39.81 | 0.22 | 0.17 | 4.25 | 3.60 | 15.33 | 24.70 | 6.41 | 0.34 |
| | 34.73 | 88.77 | 51.36 | 9.40 | 2.07 | 0.71 | 6.01 | 1.51 | 0.29 | 0.12 | 6.00 | 16.89 | 30.41 | 0.46 | 74.96 | 43.87 | 0.18 | 4.79 | 8.60 | 56.80 | 0.27 | 0.14 | 5.88 | 6.20 | 11.50 | 31.70 | 1.60 | 0.23 |
| | 29.50 | 78.90 | 48.77 | 7.96 | 2.69 | 1.30 | 5.53 | 2.86 | 0.63 | 0.10 | 5.35 | 13.97 | 31.54 | 0.50 | 68.32 | 40.58 | 0.26 | 4.31 | 8.27 | 54.40 | 0.36 | 0.14 | 5.74 | 5.91 | 10.83 | 28.63 | 3.17 | 0.29 |
| | 33.98 | 87.80 | 51.12 | 8.82 | 1.97 | 0.65 | 5.83 | 1.28 | 0.34 | 0.13 | 5.31 | 14.81 | 29.67 | 0.43 | 73.16 | 40.91 | 0.15 | 4.30 | 8.09 | 53.41 | 0.26 | 0.15 | 5.24 | 5.52 | 10.59 | 28.80 | 1.46 | 0.25 |
| | 29.77 | 86.98 | 45.11 | 7.52 | 1.93 | 0.70 | 5.32 | 2.02 | 0.35 | 0.19 | 5.34 | 14.55 | 25.75 | 0.42 | 72.67 | 36.87 | 0.13 | 4.06 | 6.97 | 51.44 | 0.22 | 0.14 | 4.61 | 4.90 | 9.11 | 27.00 | 1.38 | 0.23 |
| | 20.40 | 72.44 | 34.09 | 10.88 | 3.83 | 0.76 | 3.58 | 1.24 | 0.49 | 0.11 | 3.17 | 7.11 | 20.56 | 0.33 | 59.02 | 30.21 | 0.21 | 2.94 | 7.41 | 35.27 | 0.24 | 0.12 | 3.20 | 3.72 | 12.02 | 18.09 | 4.84 | 0.21 |
| | 20.05 | 67.29 | 32.35 | 7.54 | 4.60 | 0.72 | 4.05 | 1.45 | 0.52 | 0.14 | 3.30 | 7.65 | 18.86 | 0.29 | 56.89 | 28.11 | 0.16 | 2.94 | 7.30 | 33.17 | 0.26 | 0.12 | 3.14 | 3.35 | 10.85 | 16.36 | 4.20 | 0.22 |
| | 18.05 | 65.50 | 21.91 | 5.20 | 2.13 | 0.73 | 3.68 | 1.81 | 0.40 | 0.20 | 5.22 | 10.15 | 21.82 | 0.59 | 53.24 | 22.00 | 0.19 | 4.00 | 6.60 | 28.84 | 0.30 | 0.10 | 5.44 | 5.34 | 7.25 | 15.21 | 1.91 | 0.24 |
| | 17.10 | 68.05 | 28.64 | 4.74 | 0.28 | 0.47 | 4.04 | 0.60 | 0.32 | 0.14 | 4.50 | 10.31 | 18.47 | 0.45 | 59.35 | 26.40 | 0.16 | 3.64 | 6.72 | 37.86 | 0.23 | 0.15 | 4.65 | 4.89 | 8.09 | 15.97 | 0.23 | 0.21 |
| | 19.56 | 83.53 | 26.98 | 4.49 | 0.68 | 0.50 | 3.59 | 0.98 | 0.25 | 0.14 | 4.07 | 11.25 | 18.60 | 0.39 | 66.57 | 23.12 | 0.14 | 3.82 | 5.80 | 37.80 | 0.25 | 0.14 | 3.80 | 4.38 | 6.29 | 16.29 | 0.70 | 0.18 |
| | 20.80 | 81.44 | 28.98 | 5.79 | 0.44 | 0.42 | 4.61 | 0.56 | 0.23 | 0.12 | 4.57 | 11.05 | 17.25 | 0.37 | 63.08 | 23.55 | 0.16 | 3.44 | 7.05 | 33.58 | 0.20 | 0.09 | 4.64 | 4.46 | 7.67 | 15.63 | 0.26 | 0.21 |
| | 15.13 | 68.99 | 16.07 | 5.74 | 2.85 | 0.98 | 3.82 | 2.41 | 0.29 | 0.13 | 3.85 | 8.00 | 11.31 | 0.43 | 60.32 | 13.52 | 0.14 | 3.07 | 6.08 | 23.07 | 0.21 | 0.15 | 4.29 | 3.86 | 6.99 | 9.80 | 1.82 | 0.39 |

| | XMmi | XMnl | XMno | XMpl | XMpt | XMquz | XMro | XMru | XMsv | XMsw | XMte | XMth | XMtr | XMuk | XMvi | XMzh | WIar | WIbg | WIda | WIel | WIet | WId | WIko | WIja | WItr | WIvi | WIen | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 10.44 | 70.01 | 67.07 | 73.08 | 76.06 | 8.00 | 73.17 | 82.67 | 70.31 | 32.49 | 40.90 | 74.53 | 68.70 | 77.28 | 77.67 | 79.48 | 53.58 | 43.80 | 50.72 | 54.74 | 34.70 | 53.22 | 43.78 | 43.15 | 53.27 | 57.85 | 60.74 | 66.57 |
| | 0.71 | 68.33 | 65.69 | 72.29 | 74.59 | 5.90 | 68.14 | 81.27 | 67.89 | 19.72 | 9.95 | 39.18 | 64.99 | 69.72 | 68.59 | 63.08 | 45.16 | 41.76 | 53.55 | 37.47 | 35.80 | 60.85 | 35.04 | 37.99 | 51.41 | 59.63 | 67.86 | 59.21 |
| | 0.66 | 68.39 | 58.60 | 66.65 | 76.17 | 4.96 | 51.41 | 82.08 | 59.20 | 6.63 | 4.30 | 56.52 | 38.61 | 64.50 | 72.31 | 83.16 | 53.33 | 44.84 | 52.59 | 39.32 | 29.94 | 57.02 | 36.08 | 43.90 | 51.21 | 58.15 | 64.45 | 58.87 |
| | 0.80 | 68.36 | 57.00 | 65.66 | 75.28 | 5.41 | 48.29 | 74.78 | 63.41 | 7.61 | 0.24 | 13.63 | 53.14 | 52.27 | 40.38 | 34.23 | 34.10 | 32.52 | 53.01 | 24.51 | 31.11 | 63.11 | 22.54 | 23.41 | 55.11 | 56.87 | 74.79 | 51.11 |
| | 0.73 | 67.94 | 57.61 | 65.40 | 74.38 | 5.68 | 47.89 | 74.06 | 63.57 | 7.55 | 0.20 | 13.22 | 52.07 | 52.30 | 38.60 | 34.51 | 31.60 | 31.65 | 52.13 | 23.26 | 29.97 | 61.30 | 21.58 | 23.28 | 52.47 | 54.99 | 73.26 | 49.84 |
| | 2.02 | 45.58 | 49.48 | 48.86 | 46.98 | 4.25 | 47.20 | 51.41 | 48.40 | 23.27 | 25.74 | 46.39 | 40.31 | 48.97 | 44.09 | 46.58 | 42.36 | 38.86 | 44.28 | 43.11 | 32.05 | 47.11 | 33.97 | 34.39 | 42.55 | 49.28 | 51.77 | 46.35 |
| | 0.85 | 62.37 | 46.44 | 55.46 | 72.45 | 5.70 | 34.57 | 65.23 | 54.85 | 5.79 | 0.13 | 12.43 | 40.17 | 38.13 | 22.99 | 29.12 | 22.32 | 22.21 | 44.45 | 17.16 | 25.61 | 51.96 | 13.89 | 16.20 | 44.49 | 44.39 | 71.10 | 43.21 |
| | 0.86 | 61.92 | 45.97 | 54.62 | 72.16 | 5.59 | 34.03 | 64.97 | 54.44 | 5.85 | 0.12 | 12.49 | 39.90 | 37.41 | 23.10 | 28.77 | 21.53 | 22.72 | 44.29 | 16.64 | 25.10 | 51.13 | 13.48 | 16.13 | 43.96 | 43.49 | 69.76 | 42.55 |
| | 0.81 | 60.70 | 45.29 | 54.19 | 70.19 | 5.50 | 33.27 | 63.32 | 53.58 | 5.76 | 0.13 | 12.28 | 39.10 | 37.33 | 22.19 | 28.23 | 21.26 | 21.12 | 42.18 | 15.82 | 24.92 | 49.59 | 13.47 | 15.56 | 42.16 | 40.81 | 68.40 | 41.23 |
| | 0.73 | 60.60 | 45.09 | 53.65 | 70.64 | 5.71 | 33.52 | 63.37 | 53.23 | 5.72 | 0.11 | 12.36 | 39.06 | 37.66 | 21.48 | 28.41 | 20.32 | 21.76 | 42.33 | 15.75 | 25.08 | 50.06 | 13.10 | 15.63 | 42.04 | 41.37 | 68.31 | 41.26 |
| | 0.58 | 64.46 | 47.47 | 48.86 | 70.07 | 4.27 | 33.65 | 40.31 | 52.87 | 6.09 | 0.17 | 2.31 | 32.80 | 19.41 | 9.46 | 8.84 | 23.76 | 25.43 | 62.02 | 12.99 | 37.10 | 67.32 | 13.20 | 14.66 | 63.86 | 51.21 | 80.31 | 40.19 |
| | 0.88 | 38.17 | 29.12 | 23.21 | 50.21 | 2.73 | 19.40 | 41.98 | 32.51 | 2.78 | 0.19 | 6.91 | 10.63 | 21.89 | 4.98 | 39.55 | 20.61 | 12.01 | 38.08 | 12.47 | 20.43 | 40.12 | 10.47 | 21.15 | 33.60 | 30.15 | 58.65 | 34.96 |
| | 0.87 | 38.27 | 29.20 | 23.20 | 50.10 | 2.74 | 19.44 | 41.92 | 32.51 | 2.79 | 0.18 | 6.86 | 10.64 | 21.93 | 4.96 | 39.53 | 20.51 | 11.78 | 38.06 | 12.25 | 20.10 | 40.23 | 10.48 | 20.97 | 33.47 | 30.33 | 58.70 | 34.92 |
| | 0.75 | 44.91 | 23.46 | 13.52 | 51.09 | 4.09 | 24.85 | 9.21 | 22.91 | 3.98 | 0.14 | 2.36 | 7.52 | 3.26 | 3.65 | 5.09 | 6.77 | 10.29 | 56.83 | 10.07 | 34.12 | 60.87 | 7.65 | 14.69 | 52.20 | 47.92 | 79.50 | 28.01 |
| | 0.77 | 44.47 | 21.94 | 13.12 | 50.12 | 4.13 | 24.01 | 8.67 | 21.28 | 3.70 | 0.14 | 2.16 | 7.16 | 2.99 | 3.52 | 4.56 | 7.04 | 10.00 | 56.40 | 9.82 | 33.03 | 60.83 | 7.79 | 14.83 | 52.65 | 47.08 | 80.59 | 27.82 |
| | 0.67 | 39.83 | 18.85 | 9.98 | 45.06 | 3.70 | 18.94 | 3.83 | 18.84 | 3.90 | 0.12 | 2.19 | 6.20 | 1.71 | 2.91 | 3.99 | 6.90 | 8.24 | 53.48 | 9.23 | 32.12 | 58.12 | 7.08 | 12.91 | 49.00 | 45.52 | 78.98 | 25.92 |
| | 0.61 | 36.78 | 16.40 | 8.98 | 42.48 | 3.61 | 18.63 | 4.49 | 16.28 | 3.74 | 0.13 | 2.19 | 5.96 | 1.81 | 3.09 | 3.63 | 7.03 | 8.22 | 52.77 | 9.22 | 31.19 | 57.21 | 7.23 | 14.82 | 49.42 | 44.84 | 79.42 | 25.54 |
| | 0.72 | 36.74 | 16.82 | 9.19 | 43.04 | 3.50 | 18.47 | 4.40 | 16.76 | 3.88 | 0.13 | 2.14 | 6.15 | 1.85 | 3.10 | 3.58 | 6.87 | 8.09 | 52.44 | 9.56 | 31.65 | 57.41 | 7.61 | 13.65 | 49.28 | 44.77 | 78.73 | 25.54 |
| | 0.75 | 39.45 | 22.39 | 14.98 | 35.58 | 3.69 | 26.64 | 12.96 | 22.90 | 3.66 | 0.12 | 1.98 | 6.84 | 5.77 | 7.05 | 2.42 | 5.33 | 8.14 | 43.93 | 11.41 | 22.83 | 37.04 | 6.69 | 10.91 | 36.15 | 36.46 | 70.98 | 22.36 |
| 0K | 0.52 | 32.16 | 15.68 | 8.87 | 35.73 | 3.75 | 17.16 | 1.71 | 16.00 | 3.97 | 0.12 | 1.78 | 8.84 | 0.82 | 4.04 | 4.78 | 5.51 | 5.34 | 51.14 | 9.14 | 29.64 | 59.44 | 6.86 | 12.30 | 49.70 | 44.64 | 76.47 | 23.77 |
| | 0.63 | 30.08 | 13.29 | 6.75 | 35.87 | 3.60 | 12.83 | 1.88 | 13.61 | 3.33 | 0.12 | 2.58 | 5.39 | 0.95 | 2.40 | 1.85 | 6.73 | 5.61 | 47.55 | 8.79 | 28.78 | 53.73 | 6.58 | 13.53 | 50.73 | 39.10 | 78.60 | 23.43 |
| | 0.81 | 30.52 | 13.50 | 7.76 | 35.06 | 3.39 | 15.14 | 2.72 | 13.72 | 3.54 | 0.17 | 2.05 | 5.21 | 1.15 | 2.95 | 2.50 | 5.72 | 6.55 | 47.56 | 8.69 | 28.00 | 50.94 | 7.33 | 12.52 | 45.72 | 40.69 | 76.13 | 23.02 |
| 0K | 0.68 | 25.91 | 15.20 | 8.53 | 32.28 | 3.24 | 16.61 | 1.70 | 14.26 | 3.65 | 0.12 | 1.80 | 7.18 | 0.83 | 3.94 | 4.18 | 5.18 | 4.60 | 43.32 | 8.56 | 25.77 | 50.27 | 5.93 | 10.74 | 42.20 | 36.42 | 70.50 | 21.57 |
| 0K | 0.55 | 26.81 | 14.84 | 8.41 | 31.44 | 3.09 | 14.84 | 1.70 | 14.99 | 3.74 | 0.13 | 1.43 | 6.72 | 0.82 | 4.45 | 3.95 | 5.09 | 4.30 | 39.44 | 8.32 | 23.65 | 46.04 | 6.09 | 9.82 | 38.83 | 32.04 | 67.16 | 20.13 |
| | 0.74 | 25.91 | 13.36 | 6.44 | 29.73 | 3.02 | 14.04 | 2.27 | 13.09 | 3.32 | 0.12 | 1.78 | 4.28 | 1.09 | 2.56 | 2.25 | 5.36 | 5.56 | 42.20 | 8.05 | 24.07 | 43.14 | 5.82 | 9.61 | 38.89 | 32.82 | 71.54 | 20.13 |
| | 0.54 | 22.76 | 11.34 | 5.32 | 28.20 | 3.22 | 11.16 | 1.51 | 11.30 | 2.97 | 0.13 | 2.47 | 4.42 | 0.76 | 2.10 | 1.19 | 6.22 | 4.94 | 41.56 | 7.57 | 24.00 | 47.51 | 6.25 | 10.57 | 43.11 | 32.38 | 72.12 | 20.12 |
| | 0.56 | 20.37 | 8.36 | 5.05 | 25.77 | 2.73 | 8.73 | 1.72 | 8.33 | 3.08 | 0.12 | 2.46 | 4.12 | 0.84 | 1.86 | 1.42 | 7.59 | 5.86 | 48.24 | 8.77 | 27.93 | 53.45 | 5.78 | 14.40 | 47.65 | 34.26 | 78.47 | 20.24 |
| | 0.67 | 26.12 | 16.95 | 7.20 | 29.55 | 3.12 | 15.67 | 1.04 | 15.15 | 3.40 | 0.12 | 1.64 | 4.14 | 0.70 | 2.13 | 1.03 | 4.92 | 3.51 | 29.34 | 7.81 | 18.07 | 26.33 | 5.63 | 6.99 | 27.73 | 20.70 | 61.62 | 18.53 |
| | 0.60 | 22.19 | 13.90 | 6.32 | 27.72 | 3.16 | 14.73 | 1.74 | 13.32 | 3.47 | 0.12 | 1.88 | 3.96 | 0.89 | 2.38 | 1.92 | 5.36 | 4.71 | 32.09 | 6.91 | 19.15 | 29.56 | 6.03 | 8.19 | 29.99 | 25.92 | 62.20 | 18.09 |
| | 0.50 | 23.58 | 14.72 | 6.43 | 27.80 | 2.80 | 14.03 | 0.98 | 13.08 | 3.21 | 0.12 | 1.58 | 3.82 | 0.56 | 1.86 | 1.06 | 4.78 | 3.23 | 28.66 | 7.35 | 18.93 | 26.89 | 5.78 | 6.65 | 26.98 | 21.71 | 61.76 | 18.12 |
| | 0.54 | 23.15 | 14.08 | 5.73 | 24.97 | 2.67 | 13.31 | 0.75 | 13.39 | 2.82 | 0.13 | 1.63 | 3.68 | 0.49 | 1.67 | 1.01 | 4.23 | 3.13 | 26.70 | 7.02 | 15.11 | 24.78 | 4.87 | 7.22 | 22.81 | 19.56 | 57.75 | 16.81 |
| | 0.57 | 15.78 | 6.94 | 3.97 | 21.23 | 2.61 | 7.51 | 1.36 | 6.58 | 2.54 | 0.14 | 2.52 | 3.18 | 0.69 | 1.70 | 1.02 | 6.00 | 5.27 | 41.92 | 6.92 | 23.29 | 46.93 | 5.37 | 10.53 | 42.32 | 31.11 | 72.90 | 17.66 |
| | 0.55 | 15.47 | 7.17 | 3.76 | 19.77 | 2.63 | 6.82 | 1.18 | 7.14 | 2.43 | 0.12 | 2.24 | 3.10 | 0.62 | 1.79 | 1.05 | 6.70 | 4.82 | 39.26 | 6.66 | 21.91 | 43.30 | 4.81 | 9.06 | 38.78 | 31.37 | 71.82 | 16.73 |
| | 0.56 | 13.99 | 9.43 | 6.36 | 18.00 | 2.67 | 9.81 | 1.17 | 9.37 | 2.76 | 0.13 | 1.25 | 3.87 | 0.66 | 1.26 | 1.67 | 4.62 | 4.63 | 30.52 | 8.65 | 21.41 | 29.87 | 5.63 | 9.15 | 29.13 | 22.98 | 64.44 | 14.48 |
| | 0.45 | 16.49 | 9.56 | 5.45 | 18.77 | 2.61 | 11.22 | 0.86 | 9.90 | 3.07 | 0.11 | 1.07 | 3.22 | 0.47 | 2.12 | 0.44 | 4.59 | 2.95 | 26.57 | 6.23 | 16.53 | 24.84 | 4.55 | 5.26 | 23.80 | 21.33 | 62.25 | 13.86 |
| | 0.51 | 16.32 | 10.89 | 4.47 | 15.03 | 2.23 | 9.66 | 0.79 | 9.65 | 2.30 | 0.13 | 1.53 | 2.81 | 0.44 | 1.04 | 0.71 | 3.57 | 2.42 | 21.77 | 6.53 | 13.13 | 19.41 | 4.02 | 5.95 | 20.78 | 14.54 | 52.31 | 13.44 |
| | 0.56 | 16.66 | 10.37 | 5.23 | 17.55 | 2.71 | 11.04 | 0.43 | 10.54 | 2.75 | 0.12 | 0.81 | 3.01 | 0.38 | 1.84 | 0.41 | 3.74 | 2.85 | 17.37 | 5.51 | 11.58 | 16.00 | 4.08 | 5.59 | 16.18 | 14.37 | 50.48 | 13.05 |
| | 0.43 | 10.69 | 7.36 | 4.01 | 11.31 | 1.82 | 7.29 | 1.03 | 7.42 | 1.72 | 0.13 | 1.23 | 2.74 | 0.43 | 1.21 | 1.84 | 3.69 | 3.43 | 20.52 | 6.22 | 16.28 | 22.30 | 5.00 | 7.70 | 22.60 | 17.40 | 52.94 | 12.25 |

Table 8.12: **Multilingual Retrieval Results.** The average is the aggregated average of the 3 big tasks.

| model name | STS12 | STS13 | STS14 | STS15 | STS16 | STS17 | STS-b | mean |
|---|---|---|---|---|---|---|---|---|
| voyage-multimodal-3 | 71.62 | 81.60 | 77.98 | 86.85 | 82.62 | 89.68 | 82.55 | 81.84 |
| royokong/e5-v | 73.15 | 78.18 | 74.88 | 84.22 | 79.45 | 85.84 | 79.40 | 79.30 |
| TIGER-Lab/VLM2Vec-Full | 71.15 | 65.88 | 62.63 | 76.00 | 75.36 | 83.72 | 73.75 | 72.64 |
| TIGER-Lab/VLM2Vec-LoRA | 71.18 | 65.87 | 62.61 | 75.92 | 75.34 | 83.55 | 73.64 | 72.59 |
| EVA02-CLIP-bigE-14-plus | 63.36 | 68.00 | 66.38 | 79.45 | 75.26 | 82.87 | 68.59 | 71.99 |
| laion/CLIP-ViT-bigG-14-laion2B-39B-b160k | 62.81 | 68.16 | 65.50 | 78.67 | 74.89 | 79.97 | 66.54 | 70.93 |
| laion/CLIP-ViT-L-14-DataComp.XL-s13B-b90K | 62.36 | 67.64 | 64.25 | 77.36 | 73.48 | 80.63 | 63.38 | 69.87 |
| google/siglip-large-patch16-384 | 66.30 | 62.08 | 61.66 | 77.11 | 73.27 | 79.58 | 66.59 | 69.51 |
| laion/CLIP-ViT-g-14-laion2B-s34B-b88K | 61.85 | 66.43 | 62.32 | 76.73 | 72.67 | 79.88 | 64.13 | 69.14 |
| EVA02-CLIP-bigE-14 | 62.24 | 62.36 | 62.17 | 77.41 | 73.63 | 80.96 | 62.85 | 68.80 |
| laion/CLIP-ViT-B-16-DataComp.XL-s13B-b90K | 64.19 | 63.81 | 62.34 | 75.48 | 69.90 | 80.04 | 63.51 | 68.47 |
| google/siglip-so400m-patch14-384 | 61.90 | 62.95 | 60.58 | 76.17 | 73.48 | 78.41 | 62.63 | 68.02 |
| google/siglip-base-patch16-512 | 64.97 | 59.10 | 61.13 | 75.08 | 71.27 | 80.09 | 62.21 | 67.69 |
| google/siglip-large-patch16-256 | 63.94 | 59.44 | 59.35 | 75.74 | 71.83 | 79.21 | 62.50 | 67.43 |
| google/siglip-base-patch16-384 | 64.62 | 59.38 | 61.17 | 74.34 | 70.29 | 79.27 | 60.28 | 67.05 |
| Salesforce/blip-itm-base-coco | 62.91 | 55.14 | 60.17 | 72.83 | 71.59 | 77.32 | 66.50 | 66.64 |
| google/siglip-base-patch16-256 | 65.01 | 58.02 | 60.36 | 74.25 | 69.09 | 78.73 | 57.65 | 66.16 |
| openai/clip-vit-base-patch16 | 63.82 | 63.26 | 56.99 | 73.32 | 68.91 | 78.18 | 57.93 | 66.06 |
| laion/CLIP-ViT-L-14-laion2B-s32B-b82K | 57.52 | 62.75 | 59.94 | 74.55 | 70.61 | 75.92 | 59.43 | 65.82 |
| laion/CLIP-ViT-H-14-laion2B-s32B-b79K | 57.00 | 62.25 | 58.62 | 74.40 | 70.57 | 76.69 | 58.99 | 65.50 |
| google/siglip-base-patch16-256-multilingual | 66.62 | 54.80 | 59.00 | 72.65 | 68.33 | 80.53 | 56.29 | 65.46 |
| openai/clip-vit-large-patch14 | 53.89 | 66.78 | 55.98 | 72.03 | 70.49 | 75.26 | 56.74 | 64.45 |
| Salesforce/blip-itm-base-flickr | 59.24 | 54.45 | 57.87 | 71.10 | 68.17 | 75.97 | 62.93 | 64.25 |
| google/siglip-base-patch16-224 | 63.19 | 55.40 | 57.99 | 73.07 | 67.79 | 77.78 | 54.50 | 64.25 |
| BAAI/bge-visualized-m3 | 63.93 | 56.91 | 58.19 | 70.94 | 63.49 | 79.18 | 56.48 | 64.16 |
| EVA02-CLIP-L-14 | 53.75 | 60.82 | 57.12 | 71.53 | 67.46 | 80.14 | 50.46 | 63.04 |
| Salesforce/blip-itm-large-coco | 62.32 | 50.97 | 55.16 | 70.15 | 67.33 | 75.69 | 58.53 | 62.88 |
| kakaobrain/align-base | 53.17 | 57.50 | 56.01 | 69.13 | 66.43 | 77.55 | 59.42 | 62.74 |
| jinaai/jina-clip-v1 | 57.96 | 55.80 | 56.95 | 70.52 | 67.98 | 76.94 | 52.18 | 62.62 |
| Salesforce/blip-image-captioning-large | 61.67 | 50.18 | 54.12 | 70.03 | 66.63 | 76.43 | 56.41 | 62.21 |
| BAAI/bge-visualized-base | 55.35 | 57.31 | 57.57 | 68.27 | 62.39 | 75.52 | 54.66 | 61.58 |
| Salesforce/blip-itm-large-flickr | 59.68 | 47.46 | 52.82 | 68.29 | 64.20 | 72.77 | 55.86 | 60.16 |
| laion/CLIP-ViT-B-32-laion2B-s34B-b79K | 53.70 | 57.16 | 52.74 | 66.64 | 61.30 | 74.69 | 50.48 | 59.53 |
| laion/CLIP-ViT-B-32-DataComp.XL-s13B-b90K | 54.86 | 45.82 | 48.85 | 64.02 | 59.62 | 73.31 | 49.15 | 56.52 |
| Salesforce/blip-image-captioning-base | 49.34 | 46.84 | 48.29 | 60.57 | 60.54 | 72.56 | 49.54 | 55.38 |
| openai/clip-vit-base-patch32 | 53.81 | 52.50 | 43.69 | 59.56 | 53.01 | 71.01 | 47.17 | 54.39 |
| blip2-finetune-coco | 41.36 | 38.11 | 38.33 | 54.36 | 46.06 | 62.61 | 39.18 | 45.72 |
| EVA02-CLIP-B-16 | 40.25 | 36.57 | 39.17 | 51.18 | 48.86 | 54.15 | 31.58 | 43.11 |
| blip2-pretrain | 38.81 | 38.72 | 35.57 | 51.81 | 42.20 | 58.49 | 33.68 | 42.75 |
| nomic-ai/nomic-embed-vision-v1.5 | 40.13 | 21.22 | 21.44 | 27.21 | 31.39 | 40.46 | 23.16 | 29.29 |
| facebook/dinov2-base | 24.13 | 5.82 | 0.36 | 13.75 | 18.05 | 43.02 | 12.65 | 16.83 |
| facebook/dinov2-giant | 24.34 | 1.18 | 1.06 | 13.65 | 18.43 | 37.07 | 9.48 | 15.03 |
| nyu-visionx/moco-v3-vit-b | 24.03 | 2.19 | 0.63 | 11.48 | 19.80 | 39.54 | 6.56 | 14.89 |
| nyu-visionx/moco-v3-vit-l | 20.90 | 3.38 | -1.13 | 12.99 | 22.00 | 40.27 | 4.70 | 14.73 |
| facebook/dinov2-large | 17.45 | 0.05 | -2.39 | 12.28 | 19.35 | 43.67 | 6.31 | 13.82 |
| facebook/dinov2-small | 13.39 | 2.39 | -1.9 | 12.02 | 16.47 | 43.1 | 5.37 | 12.98 |

Table 8.13: **Visual STS English Results.** Note that for STS-17 and STS-b, we only average the English subset here.

| model name | ko-ko | ar-ar | en-ar | en-de | en-tr | es-en | es-es | fr-en | it-en | nl-en | mean |
|---|---|---|---|---|---|---|---|---|---|---|---|
| voyage-multimodal-3 | 62.80 | 65.75 | 34.40 | 80.42 | 44.98 | 74.72 | 83.70 | 75.07 | 77.76 | 78.25 | 67.79 |
| royokong/e5-v | 14.45 | 32.52 | 11.28 | 53.00 | 23.88 | 51.92 | 74.42 | 44.98 | 44.50 | 54.29 | 40.52 |
| google/siglip-so400m-patch14-384 | 13.65 | 45.76 | 11.22 | 46.07 | 30.62 | 40.08 | 73.62 | 46.36 | 36.45 | 44.95 | 38.88 |
| openai/clip-vit-large-patch14 | 11.07 | 39.12 | 18.95 | 45.71 | 39.70 | 36.76 | 70.11 | 44.06 | 40.17 | 41.63 | 38.73 |
| TIGER-Lab/VLM2Vec-Full | 17.96 | 36.74 | 7.10 | 36.47 | 16.96 | 46.72 | 72.95 | 44.67 | 35.48 | 42.37 | 35.74 |
| TIGER-Lab/VLM2Vec-LoRA | 17.99 | 37.24 | 6.54 | 36.42 | 16.69 | 47.32 | 72.77 | 44.70 | 35.33 | 42.40 | 35.74 |
| google/siglip-base-patch16-256-multilingual | 16.64 | 28.48 | 1.67 | 45.64 | 20.73 | 47.14 | 73.28 | 41.91 | 40.14 | 38.85 | 35.45 |
| google/siglip-large-patch16-384 | 15.67 | 32.51 | 14.53 | 35.06 | 30.00 | 39.64 | 72.06 | 35.62 | 33.52 | 33.19 | 34.18 |
| google/siglip-base-patch16-512 | 23.37 | 29.28 | 16.98 | 37.38 | 25.21 | 34.41 | 71.24 | 38.53 | 28.49 | 34.33 | 33.92 |
| google/siglip-base-patch16-384 | 23.08 | 34.56 | 17.04 | 35.29 | 22.75 | 32.58 | 72.25 | 37.39 | 27.05 | 32.36 | 33.43 |
| google/siglip-large-patch16-256 | 16.00 | 31.32 | 16.79 | 31.32 | 18.98 | 36.14 | 71.79 | 34.93 | 40.67 | 36.17 | 33.41 |
| laion/CLIP-ViT-L-14-DataComp.XL-s13B-b90K | 14.28 | 36.47 | 12.75 | 43.10 | 19.70 | 37.37 | 71.62 | 36.88 | 30.78 | 30.76 | 33.37 |
| openai/clip-vit-base-patch16 | 10.54 | 36.25 | 13.13 | 41.57 | 35.42 | 24.63 | 62.95 | 38.72 | 31.40 | 38.63 | 33.32 |
| laion/CLIP-ViT-H-14-laion2B-s32B-b79K | 19.39 | 33.39 | 19.49 | 43.78 | 16.68 | 27.99 | 62.58 | 39.32 | 28.59 | 37.33 | 32.85 |
| laion/CLIP-ViT-bigG-14-laion2B-39B-b160k | 14.38 | 32.39 | 12.21 | 36.74 | 14.99 | 30.44 | 69.77 | 39.77 | 36.44 | 34.83 | 32.20 |
| Salesforce/blip-itm-large-coco | 19.71 | 30.04 | 17.25 | 41.46 | 21.80 | 29.98 | 60.52 | 27.65 | 29.31 | 41.20 | 31.89 |
| google/siglip-base-patch16-256 | 21.40 | 30.46 | 12.67 | 30.19 | 19.81 | 28.50 | 71.68 | 36.55 | 28.75 | 30.72 | 31.07 |
| google/siglip-base-patch16-224 | 21.00 | 25.03 | 14.36 | 31.20 | 24.80 | 29.32 | 69.85 | 35.70 | 27.46 | 28.98 | 30.77 |
| Salesforce/blip-itm-base-flickr | 19.73 | 35.78 | 9.69 | 38.30 | 9.73 | 22.46 | 66.17 | 36.40 | 23.99 | 40.03 | 30.23 |
| Salesforce/blip-image-captioning-large | 19.14 | 32.45 | 11.21 | 36.68 | 16.77 | 23.02 | 62.57 | 27.84 | 25.01 | 39.19 | 29.39 |
| Salesforce/blip-itm-base-coco | 22.40 | 32.47 | 0.66 | 38.33 | 15.47 | 20.01 | 69.76 | 28.71 | 27.11 | 35.35 | 29.03 |
| jinaai/jina-clip-v1 | 19.32 | 27.80 | 7.55 | 31.29 | 2.29 | 29.59 | 67.75 | 24.06 | 24.69 | 34.41 | 26.88 |
| Salesforce/blip-itm-large-flickr | 22.30 | 30.66 | 8.47 | 32.23 | 6.44 | 27.43 | 54.65 | 24.72 | 24.76 | 36.65 | 26.83 |
| EVA02-CLIP-bigE-14 | 10.97 | 29.99 | 13.49 | 22.76 | 6.39 | 29.03 | 57.16 | 36.66 | 33.43 | 26.16 | 26.60 |
| laion/CLIP-ViT-g-14-laion2B-s34B-b88K | 17.17 | 29.93 | 14.27 | 28.50 | -4.79 | 34.19 | 66.07 | 29.70 | 29.02 | 21.18 | 26.52 |
| kakaobrain/align-base | 17.69 | 17.70 | 21.55 | 21.27 | 19.33 | 28.31 | 54.37 | 34.11 | 30.89 | 19.58 | 26.48 |
| EVA02-CLIP-bigE-14-plus | 11.36 | 31.51 | 10.71 | 24.33 | -10.05 | 20.18 | 59.20 | 36.12 | 28.60 | 33.18 | 24.52 |
| facebook/dinov2-small | 14.31 | 32.77 | 12.52 | 31.11 | 25.47 | 11.11 | 35.33 | 20.38 | 27.78 | 29.28 | 24.01 |
| blip2-finetune-coco | 14.35 | 38.72 | 7.17 | 25.18 | 7.01 | 20.48 | 39.44 | 30.76 | 22.85 | 32.43 | 23.84 |
| nyu-visionx/moco-v3-vit-l | 14.19 | 30.79 | 6.57 | 32.83 | 26.85 | 12.51 | 37.19 | 25.41 | 25.19 | 22.88 | 23.44 |
| Salesforce/blip-image-captioning-base | 28.34 | 31.68 | -0.59 | 22.27 | 5.81 | 16.40 | 56.30 | 17.16 | 23.48 | 27.69 | 22.85 |
| EVA02-CLIP-L-14 | 14.77 | 29.65 | 18.89 | 3.52 | 16.61 | 12.23 | 45.55 | 32.61 | 30.84 | 23.63 | 22.83 |
| facebook/dinov2-large | 21.28 | 28.50 | 17.80 | 28.70 | 27.26 | 12.43 | 39.85 | 18.48 | 18.46 | 15.34 | 22.81 |
| nyu-visionx/moco-v3-vit-b | 13.96 | 32.60 | 19.96 | 29.48 | 20.01 | 15.71 | 32.47 | 23.99 | 20.73 | 18.86 | 22.78 |
| laion/CLIP-ViT-B-16-DataComp.XL-s13B-b90K | 19.21 | 18.40 | -1.69 | 33.07 | 6.57 | 16.93 | 62.39 | 20.93 | 19.40 | 32.23 | 22.74 |
| laion/CLIP-ViT-B-32-laion2B-s34B-b79K | 16.25 | 21.73 | 4.20 | 17.82 | 17.37 | 25.07 | 57.03 | 22.91 | 21.49 | 23.38 | 22.72 |
| laion/CLIP-ViT-L-14-laion2B-s32B-b82K | 18.23 | 20.71 | 4.66 | 19.38 | 0.88 | 19.49 | 61.89 | 31.63 | 27.75 | 18.38 | 22.30 |
| openai/clip-vit-base-patch32 | 18.10 | 28.30 | 8.25 | 22.15 | 17.97 | 12.15 | 47.56 | 19.48 | 22.74 | 25.05 | 22.18 |
| blip2-pretrain | 15.88 | 28.99 | 11.13 | 23.70 | 1.60 | 21.98 | 42.55 | 26.16 | 20.60 | 25.92 | 21.85 |
| BAAI/bge-visualized-m3 | 14.76 | 18.55 | -6.92 | 30.64 | 6.53 | 8.45 | 45.41 | 34.38 | 34.44 | 30.03 | 21.63 |
| BAAI/bge-visualized-base | 19.12 | 23.67 | -1.90 | 17.37 | 3.89 | 2.68 | 50.85 | 27.90 | 25.82 | 35.52 | 20.49 |
| facebook/dinov2-base | 17.94 | 28.39 | 18.25 | 28.90 | 14.41 | 8.91 | 35.40 | 11.87 | 20.30 | 16.51 | 20.09 |
| facebook/dinov2-giant | 12.60 | 28.87 | 7.33 | 24.60 | 18.36 | 11.10 | 30.99 | 11.90 | 16.65 | 9.77 | 17.22 |
| laion/CLIP-ViT-B-32-DataComp.XL-s13B-b90K | 17.88 | 13.79 | 1.29 | 15.83 | -1.65 | 17.27 | 53.68 | 17.71 | 19.68 | 11.65 | 16.71 |
| EVA02-CLIP-B-16 | 18.02 | 19.60 | 3.58 | 7.10 | 22.06 | 4.52 | 48.86 | 12.82 | 17.07 | 11.99 | 16.56 |
| nomic-ai/nomic-embed-vision-v1.5 | 19.97 | 19.17 | -4.26 | -7.82 | -14.73 | -6.12 | 38.29 | -4.65 | 6.36 | -2.8 | 4.34 |

Table 8.14: **Visual STS cross-lingual Results.**

| model name | de | es | fr | it | nl | pl | pt | ru | zh | mean |
|---|---|---|---|---|---|---|---|---|---|---|
| voyage-multimodal-3 | 74.13 | 75.99 | 74.43 | 73.96 | 71.34 | 68.83 | 73.48 | 72.68 | 72.60 | 73.05 |
| royokong/e5-v | 58.29 | 64.24 | 61.79 | 64.11 | 55.15 | 52.17 | 63.59 | 35.88 | 12.57 | 51.98 |
| TIGER-Lab/VLM2Vec-LoRA | 52.69 | 60.83 | 58.64 | 52.77 | 49.55 | 45.77 | 55.09 | 45.43 | 17.35 | 48.68 |
| TIGER-Lab/VLM2Vec-Full | 52.65 | 60.78 | 58.68 | 52.63 | 49.62 | 45.78 | 55.06 | 45.39 | 17.22 | 48.65 |
| Salesforce/blip-itm-base-coco | 55.58 | 53.93 | 59.40 | 50.63 | 53.46 | 51.69 | 53.05 | 34.62 | 20.94 | 48.14 |
| Salesforce/blip-itm-base-flickr | 54.46 | 50.91 | 56.04 | 49.89 | 50.94 | 48.19 | 50.37 | 32.37 | 19.32 | 45.83 |
| google/siglip-large-patch16-384 | 55.72 | 56.23 | 54.78 | 54.24 | 42.45 | 41.24 | 51.62 | 36.86 | 14.97 | 45.35 |
| google/siglip-base-patch16-256-multilingual | 48.11 | 53.45 | 51.69 | 51.65 | 41.15 | 48.08 | 46.85 | 51.42 | 13.48 | 45.10 |
| google/siglip-so400m-patch14-384 | 50.73 | 56.23 | 54.72 | 51.56 | 45.65 | 35.84 | 45.88 | 39.68 | 14.86 | 43.91 |
| google/siglip-large-patch16-256 | 53.23 | 52.39 | 50.98 | 50.46 | 40.22 | 42.55 | 45.53 | 37.50 | 12.38 | 42.80 |
| google/siglip-base-patch16-512 | 45.18 | 49.45 | 53.46 | 47.26 | 43.27 | 44.95 | 43.68 | 39.47 | 14.14 | 42.32 |
| jinaai/jina-clip-v1 | 47.25 | 47.42 | 53.08 | 48.58 | 47.44 | 47.15 | 44.23 | 34.84 | 10.67 | 42.30 |
| google/siglip-base-patch16-384 | 45.57 | 48.15 | 51.66 | 45.55 | 42.49 | 44.71 | 43.36 | 36.71 | 16.70 | 41.66 |
| laion/CLIP-ViT-L-14-DataComp.XL-s13B-b90K | 47.05 | 45.13 | 50.76 | 44.24 | 38.21 | 34.94 | 37.87 | 30.89 | 14.65 | 38.19 |
| google/siglip-base-patch16-256 | 42.40 | 44.36 | 46.72 | 41.73 | 38.72 | 42.34 | 39.56 | 35.01 | 9.34 | 37.80 |
| laion/CLIP-ViT-bigG-14-laion2B-39B-b160k | 38.00 | 43.63 | 52.36 | 44.84 | 34.84 | 33.19 | 37.51 | 28.43 | 19.19 | 36.89 |
| laion/CLIP-ViT-g-14-laion2B-s34B-b88K | 48.01 | 41.47 | 45.03 | 37.56 | 36.84 | 36.02 | 32.73 | 30.53 | 23.65 | 36.87 |
| google/siglip-base-patch16-224 | 40.38 | 41.80 | 45.75 | 37.90 | 37.64 | 42.65 | 37.01 | 32.81 | 10.79 | 36.30 |
| Salesforce/blip-itm-large-coco | 42.62 | 36.03 | 43.42 | 39.51 | 37.83 | 39.55 | 32.01 | 32.30 | 16.21 | 35.50 |
| laion/CLIP-ViT-H-14-laion2B-s32B-b79K | 41.31 | 39.11 | 48.44 | 34.22 | 34.48 | 33.20 | 32.09 | 26.94 | 23.88 | 34.85 |
| laion/CLIP-ViT-B-16-DataComp.XL-s13B-b90K | 41.25 | 31.76 | 45.92 | 34.60 | 35.79 | 40.38 | 36.57 | 26.67 | 15.18 | 34.24 |
| Salesforce/blip-itm-large-flickr | 40.76 | 33.39 | 41.04 | 39.40 | 34.23 | 40.14 | 28.92 | 30.72 | 18.93 | 34.17 |
| EVA02-CLIP-bigE-14-plus | 31.96 | 37.53 | 46.88 | 38.94 | 29.78 | 27.50 | 33.35 | 25.05 | 16.20 | 31.91 |
| openai/clip-vit-large-patch14 | 37.50 | 44.18 | 47.53 | 36.89 | 32.51 | 23.41 | 35.49 | 14.06 | 12.12 | 31.52 |
| laion/CLIP-ViT-B-32-DataComp.XL-s13B-b90K | 38.22 | 28.92 | 38.00 | 23.87 | 32.90 | 43.21 | 28.62 | 27.29 | 13.95 | 30.55 |
| EVA02-CLIP-bigE-14 | 37.10 | 35.37 | 41.49 | 31.98 | 28.04 | 25.33 | 30.62 | 25.35 | 14.58 | 29.98 |
| laion/CLIP-ViT-L-14-laion2B-s32B-b82K | 39.99 | 31.22 | 40.69 | 28.57 | 28.49 | 27.58 | 25.85 | 22.66 | 22.58 | 29.74 |
| laion/CLIP-ViT-B-32-laion2B-s34B-b79K | 41.43 | 26.40 | 35.96 | 28.13 | 29.75 | 34.85 | 28.60 | 21.84 | 19.50 | 29.61 |
| BAAI/bge-visualized-base | 32.40 | 28.99 | 37.14 | 29.10 | 31.45 | 36.66 | 29.16 | 20.91 | 15.35 | 29.02 |
| EVA02-CLIP-B-16 | 30.68 | 27.02 | 36.05 | 27.13 | 29.71 | 32.41 | 29.06 | 25.40 | 16.71 | 28.24 |
| kakaobrain/align-base | 34.60 | 25.79 | 38.57 | 26.95 | 32.79 | 28.88 | 22.60 | 23.02 | 19.63 | 28.09 |
| openai/clip-vit-base-patch16 | 32.72 | 30.81 | 39.06 | 29.46 | 23.46 | 28.15 | 26.30 | 14.69 | 11.85 | 26.28 |
| BAAI/bge-visualized-m3 | 32.04 | 22.26 | 36.13 | 27.05 | 24.47 | 27.96 | 26.80 | 17.00 | 14.02 | 25.30 |
| nomic-ai/nomic-embed-vision-v1.5 | 29.92 | 23.12 | 23.35 | 22.93 | 21.92 | 30.96 | 20.85 | 25.16 | 16.55 | 23.86 |
| EVA02-CLIP-L-14 | 22.00 | 28.24 | 33.36 | 22.67 | 21.75 | 21.31 | 18.91 | 16.84 | 14.04 | 22.12 |
| blip2-finetune-coco | 24.59 | 20.31 | 27.19 | 21.90 | 21.13 | 25.57 | 22.40 | 19.31 | 14.71 | 21.90 |
| blip2-pretrain | 22.79 | 19.74 | 31.75 | 23.77 | 17.54 | 26.10 | 24.49 | 17.69 | 11.26 | 21.68 |
| openai/clip-vit-base-patch32 | 29.41 | 23.43 | 26.85 | 20.55 | 19.05 | 30.37 | 14.01 | 18.59 | 10.52 | 21.42 |
| facebook/dinov2-base | 23.73 | 16.42 | 19.46 | 16.06 | 17.90 | 21.00 | 11.48 | 17.09 | 23.32 | 18.50 |
| facebook/dinov2-giant | 21.10 | 16.14 | 22.06 | 12.74 | 16.13 | 21.79 | 16.47 | 17.30 | 19.34 | 18.12 |
| facebook/dinov2-large | 20.60 | 13.14 | 20.47 | 10.65 | 15.91 | 19.60 | 11.28 | 19.78 | 22.36 | 17.09 |
| nyu-visionx/moco-v3-vit-l | 14.76 | 9.48 | 15.35 | 8.17 | 11.68 | 16.31 | 11.01 | 12.31 | 20.54 | 13.29 |
| nyu-visionx/moco-v3-vit-b | 12.98 | 9.99 | 15.34 | 6.77 | 12.62 | 14.02 | 10.74 | 13.33 | 18.30 | 12.68 |
| facebook/dinov2-small | 10.87 | 11.58 | 12.94 | 6.70 | 9.17 | 13.27 | 8.18 | 9.96 | 17.97 | 11.18 |

Table 8.15: **Visual STS multilingual Results.**

| Model name | ArxivQA | DocVQA | InfoVQA | Shift Project | Syn.Doc QAAI | Syn.Doc QAEnergy | Syn.Doc QAGov. | Syn.Dic QAHealth. | Syn. Tabfquad | Tatdqa | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|
| voyage-multimodal-3 | 84.61 | 49.76 | 86.11 | 77.48 | 83.56 | 79.42 | 83.92 | 82.39 | 56.36 | 27.63 | 71.13 |
| royokong/e5-v | 48.27 | 34.73 | 69.22 | 42.47 | 78.91 | 78.11 | 82.16 | 82.31 | 81.37 | 29.32 | 62.69 |
| google/siglip-so400m-patch14-384 | 50.21 | 31.28 | 69.73 | 25.04 | 67.78 | 73.52 | 75.35 | 83.10 | 60.29 | 27.52 | 56.38 |
| google/siglip-large-patch16-384 | 47.45 | 28.53 | 64.11 | 25.37 | 64.87 | 67.34 | 74.52 | 74.67 | 61.09 | 25.26 | 53.32 |
| google/siglip-base-patch16-512 | 46.02 | 28.38 | 64.51 | 22.95 | 63.51 | 66.79 | 72.79 | 79.70 | 50.71 | 25.30 | 52.06 |
| TIGER-Lab/VLM2Vec-Full | 42.84 | 26.74 | 66.68 | 25.01 | 53.51 | 63.49 | 64.03 | 70.73 | 63.54 | 21.45 | 49.80 |
| TIGER-Lab/VLM2Vec-LoRA | 42.59 | 26.92 | 67.64 | 24.34 | 54.02 | 63.35 | 64.06 | 70.62 | 61.93 | 21.61 | 49.71 |
| google/siglip-base-patch16-384 | 43.59 | 26.43 | 59.28 | 14.46 | 55.75 | 57.47 | 58.54 | 67.67 | 47.61 | 19.19 | 45.00 |
| laion/CLIP-ViT-bigG-14-laion2B-39B-b160k | 38.84 | 20.44 | 60.90 | 25.02 | 55.42 | 59.95 | 62.27 | 57.86 | 35.02 | 16.21 | 43.19 |
| laion/CLIP-ViT-H-14-laion2B-s32B-b79K | 33.03 | 19.14 | 58.82 | 21.81 | 54.09 | 60.23 | 52.92 | 55.50 | 33.11 | 15.41 | 40.41 |
| google/siglip-large-patch16-256 | 40.19 | 22.39 | 54.09 | 9.13 | 43.40 | 50.79 | 55.45 | 56.03 | 49.81 | 12.38 | 39.37 |
| laion/CLIP-ViT-L-14-DataComp.XL-s13B-b90K | 34.51 | 19.68 | 55.61 | 16.19 | 47.20 | 58.93 | 50.28 | 58.04 | 30.70 | 15.27 | 38.64 |
| openai/clip-vit-large-patch14 | 28.64 | 16.69 | 62.44 | 17.05 | 38.25 | 61.62 | 52.84 | 60.23 | 30.95 | 11.00 | 37.97 |
| laion/CLIP-ViT-g-14-laion2B-s34B-b88K | 32.82 | 18.10 | 56.85 | 16.72 | 40.12 | 60.07 | 52.21 | 52.09 | 32.51 | 14.80 | 37.63 |
| laion/CLIP-ViT-L-14-laion2B-s32B-b82K | 30.96 | 17.79 | 52.46 | 13.10 | 44.98 | 57.08 | 49.29 | 53.15 | 29.13 | 14.68 | 36.26 |
| EVA02-CLIP-bigE-14-plus | 34.86 | 16.84 | 55.19 | 12.76 | 34.57 | 44.99 | 43.14 | 42.47 | 30.36 | 7.52 | 32.27 |
| google/siglip-base-patch16-256 | 35.17 | 19.42 | 48.73 | 5.45 | 31.06 | 41.28 | 40.07 | 49.94 | 37.00 | 8.50 | 31.66 |
| EVA02-CLIP-bigE-14 | 32.72 | 16.35 | 54.80 | 10.14 | 33.53 | 48.50 | 41.32 | 42.98 | 28.80 | 7.09 | 31.62 |
| kakaobrain/align-base | 23.31 | 18.03 | 43.15 | 10.47 | 41.43 | 49.76 | 42.07 | 47.46 | 29.00 | 9.69 | 31.44 |
| laion/CLIP-ViT-B-16-DataComp.XL-s13B-b90K | 28.88 | 13.97 | 46.88 | 7.25 | 32.17 | 38.53 | 31.05 | 35.83 | 26.60 | 9.07 | 27.02 |
| google/siglip-base-patch16-256-multilingual | 30.33 | 16.96 | 45.28 | 3.89 | 22.72 | 29.93 | 28.73 | 37.17 | 44.16 | 4.38 | 26.35 |
| google/siglip-base-patch16-224 | 31.49 | 16.04 | 46.11 | 3.71 | 25.27 | 35.53 | 32.35 | 37.01 | 29.04 | 5.08 | 26.16 |
| openai/clip-vit-base-patch16 | 26.54 | 14.60 | 51.70 | 7.13 | 22.86 | 32.43 | 39.84 | 37.54 | 17.61 | 4.71 | 25.50 |
| EVA02-CLIP-L-14 | 30.44 | 11.24 | 48.48 | 4.44 | 20.36 | 29.87 | 18.37 | 33.68 | 20.64 | 3.28 | 22.08 |
| Salesforce/blip-itm-large-flickr | 24.89 | 12.11 | 33.95 | 4.66 | 17.40 | 23.16 | 16.14 | 27.18 | 21.87 | 3.33 | 18.47 |
| Salesforce/blip-itm-base-coco | 20.55 | 11.68 | 32.30 | 5.05 | 18.70 | 18.68 | 24.74 | 25.58 | 19.42 | 3.42 | 18.01 |
| Salesforce/blip-itm-large-coco | 22.65 | 11.25 | 31.37 | 4.08 | 16.22 | 19.03 | 19.85 | 24.45 | 24.59 | 3.50 | 17.70 |
| jinaai/jina-clip-v1 | 25.40 | 10.99 | 35.12 | 3.84 | 15.57 | 19.34 | 21.83 | 20.84 | 20.14 | 3.34 | 17.64 |
| laion/CLIP-ViT-B-32-laion2B-s34B-b79K | 24.01 | 10.50 | 35.35 | 4.95 | 18.39 | 22.52 | 14.50 | 18.60 | 16.09 | 3.66 | 16.86 |
| blip2-finetune-coco | 14.68 | 10.37 | 31.97 | 4.08 | 13.78 | 18.23 | 17.47 | 23.95 | 17.60 | 3.80 | 15.59 |
| Salesforce/blip-itm-base-flickr | 17.06 | 10.81 | 29.65 | 4.50 | 14.73 | 15.23 | 15.23 | 20.40 | 19.33 | 2.71 | 14.96 |
| openai/clip-vit-base-patch32 | 17.11 | 9.48 | 37.15 | 1.00 | 11.06 | 18.31 | 9.14 | 13.14 | 14.09 | 1.86 | 13.23 |
| laion/CLIP-ViT-B-32-DataComp.XL-s13B-b90K | 16.57 | 9.03 | 27.09 | 3.06 | 13.62 | 15.67 | 9.08 | 12.51 | 14.92 | 2.76 | 12.43 |
| BAAI/bge-visualized-m3 | 18.10 | 8.26 | 32.79 | 1.39 | 9.91 | 8.91 | 8.41 | 12.58 | 21.61 | 1.81 | 12.38 |
| blip2-pretrain | 11.25 | 5.74 | 30.92 | 4.12 | 16.04 | 15.05 | 10.66 | 14.42 | 12.53 | 2.31 | 12.30 |
| nomic-ai/nomic-embed-vision-v1.5 | 15.86 | 9.25 | 29.55 | 0.00 | 11.10 | 10.94 | 8.90 | 15.79 | 15.20 | 2.61 | 11.92 |
| BAAI/bge-visualized-base | 15.20 | 7.05 | 29.64 | 3.02 | 7.34 | 11.05 | 6.91 | 9.39 | 11.83 | 1.94 | 10.34 |
| EVA02-CLIP-B-16 | 16.22 | 5.84 | 25.13 | 1.43 | 8.19 | 9.58 | 5.26 | 10.35 | 10.88 | 1.30 | 9.42 |

Table 8.16: **Document Understanding Results.**

| model name | CIFAR10 | DTD | EuroSAT | FER2013 | GTSRB | MNIST | PatchCamelyon | STL10 | VOC2007 | mean |
|---|---|---|---|---|---|---|---|---|---|---|
| EVA02-CLIP-bigE-14-plus | 99.50 | 81.14 | 93.86 | 50.84 | 88.99 | 92.79 | 76.48 | 99.76 | 91.68 | 86.11 |
| google/siglip-so400m-patch14-384 | 96.92 | 80.81 | 88.97 | 47.41 | 86.39 | 96.11 | 75.41 | 99.51 | 92.40 | 84.88 |
| google/siglip-large-patch16-384 | 96.74 | 80.47 | 89.62 | 46.20 | 85.76 | 96.21 | 77.31 | 99.33 | 92.23 | 84.87 |
| laion/CLIP-ViT-bigG-14-laion2B-39B-b160k | 98.42 | 79.50 | 92.22 | 47.30 | 87.69 | 96.12 | 71.25 | 99.53 | 91.81 | 84.87 |
| laion/CLIP-ViT-g-14-laion2B-s34B-b88K | 97.88 | 78.87 | 91.84 | 45.64 | 85.48 | 96.53 | 74.08 | 99.39 | 91.99 | 84.63 |
| EVA02-CLIP-bigE-14 | 99.47 | 79.74 | 93.36 | 49.19 | 85.84 | 92.60 | 72.82 | 99.73 | 85.98 | 84.30 |
| google/siglip-large-patch16-256 | 96.72 | 80.00 | 89.28 | 45.45 | 84.24 | 96.05 | 75.48 | 99.21 | 92.12 | 84.28 |
| laion/CLIP-ViT-H-14-laion2B-s32B-b79K | 97.64 | 79.26 | 92.36 | 44.29 | 83.89 | 96.17 | 73.02 | 99.39 | 92.11 | 84.24 |
| laion/CLIP-ViT-L-14-laion2B-s32B-b82K | 97.15 | 78.61 | 91.26 | 44.71 | 84.19 | 95.03 | 72.32 | 99.18 | 91.94 | 83.82 |
| royokong/e5-v | 94.14 | 72.24 | 87.51 | 53.96 | 80.02 | 91.60 | 72.39 | 98.81 | 96.11 | 82.98 |
| laion/CLIP-ViT-L-14-DataComp.XL-s13B-b90K | 98.55 | 78.50 | 77.57 | 41.32 | 88.12 | 96.16 | 73.53 | 99.44 | 91.68 | 82.76 |
| google/siglip-base-patch16-512 | 92.66 | 79.48 | 86.40 | 42.92 | 80.76 | 95.48 | 73.52 | 98.72 | 92.63 | 82.51 |
| google/siglip-base-patch16-256 | 93.34 | 78.64 | 87.61 | 42.46 | 79.87 | 95.71 | 73.10 | 98.35 | 92.29 | 82.37 |
| google/siglip-base-patch16-384 | 92.91 | 79.05 | 86.99 | 42.08 | 80.15 | 95.35 | 73.57 | 98.63 | 92.54 | 82.36 |
| google/siglip-base-patch16-256-multilingual | 92.89 | 77.94 | 87.44 | 42.73 | 80.31 | 94.98 | 73.91 | 98.24 | 91.88 | 82.26 |
| google/siglip-base-patch16-224 | 92.60 | 77.94 | 87.75 | 42.19 | 80.07 | 95.42 | 73.07 | 98.33 | 92.02 | 82.15 |
| openai/clip-vit-large-patch14 | 96.15 | 72.78 | 80.54 | 47.11 | 83.59 | 93.70 | 74.74 | 99.39 | 90.93 | 82.10 |
| blip2-finetune-coco | 97.70 | 72.60 | 76.89 | 50.45 | 79.87 | 93.44 | 71.68 | 99.38 | 94.41 | 81.82 |
| laion/CLIP-ViT-B-16-DataComp.XL-s13B-b90K | 96.84 | 76.26 | 88.84 | 35.28 | 83.10 | 95.14 | 70.21 | 98.57 | 90.74 | 81.66 |
| laion/CLIP-ViT-B-32-laion2B-s34B-b79K | 94.05 | 74.00 | 88.93 | 40.98 | 78.19 | 95.00 | 69.68 | 97.77 | 90.28 | 80.99 |
| blip2-pretrain | 98.62 | 74.29 | 78.77 | 52.37 | 68.19 | 92.86 | 73.17 | 98.60 | 90.62 | 80.83 |
| laion/CLIP-ViT-B-32-DataComp.XL-s13B-b90K | 95.86 | 74.19 | 89.07 | 32.75 | 81.41 | 95.55 | 69.92 | 97.67 | 89.63 | 80.67 |
| voyage-multimodal-3 | 95.54 | 72.56 | 79.27 | 46.69 | 75.52 | 94.48 | 67.79 | 98.78 | 78.51 | 78.79 |
| openai/clip-vit-base-patch16 | 91.60 | 69.51 | 74.21 | 45.63 | 72.69 | 91.14 | 70.60 | 98.65 | 90.46 | 78.28 |
| facebook/dinov2-giant | 98.53 | 76.09 | 84.53 | 41.06 | 55.14 | 85.82 | 74.17 | 97.84 | 85.44 | 77.62 |
| facebook/dinov2-base | 96.45 | 75.93 | 81.66 | 39.94 | 53.48 | 86.71 | 72.73 | 97.72 | 85.94 | 76.73 |
| facebook/dinov2-large | 97.95 | 76.45 | 80.23 | 41.40 | 53.22 | 82.40 | 74.37 | 97.93 | 85.51 | 76.61 |
| openai/clip-vit-base-patch32 | 89.85 | 66.61 | 67.23 | 42.88 | 70.12 | 89.47 | 70.96 | 97.73 | 90.07 | 76.10 |
| facebook/dinov2-small | 92.95 | 72.43 | 81.86 | 37.27 | 52.22 | 86.58 | 74.55 | 97.36 | 86.94 | 75.80 |
| Salesforce/blip-itm-large-coco | 95.74 | 70.59 | 80.90 | 48.09 | 64.53 | 83.92 | 67.69 | 98.85 | 69.32 | 75.52 |
| TIGER-Lab/VLM2Vec-Full | 87.93 | 68.51 | 75.80 | 51.41 | 65.06 | 86.75 | 68.21 | 97.60 | 71.00 | 74.70 |
| TIGER-Lab/VLM2Vec-LoRA | 87.94 | 68.48 | 75.74 | 51.37 | 65.09 | 86.71 | 68.16 | 97.60 | 70.97 | 74.67 |
| BAAI/bge-visualized-base | 97.75 | 68.26 | 83.34 | 44.65 | 52.04 | 81.55 | 65.19 | 99.19 | 68.89 | 73.43 |
| Salesforce/blip-itm-base-coco | 87.66 | 69.79 | 80.86 | 43.86 | 62.95 | 85.72 | 64.62 | 97.87 | 66.72 | 73.34 |
| kakaobrain/align-base | 81.23 | 74.04 | 65.29 | 35.89 | 59.44 | 86.78 | 68.30 | 95.95 | 91.81 | 73.19 |
| Salesforce/blip-itm-large-flickr | 94.34 | 69.16 | 79.05 | 45.43 | 58.18 | 83.56 | 66.09 | 98.37 | 64.03 | 73.13 |
| jinaai/jina-clip-v1 | 90.62 | 68.06 | 83.27 | 44.50 | 57.54 | 84.04 | 62.97 | 97.06 | 66.89 | 72.77 |
| nyu-visionx/moco-v3-vit-l | 90.13 | 67.04 | 89.50 | 34.78 | 49.80 | 78.94 | 73.08 | 95.39 | 72.80 | 72.39 |
| BAAI/bge-visualized-m3 | 96.27 | 62.98 | 77.77 | 44.63 | 50.14 | 80.83 | 68.29 | 98.84 | 67.35 | 71.90 |
| EVA02-CLIP-L-14 | 98.99 | 65.09 | 83.89 | 44.24 | 59.34 | 74.80 | 69.04 | 99.39 | 51.74 | 71.83 |
| nyu-visionx/moco-v3-vit-b | 89.36 | 65.95 | 88.65 | 32.70 | 45.93 | 76.74 | 72.99 | 95.14 | 71.02 | 70.94 |
| Salesforce/blip-itm-base-flickr | 83.85 | 66.71 | 78.81 | 41.93 | 56.24 | 83.78 | 63.98 | 96.89 | 60.62 | 70.31 |
| Salesforce/blip-image-captioning-large | 94.58 | 66.27 | 60.57 | 43.60 | 59.85 | 80.69 | 66.72 | 98.23 | 45.21 | 68.41 |
| EVA02-CLIP-B-16 | 98.12 | 61.34 | 77.12 | 43.70 | 38.39 | 76.46 | 65.57 | 99.00 | 45.06 | 67.20 |
| nomic-ai/nomic-embed-vision-v1.5 | 97.25 | 64.16 | 49.01 | 32.04 | 49.03 | 76.17 | 65.69 | 98.59 | 60.33 | 65.81 |
| Salesforce/blip-image-captioning-base | 81.37 | 64.47 | 53.31 | 41.23 | 34.90 | 85.13 | 63.43 | 94.20 | 34.29 | 61.37 |

Table 8.17: **Linear Probe for coarse-grained tasks.**

| model name | Birdsnap | Caltech101 | CIFAR100 | Country211 | FGVCAircraft | Food101 | Imagenet1k | OxfordFlowers | OxfordPets | RESISC45 | StanfordCars | SUN397 | UCF101 | mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| EVA02-CLIP-bigE-14-plus | 80.77 | 97.40 | 94.15 | 31.64 | 78.19 | 94.93 | 82.40 | 99.55 | 94.99 | 92.77 | 95.41 | 80.52 | 93.25 | 85.84 |
| EVA02-CLIP-bigE-14 | 78.22 | 96.40 | 93.76 | 28.84 | 74.37 | 94.70 | 81.47 | 99.47 | 94.31 | 91.96 | 95.02 | 79.93 | 92.12 | 84.66 |
| google/siglip-so400m-patch14-384 | 72.91 | 97.03 | 84.59 | 32.47 | 78.49 | 95.47 | 82.16 | 99.53 | 94.85 | 91.64 | 95.75 | 80.16 | 91.65 | 84.36 |
| google/siglip-large-patch16-384 | 72.02 | 96.83 | 83.49 | 23.00 | 75.44 | 95.00 | 81.20 | 99.57 | 95.09 | 90.94 | 95.51 | 79.03 | 90.45 | 82.89 |
| laion/CLIP-ViT-bigG-14-laion2B-39B-b160k | 74.19 | 96.53 | 88.03 | 28.57 | 70.02 | 92.52 | 78.23 | 99.33 | 93.49 | 90.94 | 95.19 | 78.35 | 90.84 | 82.79 |
| laion/CLIP-ViT-g-14-laion2B-s34B-b88K | 71.86 | 95.36 | 86.62 | 26.10 | 65.99 | 91.13 | 77.32 | 99.18 | 92.71 | 90.74 | 94.95 | 78.19 | 89.10 | 81.48 |
| laion/CLIP-ViT-L-14-DataComp.XL-s13B-b90K | 72.60 | 96.44 | 88.85 | 24.59 | 64.15 | 92.73 | 77.29 | 99.10 | 92.42 | 90.44 | 93.89 | 77.18 | 88.64 | 81.41 |
| laion/CLIP-ViT-H-14-laion2B-s32B-b79K | 72.26 | 95.75 | 86.29 | 25.11 | 65.09 | 91.04 | 76.54 | 98.98 | 91.86 | 90.88 | 94.68 | 78.11 | 89.76 | 81.26 |
| google/siglip-large-patch16-256 | 65.89 | 96.77 | 83.48 | 19.40 | 71.15 | 93.64 | 79.34 | 99.47 | 94.61 | 89.93 | 95.12 | 77.86 | 88.87 | 81.20 |
| google/siglip-base-patch16-512 | 67.46 | 96.93 | 74.47 | 18.05 | 70.08 | 92.67 | 77.54 | 99.10 | 92.92 | 88.36 | 94.57 | 77.16 | 87.73 | 79.77 |
| facebook/dinov2-giant | 81.88 | 89.14 | 89.63 | 13.53 | 70.37 | 88.10 | 78.70 | 99.71 | 94.93 | 86.30 | 83.18 | 72.31 | 89.19 | 79.77 |
| laion/CLIP-ViT-L-14-laion2B-s32B-b82K | 68.42 | 93.79 | 84.73 | 22.02 | 60.45 | 89.61 | 74.67 | 98.82 | 91.70 | 90.14 | 93.90 | 77.06 | 87.49 | 79.45 |
| google/siglip-base-patch16-384 | 66.16 | 97.11 | 74.81 | 17.38 | 69.00 | 92.13 | 76.77 | 99.02 | 92.71 | 88.44 | 94.45 | 76.93 | 86.96 | 79.37 |
| facebook/dinov2-large | 81.05 | 89.58 | 88.83 | 12.73 | 65.14 | 87.79 | 78.62 | 99.61 | 94.90 | 86.21 | 81.00 | 72.76 | 88.08 | 78.95 |
| openai/clip-vit-large-patch14 | 67.59 | 94.32 | 79.99 | 26.97 | 56.19 | 91.99 | 75.10 | 98.92 | 91.94 | 89.64 | 87.43 | 76.08 | 87.78 | 78.76 |
| google/siglip-base-patch16-256 | 60.18 | 96.82 | 75.67 | 14.98 | 66.13 | 90.25 | 74.73 | 99.00 | 91.68 | 87.52 | 93.72 | 75.66 | 85.24 | 77.82 |
| google/siglip-base-patch16-224 | 58.91 | 96.79 | 74.10 | 14.58 | 66.07 | 89.88 | 74.19 | 98.57 | 91.44 | 87.65 | 93.70 | 75.27 | 84.80 | 77.38 |
| google/siglip-base-patch16-256-multilingual | 57.22 | 96.98 | 74.68 | 15.11 | 59.93 | 89.97 | 73.95 | 99.33 | 91.98 | 85.60 | 92.96 | 74.62 | 83.83 | 76.63 |
| facebook/dinov2-base | 77.24 | 89.45 | 84.49 | 10.73 | 62.74 | 84.44 | 75.92 | 99.57 | 94.12 | 81.30 | 78.53 | 71.03 | 85.51 | 76.54 |
| laion/CLIP-ViT-B-16-DataComp.XL-s13B-b90K | 62.55 | 95.36 | 83.40 | 16.98 | 53.93 | 88.07 | 70.75 | 98.92 | 88.79 | 87.64 | 91.22 | 73.72 | 82.81 | 76.47 |
| openai/clip-vit-base-patch16 | 57.29 | 93.47 | 71.24 | 18.69 | 46.01 | 86.46 | 67.53 | 97.27 | 86.02 | 86.51 | 80.58 | 72.13 | 81.94 | 72.70 |
| laion/CLIP-ViT-B-32-DataComp.XL-s13B-b90K | 52.13 | 94.77 | 80.84 | 13.17 | 49.27 | 81.51 | 64.94 | 97.75 | 84.97 | 84.85 | 88.12 | 70.51 | 78.90 | 72.44 |
| facebook/dinov2-small | 71.37 | 88.58 | 77.02 | 8.10 | 58.67 | 77.68 | 69.40 | 99.51 | 92.01 | 76.86 | 69.89 | 66.65 | 80.53 | 72.02 |
| laion/CLIP-ViT-B-32-laion2B-s34B-b79K | 50.76 | 94.89 | 77.17 | 13.41 | 47.12 | 78.78 | 64.13 | 96.84 | 85.05 | 85.85 | 88.49 | 71.65 | 81.08 | 71.94 |
| kakaobrain/align-base | 46.25 | 96.93 | 58.83 | 15.16 | 40.19 | 82.74 | 67.18 | 96.24 | 80.71 | 83.82 | 85.22 | 74.09 | 80.12 | 69.80 |
| royokong/e5-v | 44.62 | 91.83 | 71.90 | 10.64 | 37.74 | 85.31 | 66.57 | 94.16 | 79.91 | 89.10 | 61.55 | 72.64 | 86.07 | 68.62 |
| blip2-finetune-coco | 41.23 | 90.25 | 82.26 | 8.72 | 35.21 | 84.13 | 65.51 | 94.06 | 66.92 | 87.60 | 73.49 | 72.34 | 87.55 | 68.41 |
| BAAI/bge-visualized-base | 45.16 | 90.59 | 82.96 | 10.12 | 35.91 | 84.12 | 64.86 | 95.25 | 78.21 | 83.09 | 66.29 | 73.62 | 77.76 | 68.30 |
| openai/clip-vit-base-patch32 | 47.09 | 91.51 | 67.41 | 14.73 | 38.13 | 79.45 | 61.16 | 94.82 | 80.51 | 82.82 | 73.78 | 69.53 | 78.78 | 67.67 |
| blip2-pretrain | 30.33 | 91.82 | 87.51 | 10.75 | 34.27 | 88.24 | 58.72 | 96.61 | 43.79 | 88.59 | 82.81 | 75.64 | 88.33 | 67.49 |
| Salesforce/blip-itm-large-coco | 34.67 | 89.12 | 76.45 | 9.08 | 22.37 | 81.76 | 67.91 | 96.49 | 81.62 | 85.68 | 74.80 | 72.89 | 84.50 | 67.49 |
| nomic-ai/nomic-embed-vision-v1.5 | 52.16 | 87.50 | 84.12 | 11.73 | 54.01 | 86.86 | 0.10 | 98.88 | 91.88 | 77.94 | 87.91 | 68.93 | 73.47 | 67.35 |
| Salesforce/blip-itm-large-flickr | 36.33 | 87.89 | 73.50 | 10.12 | 21.78 | 81.95 | 66.97 | 96.24 | 81.78 | 83.05 | 74.57 | 71.42 | 82.23 | 66.75 |
| jinaai/jina-clip-v1 | 46.36 | 88.22 | 70.19 | 9.59 | 32.35 | 79.50 | 60.57 | 93.06 | 80.17 | 84.01 | 71.69 | 69.62 | 76.69 | 66.31 |
| voyage-multimodal-3 | 32.69 | 91.39 | 78.13 | 8.90 | 22.52 | 87.38 | 58.51 | 92.65 | 86.56 | 87.34 | 52.91 | 76.31 | 84.33 | 66.12 |
| EVA02-CLIP-L-14 | 0.22 | 89.87 | 87.17 | 18.12 | 54.65 | 90.21 | 0.10 | 98.02 | 90.25 | 84.52 | 89.30 | 68.46 | 81.93 | 65.60 |
| Salesforce/blip-image-captioning-large | 31.82 | 87.77 | 72.95 | 8.44 | 20.62 | 78.36 | 64.42 | 94.20 | 80.46 | 82.01 | 72.30 | 70.18 | 80.57 | 64.93 |
| Salesforce/blip-itm-base-coco | 26.99 | 88.63 | 59.69 | 8.74 | 24.01 | 76.31 | 60.60 | 87.55 | 76.07 | 81.93 | 74.86 | 71.10 | 81.81 | 62.95 |
| BAAI/bge-visualized-m3 | 40.25 | 88.35 | 78.14 | 10.37 | 38.87 | 80.37 | 0.10 | 94.41 | 73.99 | 80.95 | 81.34 | 73.41 | 76.23 | 62.83 |
| Salesforce/blip-itm-base-flickr | 25.92 | 87.43 | 55.88 | 8.13 | 20.10 | 74.50 | 59.25 | 85.88 | 76.89 | 78.51 | 72.76 | 67.47 | 80.14 | 60.99 |
| Salesforce/blip-image-captioning-base | 20.63 | 86.20 | 52.57 | 9.53 | 17.31 | 68.59 | 50.03 | 85.35 | 59.01 | 75.11 | 67.39 | 64.54 | 72.88 | 56.09 |
| nyu-visionx/moco-v3-vit-l | 28.65 | 86.62 | 69.95 | 7.03 | 18.62 | 54.39 | 64.27 | 89.92 | 84.85 | 73.76 | 22.34 | 57.15 | 70.97 | 56.04 |
| nyu-visionx/moco-v3-vit-b | 26.87 | 85.44 | 70.18 | 6.56 | 18.90 | 50.54 | 62.39 | 88.80 | 82.65 | 72.55 | 20.23 | 55.18 | 70.11 | 54.65 |
| TIGER-Lab/VLM2Vec-LoRA | 0.22 | 92.40 | 60.20 | 8.35 | 22.06 | 74.68 | 0.10 | 84.29 | 81.80 | 79.41 | 40.14 | 71.06 | 76.52 | 53.17 |
| TIGER-Lab/VLM2Vec-Full | 0.22 | 92.22 | 60.19 | 8.35 | 22.02 | 74.71 | 0.10 | 84.20 | 81.79 | 79.43 | 40.22 | 71.04 | 76.50 | 53.15 |
| EVA02-CLIP-B-16 | 0.22 | 88.76 | 82.82 | 0.47 | 39.64 | 80.76 | 0.10 | 95.67 | 87.52 | 76.19 | 0.55 | 0.46 | 72.88 | 48.16 |

Table 8.18: **Linear Probe for fine-grained tasks.**

| model name | CIFAR10 ZeroShot | CLEVR ZeroShot | CLEVRCount ZeroShot | DTD ZeroShot | EuroSAT ZeroShot | FER2013 ZeroShot | GTSRB ZeroShot | MNIST ZeroShot | PatchCamelyon ZeroShot | RenderedSST2 | STL10 ZeroShot | mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| google/siglip-so400m-patch14-384 | 96.89 | 22.43 | 40.57 | 66.91 | 58.69 | 54.21 | 64.46 | 88.56 | 54.82 | 70.07 | 98.75 | 65.12 |
| voyage-multimodal-3 | 94.56 | 13.91 | 45.65 | 59.04 | 52.35 | 51.35 | 55.11 | 88.83 | 61.11 | 85.45 | 98.71 | 64.19 |
| EVA02-CLIP-bigE-14-plus | 99.37 | 19.97 | 29.72 | 63.99 | 71.30 | 54.51 | 68.24 | 73.93 | 64.05 | 61.50 | 98.96 | 64.14 |
| laion/CLIP-ViT-bigG-14-laion2B-39B-b160k | 97.94 | 20.05 | 29.71 | 64.20 | 66.56 | 57.13 | 61.15 | 77.09 | 62.45 | 63.37 | 98.23 | 63.44 |
| google/siglip-large-patch16-256 | 96.14 | 21.47 | 40.65 | 64.73 | 53.80 | 56.59 | 61.30 | 85.00 | 52.30 | 61.67 | 99.11 | 62.98 |
| laion/CLIP-ViT-g-14-laion2B-s34B-b88K | 97.81 | 18.59 | 37.43 | 66.17 | 63.09 | 55.66 | 49.78 | 78.74 | 54.67 | 65.57 | 98.83 | 62.39 |
| google/siglip-large-patch16-384 | 95.67 | 20.73 | 32.99 | 63.78 | 55.11 | 58.28 | 63.71 | 85.17 | 52.38 | 56.40 | 99.30 | 62.14 |
| laion/CLIP-ViT-L-14-DataComp.XL-s13B-b90K | 98.10 | 25.19 | 35.57 | 64.47 | 70.70 | 40.97 | 56.92 | 81.44 | 51.99 | 55.63 | 99.18 | 61.83 |
| EVA02-CLIP-bigE-14 | 99.21 | 16.18 | 16.24 | 63.30 | 74.76 | 54.42 | 65.31 | 79.46 | 49.20 | 58.21 | 99.08 | 61.40 |
| laion/CLIP-ViT-H-14-laion2B-s32B-b79K | 97.12 | 16.85 | 26.57 | 62.50 | 72.30 | 50.82 | 57.49 | 78.10 | 51.87 | 62.00 | 98.31 | 61.27 |
| laion/CLIP-ViT-L-14-laion2B-s32B-b82K | 96.92 | 16.09 | 31.31 | 58.88 | 65.35 | 56.27 | 58.27 | 64.12 | 56.02 | 60.57 | 98.75 | 60.23 |
| EVA02-CLIP-L-14 | 99.09 | 20.17 | 31.47 | 62.77 | 67.20 | 49.44 | 56.77 | 62.41 | 50.96 | 61.50 | 99.63 | 60.13 |
| google/siglip-base-patch16-384 | 93.16 | 22.37 | 22.02 | 65.27 | 39.96 | 51.23 | 51.87 | 80.88 | 69.18 | 55.96 | 98.65 | 59.14 |
| google/siglip-base-patch16-512 | 92.96 | 22.21 | 24.09 | 65.53 | 38.37 | 51.41 | 51.17 | 82.85 | 60.91 | 57.22 | 98.55 | 58.66 |
| google/siglip-base-patch16-256 | 93.60 | 23.39 | 22.69 | 65.37 | 44.19 | 53.01 | 50.78 | 84.57 | 50.56 | 57.55 | 98.19 | 58.54 |
| google/siglip-base-patch16-224 | 92.57 | 24.00 | 23.66 | 63.51 | 41.09 | 52.42 | 51.98 | 83.58 | 53.70 | 52.33 | 98.23 | 57.92 |
| royokong/e5-v | 89.66 | 15.80 | 20.72 | 54.52 | 50.48 | 58.44 | 46.17 | 73.55 | 53.84 | 76.06 | 96.58 | 57.80 |
| google/siglip-base-patch16-256-multilingual | 91.23 | 20.36 | 25.40 | 61.06 | 33.33 | 51.18 | 53.06 | 83.09 | 51.94 | 56.84 | 97.63 | 56.83 |
| laion/CLIP-ViT-B-16-DataComp.XL-s13B-b90K | 96.27 | 23.57 | 32.53 | 55.37 | 52.06 | 30.66 | 53.34 | 75.34 | 55.40 | 52.44 | 98.09 | 56.82 |
| openai/clip-vit-large-patch14 | 95.17 | 16.08 | 19.43 | 52.82 | 60.85 | 47.52 | 48.71 | 62.91 | 50.44 | 69.80 | 99.46 | 56.65 |
| laion/CLIP-ViT-B-32-laion2B-s34B-b79K | 93.70 | 18.85 | 15.29 | 54.36 | 48.70 | 47.21 | 45.74 | 63.93 | 60.14 | 56.23 | 96.39 | 54.60 |
| Salesforce/blip-itm-large-coco | 94.78 | 19.08 | 29.03 | 54.63 | 49.20 | 47.35 | 34.82 | 60.68 | 52.68 | 50.08 | 98.33 | 53.70 |
| laion/CLIP-ViT-B-32-DataComp.XL-s13B-b90K | 95.30 | 20.64 | 12.94 | 54.41 | 55.31 | 27.90 | 48.89 | 72.09 | 50.15 | 49.42 | 96.30 | 53.03 |
| blip2-pretrain | 98.11 | 24.71 | 18.26 | 46.28 | 65.69 | 51.10 | 27.05 | 50.49 | 51.13 | 51.78 | 97.99 | 52.96 |
| EVA02-CLIP-B-16 | 98.29 | 15.69 | 21.09 | 50.48 | 58.46 | 48.31 | 42.35 | 42.95 | 50.05 | 53.93 | 99.46 | 52.82 |
| nomic-ai/nomic-embed-vision-v1.5 | 95.25 | 15.81 | 23.39 | 47.77 | 42.37 | 26.99 | 44.26 | 69.64 | 62.84 | 56.51 | 96.16 | 52.82 |
| blip2-finetune-coco | 97.37 | 25.03 | 15.96 | 44.84 | 52.31 | 51.88 | 40.33 | 48.00 | 52.97 | 50.63 | 99.03 | 52.58 |
| jinaai/jina-clip-v1 | 93.39 | 15.62 | 22.35 | 52.87 | 47.37 | 47.21 | 38.76 | 48.58 | 50.73 | 58.98 | 97.81 | 52.15 |
| openai/clip-vit-base-patch16 | 90.13 | 15.83 | 21.21 | 42.87 | 48.59 | 43.55 | 41.05 | 62.33 | 49.00 | 60.52 | 98.38 | 52.13 |
| TIGER-Lab/VLM2Vec-Full | 85.21 | 19.26 | 31.59 | 47.23 | 24.80 | 32.22 | 41.51 | 60.98 | 49.97 | 78.42 | 95.43 | 51.51 |
| TIGER-Lab/VLM2Vec-LoRA | 85.26 | 19.25 | 31.69 | 46.91 | 24.93 | 31.72 | 41.93 | 59.29 | 49.98 | 78.75 | 95.46 | 51.38 |
| openai/clip-vit-base-patch32 | 88.31 | 16.34 | 23.20 | 41.91 | 49.74 | 43.52 | 34.42 | 48.99 | 61.88 | 58.48 | 97.30 | 51.28 |
| Salesforce/blip-itm-large-flickr | 93.88 | 17.32 | 13.94 | 52.98 | 43.50 | 47.99 | 33.25 | 58.51 | 51.54 | 52.83 | 98.21 | 51.27 |
| Salesforce/blip-itm-base-coco | 80.87 | 19.93 | 21.46 | 51.49 | 40.39 | 40.05 | 35.09 | 57.62 | 50.07 | 49.20 | 97.04 | 49.38 |
| BAAI/bge-visualized-base | 97.41 | 15.79 | 16.06 | 42.55 | 51.37 | 35.59 | 32.34 | 45.80 | 50.00 | 54.04 | 98.14 | 49.01 |
| BAAI/bge-visualized-m3 | 94.71 | 21.03 | 14.01 | 33.40 | 41.11 | 37.77 | 31.08 | 57.94 | 51.36 | 57.00 | 96.76 | 48.74 |
| kakaobrain/align-base | 75.59 | 23.52 | 19.47 | 57.71 | 36.87 | 37.81 | 26.84 | 37.74 | 48.45 | 57.50 | 93.73 | 46.84 |
| Salesforce/blip-itm-base-flickr | 77.72 | 12.21 | 14.08 | 47.23 | 34.43 | 29.84 | 32.54 | 54.71 | 50.02 | 51.18 | 97.23 | 45.56 |

Table 8.19: **Zero-shot Classification for coarse-grained tasks.**

| model name | Birdsnap ZeroShot | Caltech101 ZeroShot | CIFAR100 ZeroShot | Country211 ZeroShot | FGVCAircraft ZeroShot | Food101 ZeroShot | Imagenet1k ZeroShot | OxfordPets ZeroShot | RESISC45 ZeroShot | StanfordCars ZeroShot | SUN397 ZeroShot | UCF101 ZeroShot | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| EVA02-CLIP-bigE-14-plus | 79.20 | 84.42 | 91.09 | 34.05 | 54.10 | 94.62 | 79.13 | 95.80 | 72.46 | 94.17 | 74.09 | 69.61 | 76.89 |
| EVA02-CLIP-bigE-14 | 76.72 | 85.04 | 90.81 | 33.73 | 48.06 | 94.64 | 79.93 | 95.69 | 73.51 | 93.98 | 76.04 | 71.12 | 76.61 |
| google/siglip-so400m-patch14-384 | 62.51 | 85.65 | 81.51 | 33.81 | 60.25 | 95.46 | 80.89 | 96.54 | 69.57 | 94.63 | 75.26 | 76.85 | 76.08 |
| laion/CLIP-ViT-bigG-14-laion2B-39B-b160k | 74.39 | 84.27 | 85.35 | 32.39 | 49.56 | 92.78 | 77.65 | 95.28 | 69.76 | 94.02 | 73.54 | 68.62 | 74.80 |
| laion/CLIP-ViT-L-14-DataComp.XL-s13B-b90K | 75.63 | 86.64 | 85.54 | 29.92 | 47.40 | 94.24 | 77.41 | 94.96 | 71.22 | 92.81 | 73.89 | 63.21 | 74.41 |
| google/siglip-large-patch16-384 | 63.53 | 84.39 | 79.23 | 23.05 | 52.84 | 94.95 | 79.93 | 96.78 | 68.65 | 94.19 | 73.22 | 69.76 | 73.38 |
| laion/CLIP-ViT-H-14-laion2B-s32B-b79K | 71.75 | 82.96 | 83.57 | 28.86 | 42.51 | 92.19 | 76.11 | 94.55 | 70.65 | 93.02 | 74.65 | 67.05 | 73.16 |
| laion/CLIP-ViT-g-14-laion2B-s34B-b88K | 69.10 | 83.19 | 77.93 | 29.52 | 44.43 | 91.91 | 76.44 | 93.95 | 71.21 | 93.76 | 73.28 | 70.00 | 72.89 |
| google/siglip-large-patch16-256 | 57.54 | 84.42 | 79.92 | 19.92 | 52.45 | 93.18 | 78.77 | 96.16 | 67.75 | 93.61 | 72.87 | 68.75 | 72.11 |
| EVA02-CLIP-L-14 | 58.40 | 83.09 | 90.07 | 29.23 | 35.67 | 92.91 | 78.06 | 93.95 | 69.30 | 90.03 | 73.98 | 69.18 | 71.99 |
| laion/CLIP-ViT-L-14-laion2B-s32B-b82K | 64.34 | 83.76 | 82.17 | 25.03 | 35.25 | 90.41 | 73.23 | 93.21 | 71.00 | 92.12 | 73.57 | 67.20 | 70.94 |
| google/siglip-base-patch16-512 | 54.02 | 84.11 | 69.72 | 18.03 | 45.72 | 91.89 | 78.13 | 94.96 | 62.97 | 92.51 | 71.29 | 64.68 | 69.00 |
| google/siglip-base-patch16-384 | 53.43 | 84.24 | 70.00 | 17.31 | 45.09 | 91.27 | 77.33 | 95.01 | 62.83 | 92.38 | 70.74 | 64.14 | 68.65 |
| laion/CLIP-ViT-B-16-DataComp.XL-s13B-b90K | 65.91 | 84.80 | 81.21 | 20.52 | 29.85 | 90.08 | 72.23 | 92.80 | 65.08 | 88.30 | 70.57 | 54.78 | 68.01 |
| google/siglip-base-patch16-256 | 48.95 | 83.74 | 71.91 | 15.08 | 44.76 | 89.16 | 75.64 | 94.25 | 61.19 | 91.01 | 70.02 | 61.10 | 67.24 |
| openai/clip-vit-large-patch14 | 53.16 | 81.99 | 75.00 | 29.08 | 32.55 | 92.34 | 71.21 | 93.40 | 67.67 | 76.57 | 64.40 | 66.66 | 67.00 |
| google/siglip-base-patch16-224 | 48.30 | 83.66 | 70.15 | 14.33 | 43.86 | 89.19 | 75.12 | 94.17 | 62.27 | 90.83 | 69.85 | 61.90 | 66.97 |
| EVA02-CLIP-B-16 | 52.08 | 83.83 | 87.01 | 20.45 | 24.81 | 88.36 | 73.19 | 92.26 | 63.11 | 78.82 | 70.32 | 64.66 | 66.58 |
| google/siglip-base-patch16-256-multilingual | 41.06 | 84.55 | 71.01 | 15.20 | 32.34 | 89.48 | 74.36 | 93.79 | 58.48 | 89.13 | 69.46 | 62.87 | 65.14 |
| laion/CLIP-ViT-B-32-DataComp.XL-s13B-b90K | 58.40 | 84.47 | 79.64 | 15.99 | 24.39 | 83.45 | 67.87 | 90.43 | 60.90 | 85.64 | 66.97 | 49.73 | 63.99 |
| laion/CLIP-ViT-B-32-laion2B-s34B-b79K | 51.81 | 82.38 | 74.64 | 15.27 | 23.85 | 81.67 | 65.15 | 90.62 | 63.32 | 85.59 | 68.23 | 53.95 | 63.04 |
| nomic-ai/nomic-embed-vision-v1.5 | 51.59 | 72.37 | 81.79 | 15.17 | 28.08 | 85.99 | 69.82 | 91.09 | 57.62 | 87.44 | 64.42 | 50.51 | 62.99 |
| openai/clip-vit-base-patch16 | 44.03 | 79.04 | 65.97 | 21.29 | 24.90 | 87.67 | 63.99 | 89.04 | 60.54 | 63.54 | 60.56 | 62.19 | 60.23 |
| openai/clip-vit-base-patch32 | 40.57 | 78.55 | 61.65 | 15.96 | 18.87 | 82.71 | 58.84 | 87.49 | 53.40 | 58.61 | 61.06 | 58.15 | 56.32 |
| jinaai/jina-clip-v1 | 32.36 | 80.87 | 71.78 | 12.19 | 11.52 | 76.70 | 58.70 | 80.84 | 55.63 | 68.09 | 65.44 | 50.67 | 55.40 |
| Salesforce/blip-itm-large-flickr | 20.42 | 81.84 | 71.15 | 11.67 | 5.85 | 76.01 | 60.95 | 77.92 | 57.14 | 71.16 | 70.40 | 51.54 | 54.67 |
| kakaobrain/align-base | 25.23 | 80.57 | 51.46 | 16.21 | 11.34 | 80.98 | 62.70 | 84.30 | 48.89 | 72.95 | 70.48 | 45.45 | 54.21 |
| Salesforce/blip-itm-large-coco | 20.10 | 82.82 | 73.83 | 10.49 | 6.99 | 76.05 | 60.94 | 76.64 | 57.03 | 69.71 | 67.14 | 47.49 | 54.10 |
| voyage-multimodal-3 | 13.61 | 78.17 | 71.92 | 11.07 | 11.67 | 74.42 | 60.91 | 70.51 | 61.73 | 49.40 | 68.64 | 61.47 | 52.79 |
| BAAI/bge-visualized-base | 25.72 | 77.70 | 76.85 | 8.63 | 10.53 | 63.73 | 50.35 | 56.23 | 59.24 | 43.10 | 62.01 | 44.84 | 48.24 |
| Salesforce/blip-itm-base-coco | 12.91 | 79.90 | 52.39 | 6.82 | 5.55 | 68.03 | 53.47 | 69.09 | 50.40 | 66.29 | 64.76 | 44.23 | 47.82 |
| blip2-pretrain | 3.30 | 73.06 | 80.75 | 9.25 | 3.18 | 75.22 | 39.04 | 23.68 | 55.57 | 62.89 | 64.46 | 50.51 | 45.08 |
| BAAI/bge-visualized-m3 | 18.64 | 73.13 | 68.25 | 9.18 | 11.88 | 55.13 | 38.48 | 49.74 | 52.06 | 56.35 | 56.30 | 43.85 | 44.42 |
| royokong/e5-v | 7.29 | 80.34 | 60.63 | 7.21 | 7.80 | 60.69 | 48.66 | 32.60 | 57.95 | 26.08 | 64.79 | 60.48 | 42.88 |
| Salesforce/blip-itm-base-flickr | 9.99 | 69.31 | 47.22 | 6.00 | 5.94 | 59.57 | 48.28 | 62.20 | 45.46 | 54.12 | 61.11 | 44.83 | 42.84 |
| TIGER-Lab/VLM2Vec-LoRA | 9.56 | 79.08 | 50.29 | 7.99 | 7.08 | 57.10 | 50.38 | 48.02 | 52.71 | 22.37 | 60.51 | 53.92 | 41.59 |
| TIGER-Lab/VLM2Vec-Full | 9.62 | 79.24 | 50.26 | 7.91 | 6.99 | 57.16 | 50.41 | 48.24 | 52.59 | 22.26 | 60.37 | 53.85 | 41.58 |
| blip2-finetune-coco | 2.70 | 72.65 | 73.58 | 4.52 | 5.13 | 61.85 | 44.20 | 29.93 | 52.87 | 29.71 | 60.32 | 46.69 | 40.35 |

Table 8.20: **Zero-shot Classification for fine-grained tasks.**

| model name | AROCocoOrde | AROFlickrOrder | AROVisualAttribution | AROVisualRelation | SugarCrepe | Winoground | ImageCoDE | Avg. |
|---|---|---|---|---|---|---|---|---|
| royokong/e5-v | 41.30 | 36.12 | 74.54 | 59.20 | 88.02 | 11.75 | 13.21 | 46.30 |
| EVA02-CLIP-bigE-14-plus | 45.05 | 52.82 | 60.52 | 51.40 | 86.50 | 10.75 | 12.69 | 45.67 |
| openai/clip-vit-base-patch16 | 48.18 | 56.12 | 61.84 | 53.64 | 76.90 | 7.25 | 12.16 | 45.16 |
| jinaai/jina-clip-v1 | 52.83 | 49.02 | 61.70 | 51.36 | 81.81 | 6.00 | 13.03 | 45.11 |
| openai/clip-vit-base-patch32 | 46.37 | 56.60 | 61.40 | 51.76 | 76.61 | 9.00 | 13.21 | 44.99 |
| openai/clip-vit-large-patch14 | 45.66 | 54.90 | 61.63 | 53.27 | 76.71 | 8.25 | 12.86 | 44.75 |
| EVA02-CLIP-B-16 | 40.54 | 51.74 | 61.79 | 53.92 | 81.44 | 9.00 | 13.47 | 44.56 |
| EVA02-CLIP-L-14 | 39.41 | 47.00 | 61.98 | 53.31 | 84.38 | 10.25 | 12.64 | 44.14 |
| voyage-multimodal-3 | 40.90 | 37.18 | 65.88 | 51.94 | 89.62 | 6.25 | 12.81 | 43.51 |
| Salesforce/blip-itm-base-flickr | 29.56 | 32.16 | 76.30 | 53.72 | 86.89 | 7.75 | 12.77 | 42.74 |
| laion/CLIP-ViT-bigG-14-laion2B-39B-b160k | 33.20 | 38.84 | 61.11 | 51.80 | 86.47 | 12.25 | 12.81 | 42.36 |
| EVA02-CLIP-bigE-14 | 33.23 | 39.06 | 62.01 | 49.40 | 85.91 | 12.50 | 14.29 | 42.35 |
| laion/CLIP-ViT-H-14-laion2B-s32B-b79K | 33.01 | 39.42 | 62.19 | 50.17 | 85.53 | 10.50 | 13.38 | 42.03 |
| laion/CLIP-ViT-g-14-laion2B-s34B-b88K | 32.92 | 37.44 | 62.89 | 50.90 | 85.22 | 9.00 | 15.20 | 41.94 |
| Salesforce/blip-itm-base-coco | 21.63 | 26.02 | 76.91 | 52.06 | 91.35 | 12.75 | 12.73 | 41.92 |
| laion/CLIP-ViT-L-14-laion2B-s32B-b82K | 31.07 | 38.70 | 58.94 | 50.97 | 84.22 | 8.75 | 13.25 | 40.84 |
| google/siglip-large-patch16-256 | 32.63 | 37.76 | 57.97 | 45.82 | 84.76 | 13.00 | 13.29 | 40.75 |
| laion/CLIP-ViT-B-32-laion2B-s34B-b79K | 33.28 | 40.46 | 58.05 | 50.39 | 81.36 | 7.50 | 13.51 | 40.65 |
| Salesforce/blip-itm-large-coco | 18.27 | 20.56 | 76.52 | 52.28 | 91.07 | 10.50 | 14.07 | 40.47 |
| google/siglip-so400m-patch14-384 | 30.08 | 34.92 | 59.67 | 46.56 | 85.93 | 12.25 | 13.08 | 40.36 |
| kakaobrain/align-base | 25.26 | 36.08 | 66.95 | 51.09 | 81.20 | 8.25 | 13.21 | 40.29 |
| google/siglip-base-patch16-224 | 31.67 | 38.40 | 54.04 | 46.24 | 83.76 | 10.25 | 14.51 | 39.84 |
| laion/CLIP-ViT-B-32-DataComp.XL-s13B-b90K | 33.17 | 38.10 | 57.35 | 49.52 | 79.87 | 8.00 | 12.21 | 39.75 |
| google/siglip-large-patch16-384 | 29.82 | 33.34 | 57.80 | 45.25 | 85.18 | 13.00 | 13.47 | 39.69 |
| google/siglip-base-patch16-256 | 29.97 | 38.62 | 54.10 | 45.84 | 83.56 | 11.25 | 13.34 | 39.52 |
| Salesforce/blip-itm-large-flickr | 16.59 | 15.04 | 75.34 | 53.44 | 88.98 | 13.50 | 12.86 | 39.39 |
| laion/CLIP-ViT-L-14-DataComp.XL-s13B-b90K | 28.70 | 34.56 | 59.18 | 47.73 | 83.86 | 7.50 | 12.25 | 39.11 |
| laion/CLIP-ViT-B-16-DataComp.XL-s13B-b90K | 30.16 | 35.92 | 56.18 | 46.44 | 82.17 | 7.50 | 12.16 | 38.65 |
| google/siglip-base-patch16-384 | 27.34 | 34.18 | 54.38 | 45.99 | 84.00 | 11.00 | 12.90 | 38.54 |
| google/siglip-base-patch16-256-multilingual | 27.60 | 36.00 | 53.85 | 45.19 | 82.80 | 9.00 | 12.51 | 38.14 |
| google/siglip-base-patch16-512 | 22.99 | 32.14 | 54.13 | 46.09 | 84.36 | 9.25 | 13.60 | 37.51 |
| nomic-ai/nomic-embed-vision-v1.5 | 29.59 | 37.20 | 55.10 | 46.36 | 75.04 | 5.75 | 11.95 | 37.28 |
| TIGER-Lab/VLM2Vec-Full | 22.27 | 22.20 | 62.39 | 55.72 | 67.00 | 5.00 | 13.47 | 35.43 |
| TIGER-Lab/VLM2Vec-LoRA | 20.33 | 20.36 | 61.41 | 55.02 | 65.98 | 5.75 | 13.42 | 34.61 |
| blip2-finetune-coco | 6.09 | 6.38 | 68.32 | 51.55 | 84.30 | 8.25 | 13.73 | 34.09 |
| BAAI/bge-visualized-m3 | 22.87 | 8.74 | 58.43 | 45.89 | 75.66 | 2.75 | 11.55 | 32.27 |
| BAAI/bge-visualized-base | 16.44 | 6.16 | 53.89 | 46.56 | 83.60 | 4.75 | 12.12 | 31.93 |
| blip2-pretrain | 4.21 | 5.12 | 67.21 | 49.82 | 72.49 | 6.00 | 11.90 | 30.96 |

Table 8.21: **Compositionality Evaluation Results.**

| ion200kI2T | Fashion200kT2I | Flickr30kI2T | Flickr30kT2I | FORBI2I | InfoSeekIT2IT | InfoSeekIT2T | METI2I | MSCOCOI2T | MSCOCOT2I | NIGHTSI2I | OVENIT2IT | OVENIT2T | EDIST2IT | SketchyI2I | SOPI2I | TUBerlinT2I | WebQAT2IT | WebQAT2T | VisualNewsI2T | VisualNewsT2I | RP2kI2I |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 15.39 | 14.21 | 87.79 | 88.41 | 53.85 | 26.28 | 5.74 | 43.82 | 60.18 | 65.60 | 26.02 | 16.19 | 6.54 | 31.38 | 78.89 | 65.43 | 92.13 | 46.10 | 36.34 | 44.50 | 44.65 | 68.94 |
| 23.81 | 19.73 | 89.94 | 91.31 | 66.90 | 4.38 | 0.24 | 42.69 | 63.53 | 68.94 | 23.63 | 2.54 | 0.57 | 12.41 | 78.57 | 64.30 | 95.34 | 26.17 | 20.04 | 36.50 | 36.46 | 61.95 |
| 15.66 | 13.92 | 88.25 | 88.71 | 56.05 | 0.31 | 0.12 | 44.70 | 61.18 | 66.60 | 26.46 | 0.44 | 0.18 | 18.36 | 85.50 | 69.06 | 94.11 | 47.13 | 39.18 | 46.36 | 45.25 | 71.80 |
| 24.06 | 21.06 | 88.32 | 90.12 | 65.25 | 9.37 | 1.09 | 42.84 | 62.35 | 68.47 | 24.86 | 6.52 | 1.60 | 14.15 | 79.44 | 63.80 | 94.85 | 30.46 | 21.48 | 28.69 | 27.90 | 62.18 |
| 14.11 | 12.57 | 85.55 | 86.77 | 44.91 | 24.53 | 8.74 | 43.35 | 58.83 | 64.26 | 26.45 | 15.33 | 9.88 | 30.23 | 77.12 | 63.39 | 91.85 | 42.41 | 34.34 | 41.01 | 41.44 | 68.30 |
| 14.20 | 12.51 | 86.02 | 86.95 | 48.46 | 23.63 | 5.74 | 43.43 | 59.13 | 64.36 | 26.04 | 14.91 | 7.01 | 28.35 | 74.49 | 63.97 | 91.23 | 42.94 | 33.99 | 40.16 | 40.64 | 67.33 |
| 16.11 | 13.86 | 87.65 | 88.28 | 51.73 | 0.21 | 0.16 | 44.48 | 60.68 | 65.83 | 25.93 | 0.32 | 0.14 | 15.47 | 81.43 | 68.17 | 94.00 | 42.34 | 35.73 | 44.51 | 43.79 | 70.12 |
| 7.32 | 5.47 | 92.17 | 93.75 | 51.59 | 33.95 | 22.65 | 31.97 | 62.16 | 70.36 | 24.86 | 20.45 | 16.40 | 30.75 | 60.23 | 47.18 | 60.25 | 65.64 | 62.62 | 12.47 | 16.91 | 53.52 |
| 22.55 | 19.69 | 86.58 | 88.19 | 59.89 | 10.00 | 1.09 | 41.80 | 60.78 | 67.36 | 23.98 | 7.30 | 1.60 | 14.73 | 77.62 | 62.47 | 95.10 | 32.61 | 23.58 | 26.05 | 25.36 | 62.20 |
| 21.75 | 17.69 | 86.66 | 88.69 | 66.36 | 5.73 | 0.37 | 42.09 | 59.79 | 65.60 | 25.59 | 3.86 | 0.63 | 12.31 | 70.19 | 60.70 | 94.92 | 31.43 | 22.04 | 22.73 | 21.64 | 61.80 |
| 11.12 | 7.64 | 83.05 | 84.20 | 48.05 | 16.89 | 1.52 | 42.69 | 55.56 | 60.69 | 25.68 | 11.78 | 4.13 | 23.46 | 75.64 | 62.47 | 92.89 | 40.71 | 32.44 | 34.67 | 33.90 | 65.52 |
| 12.88 | 10.11 | 84.11 | 86.29 | 45.86 | 18.21 | 2.74 | 41.57 | 56.11 | 61.72 | 25.69 | 12.13 | 5.55 | 26.41 | 70.30 | 61.42 | 92.10 | 39.68 | 31.88 | 35.34 | 35.88 | 65.27 |
| 21.39 | 17.07 | 85.96 | 88.00 | 65.06 | 5.92 | 0.38 | 41.80 | 59.28 | 65.22 | 25.52 | 4.13 | 0.71 | 12.42 | 70.17 | 60.92 | 95.48 | 31.20 | 22.05 | 22.57 | 21.29 | 61.65 |
| 19.23 | 15.70 | 83.26 | 85.63 | 58.63 | 6.33 | 0.48 | 41.16 | 57.29 | 63.62 | 25.66 | 4.43 | 0.77 | 12.56 | 70.87 | 59.73 | 95.24 | 31.50 | 23.54 | 20.02 | 18.81 | 61.44 |
| 19.04 | 15.24 | 82.78 | 85.08 | 55.75 | 5.96 | 0.37 | 40.78 | 57.19 | 63.81 | 25.75 | 4.04 | 0.59 | 12.01 | 71.71 | 59.19 | 95.49 | 30.84 | 22.19 | 19.54 | 18.41 | 60.46 |
| 18.94 | 16.22 | 81.58 | 83.38 | 56.84 | 6.80 | 0.45 | 39.63 | 54.73 | 60.43 | 24.65 | 5.41 | 0.81 | 10.92 | 68.69 | 60.08 | 95.59 | 29.42 | 21.09 | 18.09 | 16.31 | 62.57 |
| 11.43 | 9.30 | 84.62 | 86.78 | 42.88 | 0.05 | 0.05 | 39.48 | 56.08 | 60.74 | 23.06 | 0.12 | 0.08 | 16.51 | 73.13 | 59.24 | 95.21 | 38.03 | 31.20 | 38.69 | 37.38 | 59.18 |
| 8.63 | 5.54 | 78.15 | 79.90 | 42.77 | 9.99 | 0.81 | 41.22 | 50.97 | 56.58 | 25.66 | 8.12 | 2.03 | 21.41 | 64.34 | 59.46 | 91.72 | 35.59 | 29.19 | 26.02 | 24.82 | 65.42 |
| 2.04 | 1.00 | 84.74 | 89.75 | 55.28 | 8.04 | 6.42 | 35.15 | 55.19 | 68.14 | 22.38 | 7.31 | 6.90 | 23.43 | 68.76 | 46.76 | 88.92 | 49.63 | 60.47 | 8.40 | 12.97 | 59.22 |
| 4.14 | 2.63 | 77.86 | 79.31 | 47.95 | 10.29 | 0.14 | 37.34 | 48.84 | 52.66 | 21.89 | 5.71 | 0.20 | 26.05 | 67.56 | 53.66 | 89.24 | 34.30 | 23.98 | 35.24 | 37.39 | 56.53 |
| 9.64 | 7.14 | 77.29 | 79.60 | 38.10 | 10.64 | 0.96 | 39.56 | 50.21 | 56.23 | 25.79 | 7.92 | 1.67 | 24.47 | 59.15 | 56.19 | 91.02 | 37.59 | 31.12 | 28.02 | 26.59 | 63.38 |
| 11.25 | 9.37 | 87.73 | 90.31 | 52.45 | 11.40 | 3.40 | 39.77 | 68.27 | 73.85 | 25.30 | 7.07 | 1.67 | 20.83 | 73.30 | 57.83 | 94.66 | 45.99 | 30.75 | 17.03 | 16.28 | 65.74 |
| 4.78 | 4.18 | 75.65 | 80.94 | 38.10 | 14.22 | 3.46 | 38.77 | 48.28 | 58.37 | 23.87 | 7.96 | 2.30 | 22.46 | 34.56 | 56.59 | 90.81 | 51.61 | 62.41 | 12.77 | 13.95 | 66.40 |
| 8.19 | 5.79 | 72.62 | 75.55 | 31.87 | 7.66 | 0.44 | 39.97 | 47.22 | 53.49 | 25.71 | 6.61 | 1.61 | 19.95 | 58.56 | 56.40 | 91.40 | 37.22 | 29.46 | 20.76 | 19.21 | 64.36 |
| 10.38 | 8.56 | 88.71 | 90.85 | 54.57 | 9.65 | 2.80 | 39.38 | 61.67 | 69.77 | 24.04 | 5.93 | 1.50 | 19.29 | 70.19 | 58.21 | 94.00 | 42.99 | 30.72 | 16.15 | 16.15 | 65.07 |
| 7.92 | 6.60 | 80.31 | 82.61 | 33.27 | 0.00 | 0.00 | 37.53 | 51.77 | 57.29 | 23.22 | 0.16 | 0.06 | 15.29 | 60.75 | 55.39 | 91.35 | 36.21 | 26.90 | 29.28 | 27.55 | 59.99 |
| 10.37 | 7.83 | 82.28 | 85.21 | 38.91 | 0.06 | 0.01 | 36.63 | 53.26 | 58.34 | 22.57 | 0.16 | 0.09 | 10.41 | 60.20 | 51.44 | 78.62 | 35.51 | 26.15 | 24.77 | 25.43 | 51.66 |
| 8.80 | 7.17 | 85.23 | 88.96 | 56.47 | 8.20 | 1.19 | 40.58 | 65.62 | 71.64 | 24.97 | 5.24 | 0.68 | 18.43 | 63.27 | 56.89 | 88.79 | 43.28 | 27.32 | 12.33 | 12.66 | 64.77 |
| 7.88 | 6.38 | 59.80 | 67.62 | 22.85 | 21.98 | 3.02 | 29.98 | 37.11 | 46.67 | 23.09 | 9.83 | 0.98 | 20.62 | 71.63 | 44.96 | 92.01 | 52.67 | 64.90 | 11.59 | 12.50 | 44.32 |
| 2.86 | 1.65 | 74.31 | 76.76 | 40.14 | 5.24 | 0.15 | 35.69 | 45.80 | 50.33 | 21.73 | 3.86 | 0.10 | 25.05 | 42.80 | 50.30 | 86.69 | 27.64 | 21.22 | 28.64 | 29.68 | 54.76 |
| 1.96 | 1.45 | 49.04 | 69.00 | 24.21 | 9.19 | 6.58 | 32.98 | 25.11 | 39.62 | 23.56 | 6.76 | 6.45 | 21.64 | 72.54 | 45.57 | 82.82 | 59.38 | 68.58 | 4.89 | 8.93 | 49.50 |
| 7.31 | 5.50 | 84.67 | 87.62 | 52.05 | 5.26 | 1.02 | 40.39 | 54.58 | 61.69 | 24.28 | 2.49 | 0.47 | 16.18 | 55.60 | 56.85 | 88.96 | 42.00 | 28.53 | 9.65 | 8.94 | 63.88 |
| 3.67 | 3.13 | 49.45 | 74.12 | 21.72 | 14.58 | 11.03 | 34.06 | 31.43 | 50.04 | 22.23 | 9.67 | 11.68 | 23.79 | 71.31 | 48.31 | 88.16 | 59.80 | 68.91 | 3.52 | 8.41 | 53.77 |
| 1.34 | 0.82 | 74.48 | 81.92 | 35.89 | 5.30 | 3.38 | 35.59 | 50.61 | 59.50 | 25.67 | 6.75 | 9.38 | 16.47 | 67.17 | 49.19 | 88.77 | 20.91 | 35.50 | 11.62 | 12.18 | 61.71 |
| 1.33 | 0.75 | 74.47 | 81.84 | 35.86 | 5.39 | 3.43 | 35.40 | 50.57 | 59.48 | 25.22 | 6.74 | 9.35 | 16.36 | 67.05 | 49.12 | 89.13 | 20.80 | 35.56 | 11.43 | 12.00 | 61.68 |
| 2.26 | 1.58 | 71.49 | 73.72 | 29.96 | 3.57 | 0.16 | 35.84 | 44.40 | 47.65 | 23.05 | 3.44 | 0.08 | 24.48 | 39.56 | 48.56 | 81.94 | 23.94 | 21.06 | 24.84 | 25.48 | 55.89 |
| 6.40 | 4.20 | 69.83 | 72.68 | 28.49 | 10.92 | 2.41 | 39.63 | 48.73 | 48.63 | 20.23 | 8.01 | 5.56 | 12.72 | 80.01 | 50.98 | 93.43 | 21.12 | 19.00 | 6.71 | 10.57 | 59.27 |
| 3.19 | 3.31 | 76.53 | 83.71 | 28.85 | 6.07 | 2.41 | 38.51 | 55.70 | 63.20 | 18.70 | 5.80 | 6.51 | 9.22 | 78.11 | 46.08 | 90.63 | 19.14 | 22.39 | 4.13 | 7.66 | 58.17 |

| UB200I2I | StanfordCarsI2I | FashionIQIT2I | GLDv2I2T | HatefulMemesI2T | HatefulMemesT2I | MemotionI2T | MemotionT2I | SciMMIRI2T | SciMMIRT2I | VizWizIT2T | VQA2IT2T | ImageCoDeT2I | BLINKIT2T | BLINKIT2I | ROxfordEasyI2I | ROxfordMediumI2I | ROxfordHardI2I | RParisEasyI2I | RParisMediumI2I | RParisHardI2I | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 81.20 | 90.78 | 5.16 | 80.05 | 66.32 | 67.44 | 90.76 | 93.70 | 29.76 | 29.05 | 9.89 | 2.09 | 11.38 | 39.55 | 29.76 | 30.75 | 17.01 | 10.54 | 11.51 | 2.72 | 4.88 | 41.50 |
| 78.51 | 93.22 | 11.06 | 81.79 | 84.16 | 84.39 | 97.21 | 98.06 | 33.38 | 36.05 | 3.13 | 0.76 | 11.95 | 36.82 | 20.18 | 30.75 | 18.11 | 13.12 | 10.94 | 2.72 | 4.70 | 40.78 |
| 84.36 | 91.06 | 10.88 | 80.73 | 65.29 | 66.54 | 90.26 | 92.73 | 26.70 | 25.30 | 2.59 | 0.60 | 12.12 | 0.75 | 28.12 | 34.96 | 18.64 | 12.02 | 11.54 | 2.75 | 4.94 | 40.12 |
| 78.13 | 93.36 | 7.64 | 67.12 | 83.25 | 84.70 | 96.20 | 97.87 | 29.92 | 31.88 | 4.70 | 1.01 | 11.38 | 33.58 | 19.80 | 34.52 | 18.33 | 12.82 | 10.19 | 2.70 | 4.56 | 39.91 |
| 78.74 | 89.52 | 3.92 | 75.92 | 64.79 | 66.00 | 88.72 | 91.57 | 25.76 | 24.23 | 13.70 | 1.90 | 11.16 | 38.56 | 21.69 | 28.52 | 15.09 | 9.13 | 10.82 | 2.69 | 4.76 | 39.85 |
| 78.81 | 89.68 | 4.19 | 75.03 | 64.23 | 65.31 | 88.76 | 91.38 | 25.70 | 24.30 | 10.54 | 1.69 | 11.55 | 37.81 | 21.06 | 32.93 | 18.69 | 13.08 | 10.86 | 2.73 | 4.86 | 39.73 |
| 84.05 | 90.66 | 9.92 | 78.27 | 65.37 | 66.69 | 90.15 | 91.69 | 24.03 | 23.13 | 2.32 | 0.59 | 12.73 | 0.25 | 21.82 | 32.67 | 17.95 | 14.03 | 11.67 | 2.74 | 4.81 | 39.02 |
| 64.43 | 46.78 | 15.46 | 46.66 | 55.51 | 71.98 | 87.06 | 90.83 | 56.16 | 58.65 | 24.91 | 7.69 | 11.51 | 37.31 | 33.80 | 16.47 | 9.26 | 6.66 | 10.27 | 2.62 | 4.26 | 38.84 |
| 72.23 | 92.87 | 6.75 | 64.81 | 82.53 | 83.83 | 96.05 | 98.02 | 25.34 | 26.45 | 4.52 | 0.94 | 10.82 | 35.57 | 19.42 | 34.44 | 16.84 | 8.95 | 10.01 | 2.72 | 4.55 | 38.83 |
| 74.85 | 91.97 | 6.86 | 58.31 | 83.65 | 83.90 | 96.61 | 97.77 | 27.55 | 28.23 | 3.52 | 0.77 | 10.77 | 38.31 | 13.74 | 31.92 | 17.13 | 10.36 | 10.21 | 2.70 | 4.29 | 38.11 |
| 79.98 | 88.73 | 4.39 | 75.65 | 63.64 | 64.65 | 88.82 | 90.80 | 19.30 | 18.09 | 8.20 | 1.99 | 10.25 | 36.32 | 29.26 | 33.10 | 17.99 | 9.79 | 10.50 | 2.74 | 4.68 | 38.11 |
| 77.29 | 90.01 | 4.13 | 70.26 | 63.62 | 64.78 | 87.35 | 90.42 | 22.55 | 21.00 | 8.20 | 1.67 | 10.60 | 38.56 | 24.09 | 30.79 | 15.92 | 10.14 | 10.77 | 2.71 | 4.83 | 37.96 |
| 73.42 | 92.00 | 6.49 | 58.45 | 83.12 | 83.77 | 96.48 | 97.70 | 25.50 | 26.17 | 3.49 | 0.78 | 11.08 | 35.82 | 12.74 | 29.73 | 16.29 | 9.17 | 9.51 | 2.70 | 4.38 | 37.66 |
| 67.22 | 90.64 | 5.51 | 56.23 | 81.65 | 82.23 | 96.25 | 97.48 | 19.79 | 20.24 | 3.72 | 0.78 | 11.08 | 34.58 | 12.74 | 31.66 | 17.39 | 9.78 | 9.96 | 2.72 | 4.42 | 36.59 |
| 68.61 | 90.67 | 6.12 | 56.02 | 80.72 | 81.98 | 95.96 | 97.32 | 17.92 | 18.36 | 3.47 | 0.78 | 11.47 | 34.08 | 13.87 | 31.53 | 17.09 | 8.53 | 10.00 | 2.71 | 4.34 | 36.27 |
| 61.15 | 89.29 | 5.53 | 57.96 | 79.84 | 80.90 | 94.79 | 96.97 | 15.69 | 15.97 | 3.96 | 0.72 | 10.82 | 37.31 | 18.92 | 28.83 | 15.55 | 7.75 | 9.48 | 2.69 | 4.18 | 35.61 |
| 80.62 | 88.67 | 7.56 | 72.88 | 59.68 | 62.11 | 82.35 | 89.11 | 16.54 | 15.87 | 1.86 | 0.49 | 12.03 | 0.25 | 27.87 | 27.44 | 15.57 | 9.29 | 11.25 | 2.72 | 4.39 | 35.59 |
| 70.78 | 86.07 | 3.44 | 70.21 | 62.19 | 61.87 | 85.03 | 88.16 | 14.65 | 12.88 | 6.96 | 1.94 | 9.90 | 34.08 | 16.90 | 33.57 | 17.09 | 10.05 | 10.63 | 2.71 | 4.92 | 34.84 |
| 42.87 | 42.57 | 5.76 | 28.55 | 62.50 | 69.39 | 80.31 | 92.12 | 30.07 | 36.42 | 16.30 | 6.96 | 12.16 | 31.34 | 36.07 | 13.22 | 7.04 | 3.02 | 9.65 | 2.60 | 2.77 | 34.04 |
| 73.49 | 80.66 | 3.11 | 73.51 | 56.96 | 61.85 | 76.09 | 86.77 | 17.06 | 15.53 | 4.21 | 1.13 | 10.04 | 35.82 | 9.33 | 21.29 | 14.04 | 10.94 | 10.17 | 2.66 | 4.26 | 33.67 |
| 59.77 | 81.18 | 3.05 | 61.27 | 57.54 | 58.53 | 84.13 | 87.27 | 14.62 | 13.25 | 6.02 | 1.80 | 9.82 | 32.84 | 7.44 | 28.26 | 14.18 | 7.38 | 9.59 | 2.69 | 4.57 | 33.27 |
| 54.35 | 66.63 | 6.65 | 33.34 | 35.66 | 46.20 | 64.11 | 73.14 | 7.38 | 7.93 | 6.41 | 1.08 | 12.99 | 42.54 | 24.34 | 20.44 | 11.29 | 7.42 | 9.93 | 2.66 | 3.99 | 32.80 |
| 66.93 | 64.86 | 4.51 | 47.10 | 50.31 | 54.03 | 79.01 | 83.67 | 10.65 | 11.10 | 9.45 | 2.37 | 10.21 | 33.33 | 25.35 | 22.62 | 11.71 | 6.68 | 11.13 | 2.71 | 4.22 | 32.35 |
| 62.29 | 82.13 | 2.83 | 65.62 | 55.31 | 57.24 | 81.17 | 84.08 | 8.84 | 7.35 | 6.04 | 1.67 | 8.25 | 30.35 | 17.91 | 28.99 | 16.27 | 11.04 | 9.53 | 2.70 | 4.64 | 32.18 |
| 53.19 | 66.36 | 6.59 | 37.18 | 36.14 | 42.35 | 63.40 | 69.88 | 7.79 | 8.28 | 4.76 | 0.86 | 12.64 | 35.82 | 22.19 | 17.91 | 9.84 | 5.49 | 10.20 | 2.67 | 3.99 | 31.85 |
| 76.46 | 82.18 | 6.03 | 62.52 | 46.96 | 48.48 | 70.26 | 76.80 | 7.57 | 6.83 | 1.70 | 0.44 | 11.03 | 0.25 | 15.51 | 24.82 | 13.16 | 4.91 | 11.21 | 2.72 | 4.31 | 31.06 |
| 51.81 | 78.04 | 6.73 | 51.22 | 54.68 | 57.25 | 85.09 | 92.18 | 16.14 | 15.01 | 1.86 | 0.57 | 10.29 | 0.00 | 23.33 | 22.52 | 11.04 | 3.13 | 9.05 | 2.65 | 3.95 | 30.98 |
| 42.82 | 70.23 | 5.38 | 25.70 | 35.74 | 46.16 | 60.31 | 69.51 | 5.35 | 6.97 | 5.73 | 0.84 | 12.21 | 40.30 | 22.57 | 22.20 | 11.98 | 7.33 | 11.05 | 2.71 | 3.82 | 30.93 |
| 74.53 | 89.01 | 2.02 | 57.14 | 37.48 | 44.65 | 51.20 | 75.95 | 4.65 | 5.78 | 3.55 | 1.08 | 7.60 | 35.57 | 21.18 | 27.73 | 16.28 | 10.52 | 10.60 | 2.70 | 4.75 | 30.45 |
| 61.86 | 71.66 | 2.59 | 64.55 | 51.57 | 55.78 | 72.61 | 83.20 | 14.41 | 12.78 | 3.91 | 0.94 | 9.86 | 34.58 | 6.81 | 19.19 | 10.57 | 7.28 | 9.53 | 2.72 | 4.30 | 30.13 |
| 63.93 | 82.32 | 5.31 | 22.99 | 37.31 | 50.75 | 58.41 | 79.97 | 4.29 | 7.53 | 7.34 | 1.58 | 6.00 | 32.59 | 19.80 | 24.27 | 11.77 | 6.55 | 9.61 | 2.57 | 3.39 | 28.44 |
| 43.68 | 69.47 | 5.01 | 24.81 | 35.86 | 39.83 | 51.62 | 59.73 | 5.47 | 4.65 | 3.13 | 0.73 | 10.08 | 32.09 | 9.08 | 17.02 | 9.20 | 5.53 | 10.43 | 2.66 | 3.75 | 28.22 |
| 62.79 | 65.04 | 4.52 | 28.24 | 26.17 | 34.61 | 47.27 | 64.35 | 4.03 | 6.84 | 14.53 | 3.35 | 9.30 | 33.83 | 16.02 | 19.88 | 8.53 | 3.79 | 10.43 | 2.68 | 3.61 | 28.05 |
| 43.67 | 43.29 | 2.12 | 18.50 | 20.60 | 43.59 | 20.02 | 59.27 | 24.41 | 21.87 | 24.96 | 17.01 | 10.21 | 32.34 | 39.85 | 15.00 | 9.61 | 5.90 | 9.58 | 2.65 | 2.93 | 27.70 |
| 43.08 | 43.10 | 2.21 | 18.40 | 20.87 | 43.83 | 19.82 | 58.55 | 24.36 | 21.80 | 25.00 | 16.96 | 9.99 | 32.09 | 39.98 | 15.00 | 9.53 | 5.81 | 9.68 | 2.65 | 2.88 | 27.64 |
| 52.05 | 62.43 | 2.21 | 59.09 | 43.22 | 46.49 | 65.15 | 75.61 | 9.71 | 8.14 | 3.81 | 0.82 | 8.90 | 31.59 | 6.68 | 21.55 | 11.21 | 4.54 | 9.22 | 2.71 | 3.73 | 27.49 |
| 39.78 | 68.57 | 1.63 | 31.29 | 42.93 | 44.71 | 56.77 | 56.39 | 3.72 | 2.75 | 10.40 | 0.92 | 10.21 | 37.06 | 19.55 | 12.40 | 7.11 | 3.33 | 9.30 | 2.50 | 3.12 | 26.69 |
| 32.12 | 49.20 | 1.86 | 11.67 | 25.71 | 28.54 | 41.07 | 42.73 | 2.95 | 2.68 | 10.85 | 2.20 | 11.03 | 36.32 | 26.23 | 11.43 | 6.80 | 4.90 | 9.71 | 2.40 | 2.37 | 24.52 |

Table 8.22: **Retrieval Results.**

| Model Name | Clus. (5) | Compo. (7) | Vis. STS (en) (7) | Doc. (10) | Cls. Coarse (8) | Cls. Fine (13) | ZS. Coarse (11) | ZS. Fine (12) | Vision Centric (6) | Retr. (41) | Multiling. Retrieval (3 (55)) | Vis. STS (cross&multi) (2 (19)) | Mean (Eng.) (125) | Mean (Multiling.) (130) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| voyage-multimodal-3 | 82.41 | 43.51 | **81.84** | 71.13 | 78.83 | 66.12 | 64.19 | 52.79 | 48.56 | 38.49 | 58.87 | 70.42 | 62.79 | **63.10** |
| google/siglip-so400m-patch14-384 | 82.11 | 40.36 | 68.02 | 56.38 | 83.94 | 84.36 | 65.12 | 76.08 | 46.25 | 39.01 | 40.19 | 41.39 | **64.16** | 60.27 |
| google/siglip-large-patch16-384 | 79.94 | 39.69 | 69.51 | 53.32 | 83.96 | 82.89 | 62.14 | 73.38 | 45.36 | 38.48 | 51.11 | 39.76 | 62.87 | 59.96 |
| royokong/e5-v | 70.05 | **46.30** | 79.30 | 62.69 | 81.34 | 68.62 | 57.80 | 42.88 | 51.90 | 33.71 | 66.57 | 46.25 | 59.46 | 58.95 |
| google/siglip-large-patch16-256 | 82.13 | 40.75 | 67.43 | 39.37 | 83.30 | 81.20 | 62.98 | 72.11 | 44.87 | 37.45 | 49.84 | 38.11 | 61.16 | 58.29 |
| google/siglip-base-patch16-512 | 74.65 | 37.51 | 67.69 | 52.06 | 81.24 | 79.77 | 58.66 | 69.00 | 53.20 | 37.05 | 43.21 | 38.12 | 61.08 | 57.68 |
| laion/CLIP-ViT-bigG-14-laion2B-39B-b160k | 85.59 | 42.36 | 70.93 | 43.19 | 84.01 | 82.79 | 63.44 | 74.80 | 43.16 | 39.95 | 28.01 | 34.54 | 63.02 | 57.73 |
| google/siglip-base-patch16-384 | 76.27 | 38.54 | 67.05 | 45.00 | 81.09 | 79.37 | 59.14 | 68.65 | 52.80 | 36.55 | 42.55 | 37.54 | 60.45 | 57.05 |
| laion/CLIP-ViT-L-14-DataComp.XL-s13B-b90K | 86.44 | 39.11 | 69.87 | 38.64 | 81.65 | 81.41 | 61.83 | 74.41 | 52.33 | 36.51 | 23.77 | 35.78 | 62.22 | 56.81 |
| EVA02-CLIP-bigE-14-plus | **92.38** | 45.67 | 71.99 | 32.27 | 85.42 | 85.84 | 64.14 | 76.89 | 39.43 | 38.03 | 27.82 | 28.21 | 63.21 | 57.34 |
| google/siglip-base-patch16-256-multilingual | 74.56 | 38.14 | 65.46 | 26.35 | 81.06 | 76.63 | 56.83 | 65.14 | 51.25 | 34.40 | 59.21 | 40.27 | 56.98 | 55.77 |
| laion/CLIP-ViT-H-14-laion2B-s32B-b79K | 83.86 | 42.03 | 65.50 | 40.41 | 83.25 | 81.26 | 61.27 | 73.16 | 45.80 | 38.30 | 25.54 | 33.85 | 61.48 | 56.19 |
| laion/CLIP-ViT-g-14-laion2B-s34B-b88K | 82.74 | 41.94 | 69.14 | 37.63 | 83.71 | 81.48 | 62.39 | 72.89 | 44.16 | 38.36 | 25.92 | 31.70 | 61.45 | 56.01 |
| EVA02-CLIP-bigE-14 | 89.42 | 42.35 | 68.80 | 31.62 | 84.10 | 84.66 | 61.40 | 76.61 | 43.57 | 37.03 | 25.54 | 28.29 | 61.95 | 56.11 |
| google/siglip-base-patch16-256 | 75.24 | 39.52 | 66.16 | 31.66 | 81.14 | 77.82 | 58.54 | 67.24 | 52.18 | 35.47 | 41.26 | 34.44 | 58.49 | 55.05 |
| google/siglip-base-patch16-224 | 74.50 | 39.84 | 64.25 | 26.16 | 80.92 | 77.38 | 57.92 | 66.97 | 51.06 | 35.10 | 41.23 | 33.54 | 57.41 | 54.07 |
| laion/CLIP-ViT-L-14-laion2B-s32B-b82K | 83.50 | 40.84 | 65.82 | 36.26 | 82.80 | 79.45 | 60.23 | 70.94 | 45.85 | 36.64 | 23.02 | 26.02 | 60.23 | 54.28 |
| openai/clip-vit-large-patch14 | 76.41 | 44.75 | 64.45 | 37.97 | 81.00 | 78.76 | 56.65 | 67.00 | 44.10 | 32.13 | 20.24 | 35.12 | 58.32 | 53.22 |
| laion/CLIP-ViT-B-16-DataComp.XL-s13B-b90K | 81.73 | 38.65 | 68.47 | 27.02 | 80.53 | 76.47 | 56.82 | 68.01 | 54.34 | 33.42 | 21.57 | 28.49 | 58.55 | 52.96 |
| TIGER-Lab/VLM2Vec-LoRA | 72.64 | 34.61 | 72.59 | 49.71 | 75.14 | 53.17 | 51.38 | 41.59 | 62.02 | 27.06 | 34.92 | 42.21 | 53.99 | 51.42 |
| TIGER-Lab/VLM2Vec-Full | 70.72 | 35.43 | 72.64 | 49.80 | 75.16 | 53.15 | 51.51 | 41.58 | 62.12 | 27.00 | 34.96 | 42.19 | 53.91 | 51.35 |
| EVA02-CLIP-L-14 | 88.27 | 44.14 | 63.04 | 22.08 | 74.35 | 65.60 | 60.13 | 71.99 | 39.37 | 33.87 | 23.43 | 22.48 | 56.28 | 50.73 |
| openai/clip-vit-base-patch16 | 69.47 | 45.16 | 66.06 | 25.50 | 76.75 | 72.70 | 52.13 | 60.23 | 46.92 | 29.16 | 17.66 | 29.80 | 54.41 | 49.29 |
| Salesforce/blip-itm-large-coco | 77.53 | 40.47 | 62.88 | 17.70 | 76.29 | 67.49 | 53.70 | 54.10 | 51.10 | 31.82 | 18.53 | 33.69 | 53.31 | 48.77 |
| laion/CLIP-ViT-B-32-laion2B-s34B-b79K | 77.99 | 40.65 | 59.53 | 16.86 | 79.82 | 71.94 | 54.60 | 63.04 | 42.97 | 32.14 | 20.13 | 26.16 | 53.95 | 48.82 |
| jinaai/jina-clip-v1 | 69.95 | 45.11 | 62.62 | 17.64 | 73.51 | 66.31 | 52.15 | 55.40 | 45.38 | 32.02 | 18.09 | 34.59 | 52.01 | 47.73 |
| Salesforce/blip-itm-base-coco | 70.59 | 41.92 | 66.64 | 18.01 | 74.17 | 62.95 | 49.38 | 47.82 | 47.26 | 30.26 | 16.81 | 38.59 | 50.90 | 47.03 |
| laion/CLIP-ViT-B-32-DataComp.XL-s13B-b90K | 77.37 | 39.75 | 56.52 | 12.43 | 79.55 | 72.44 | 53.03 | 63.99 | 46.05 | 30.85 | 20.13 | 23.63 | 53.20 | 47.98 |
| Salesforce/blip-itm-large-flickr | 76.43 | 39.39 | 60.16 | 18.47 | 74.27 | 66.75 | 51.27 | 54.67 | 47.01 | 30.76 | 18.12 | 30.50 | 51.92 | 47.32 |
| kakaobrain/align-base | 59.59 | 40.29 | 62.74 | 31.44 | 70.87 | 69.80 | 46.84 | 54.21 | 45.69 | 29.87 | 22.36 | 27.29 | 51.13 | 46.75 |
| Salesforce/blip-itm-base-flickr | 67.27 | 42.74 | 64.25 | 14.96 | 71.52 | 60.99 | 45.56 | 42.84 | 52.35 | 27.50 | 13.44 | 38.03 | 49.00 | 45.12 |
| BAAI/bge-visualized-m3 | 73.57 | 32.27 | 64.16 | 12.38 | 72.47 | 62.83 | 48.74 | 44.42 | 43.85 | 27.64 | 46.35 | 23.46 | 48.23 | 46.01 |
| BAAI/bge-visualized-base | 76.19 | 31.93 | 61.58 | 10.34 | 73.99 | 68.30 | 49.01 | 48.24 | 52.43 | 27.07 | 12.25 | 24.75 | 49.91 | 44.67 |
| openai/clip-vit-base-patch32 | 67.90 | 44.99 | 54.39 | 13.23 | 74.36 | 67.67 | 51.28 | 56.32 | 42.73 | 26.53 | 16.73 | 21.80 | 49.94 | 44.83 |
| EVA02-CLIP-B-16 | 82.55 | 44.56 | 43.11 | 9.42 | 69.96 | 48.16 | 52.82 | 66.58 | 45.34 | 29.59 | 20.12 | 22.4 | 49.21 | 44.55 |
| blip2-pretrain | 74.01 | 30.96 | 42.75 | 12.30 | 79.61 | 67.49 | 52.96 | 45.08 | 53.14 | 25.30 | 13.86 | 21.77 | 48.36 | 43.27 |
| blip2-finetune-coco | 67.84 | 34.09 | 45.72 | 15.59 | 80.25 | 68.41 | 52.58 | 40.35 | 52.72 | 23.55 | 13.05 | 22.87 | 48.11 | 43.08 |
| nomic-ai/nomic-embed-vision-v1.5 | 83.64 | 37.28 | 29.29 | 11.92 | 66.49 | 67.35 | 52.82 | 62.99 | 46.72 | 29.13 | 14.48 | 14.10 | 48.76 | 43.02 |

Table 8.23: **MIEB overall per-task category results, grouped by categories assessed.** We provide averages of both English-only tasks and tasks of all languages, and the table is ranked by average on all tasks, including multilingual ones.

| Model Name | Type | Model Size | Modalities |
|---|---|---|---|
| kakaobrain/align-base (45) | Encoder | 176 | image, text |
| blip2-pretrain (192) | Encoder | 1173 | image, text |
| blip2-finetune-coco (192) | Encoder | 1173 | image, text |
| Salesforce/blip-vqa-base (191) | Encoder | 247 | image, text |
| Salesforce/blip-vqa-capfilt-large (191) | Encoder | 247 | image, text |
| Salesforce/blip-itm-base-coco (191) | Encoder | 247 | image, text |
| Salesforce/blip-itm-large-coco (191) | Encoder | 470 | image, text |
| Salesforce/blip-itm-base-flickr (191) | Encoder | 247 | image, text |
| Salesforce/blip-itm-large-flickr (191) | Encoder | 470 | image, text |
| openai/clip-vit-large-patch14 (23) | Encoder | 428 | image, text |
| openai/clip-vit-base-patch32 (23) | Encoder | 151 | image, text |
| openai/clip-vit-base-patch16 (23) | Encoder | 151 | image, text |
| facebook/dinov2-small (42) | Encoder | 22 | image |
| facebook/dinov2-base (42) | Encoder | 86 | image |
| facebook/dinov2-large (42) | Encoder | 304 | image |
| facebook/dinov2-giant (42) | Encoder | 1140 | image |
| royokong/e5-v (7) | MLLM | 8360 | image, text |
| QuanSun/EVA02-CLIP-B-16 (175) | Encoder | 149 | image, text |
| QuanSun/EVA02-CLIP-L-14 (175) | Encoder | 428 | image, text |
| QuanSun/EVA02-CLIP-bigE-14 (175) | Encoder | 4700 | image, text |
| QuanSun/EVA02-CLIP-bigE-14-plus (175) | Encoder | 5000 | image, text |
| jinaai/jina-clip-v1 (173) | Encoder | 223 | image, text |
| nyu-visionx/moco-v3-vit-b (168) | Encoder | 86 | image |
| nyu-visionx/moco-v3-vit-l (168) | Encoder | 304 | image |
| nomic-ai/nomic-embed-vision-v1.5 (249; 250) | Encoder | 92 | image, text |
| laion/CLIP-ViT-L-14-DataComp.XL-s13B-b90K (174) | Encoder | 428 | image, text |
| laion/CLIP-ViT-B-32-DataComp.XL-s13B-b90K (174) | Encoder | 151 | image, text |
| laion/CLIP-ViT-B-16-DataComp.XL-s13B-b90K (174) | Encoder | 150 | image, text |
| laion/CLIP-ViT-bigG-14-laion2B-39B-b160k (43) | Encoder | 2540 | image, text |
| laion/CLIP-ViT-g-14-laion2B-s34B-b88K (43) | Encoder | 1367 | image, text |
| laion/CLIP-ViT-H-14-laion2B-s32B-b79K (43) | Encoder | 986 | image, text |
| laion/CLIP-ViT-L-14-laion2B-s32B-b82K (43) | Encoder | 428 | image, text |
| laion/CLIP-ViT-B-32-laion2B-s34B-b79K (43) | Encoder | 151 | image, text |
| Alibaba-NLP/gme-Qwen2-VL-2B-Instruct (201) | Encoder | 2210 | image, text |
| Alibaba-NLP/gme-Qwen2-VL-7B-Instruct (201) | Encoder | 8290 | image, text |
| google/siglip-so400m-patch14-224 (44) | Encoder | 877 | image, text |
| google/siglip-so400m-patch14-384 (44) | Encoder | 878 | image, text |
| google/siglip-so400m-patch16-256-i18n (44) | Encoder | 1130 | image, text |
| google/siglip-base-patch16-256-multilingual (44) | Encoder | 371 | image, text |
| google/siglip-base-patch16-256 (44) | Encoder | 203 | image, text |
| google/siglip-base-patch16-512 (44) | Encoder | 204 | image, text |
| google/siglip-base-patch16-384 (44) | Encoder | 203 | image, text |
| google/siglip-base-patch16-224 (44) | Encoder | 203 | image, text |
| google/siglip-large-patch16-256 (44) | Encoder | 652 | image, text |
| google/siglip-large-patch16-384 (44) | Encoder | 652 | image, text |
| BAAI/bge-visualized-base (176) | Encoder | 196 | image, text |
| BAAI/bge-visualized-m3 (176) | Encoder | 873 | image, text |
| TIGER-Lab/VLM2Vec-LoRA (178) | MLLM | 4150 | image, text |
| TIGER-Lab/VLM2Vec-Full (178) | MLLM | 4150 | image, text |
| voyageai/voyage-multimodal-3 (179) | MLLM | N/A | image, text |

Table 8.24: **List of all models evaluated in MIEB.** Model sizes are in millions of parameters.

# CHAPTER 9

# Conclusions and Future Work

This thesis investigates contrastive learning for sentence representations, with a focus on three main areas: retrieval, reasoning, and perception. In this thesis, we have analyzed the properties of contrastive learning, applies these findings to information retrieval, proposes new benchmarks to evaluate reasoning and multimodal capabilities, and explores new methods for text representation.

We summarize the main contribution of this thesis as follows: First, we provide a framework to analyze properties of embedding models trained using contrastive learning, which is shown able to provide transferable insights by applying the analysis principles to various research questions in later chapters. Second, we provide methods to train state-of-the-art dense retrieval text models and visual text representation models. Third, we construct two comprehensive benchmarks to evaluate respectively reasoning capabilities and multimodal capabilities of embedding models. Last, we contribute the conceptual principle that training representation models is aligning their representation capabilities with their generative capabilities, and ground this principles in results throughout the thesis. Below we provide overview of each chapter in detail.

We first examined why contrastive learning is effective for sentence embeddings

(Chapter 3). We found that this training method improves the isotropy of the embedding space, which helps address the representation degeneration problem. A key finding was that contrastive learning produces very high intra-sentence similarity, meaning tokens in the same sentence are mapped to similar vector representations. This occurs because functional tokens learn to follow the representations of more semantically important tokens, helping to form a single, coherent representation for the sentence.

Based on these findings, we addressed a specific problem in Information Retrieval (IR) (Chapter 4). We found that standard contrastive models are vulnerable to changes in document length, a phenomenon we termed "length attack". This happens because model isotropy is often limited to the document lengths seen during training, and intra-document similarity incorrectly increases with length. To address this, we proposed LA(SER)$^{\mathbf{3}}$, an unsupervised method that uses document elongation as a training signal. This approach made models more robust to length variations and achieved state-of-the-art performance on unsupervised IR benchmarks.

The rise of Retrieval-Augmented Generation (RAG) with Large Language Models (LLMs) requires embedding models to perform more complex tasks than semantic matching. In Chapter 5, we argued that existing STS and IR benchmarks are insufficient to evaluate these new requirements. We therefore introduced a new evaluation framework, Reasoning as Retrieval (RAR-b), to directly test the reasoning and instruction-following capabilities of embedding models. Our experiments with RAR-b showed that many current retrievers perform poorly on these tasks, but that newer decoder-based models show significant promise in this area. More importantly, we conceptualize "Training representation models is aligning their representational capabilities with their generative capabilities". In the later two sections, we seek to then explore whether the upperbound of this alignment can be raised by grounding in more modalities.

The first work we did was pixel sentence representation learning. In Chapter 6, we explored a new approach: learning sentence representations directly from pixels, as a visual perception task. This tokenization-free method allows for data augmentations like typos and word shuffling, which are difficult to apply to standard language models.

The resulting model achieved performance comparable to token-based models on several benchmarks. We also observed a "leapfrogging" effect during cross-lingual training, where training on multiple languages improved the model's performance on all of them, including the original source language.

Finally, the research expanded to multimodal representations in Chapter 7. We identified that existing evaluation methods for image and image-text models are inconsistent and task-specific. To solve this, we created the Massive Image Embedding Benchmark (MIEB), a large-scale, unified benchmark covering 130 tasks and 38 languages. Our evaluation of 50 models on MIEB showed that no single model performs best across all task categories. We also found a high correlation between a vision encoder's performance on MIEB and the performance of an MLLM that uses it, making the benchmark a useful tool for model selection. More importantly, we see how representations in MLLM-based models can be activated by contrastive learning of much smaller scale compared to CLIP-style models, again validating our conceptualization of the relationship between generative capabilities and representational capabilities (251; 252; 253).

In summary, this thesis provides an analysis of contrastive learning, develops new methods for robust retrieval, introduces new benchmarks for reasoning and multimodal evaluation, and explores unified cross-modal approach to model text representation. The work contributes new tools and insights towards building more general-purpose representation models. Through progression from analysis, to robust training method, to reasoning benchmark, to pixel text and multimodal benchmark, this thesis reveals a conceptual principle: Training representation models is the alignment of representation to models' inherently strong generative capabilities, which provides a high-level pathway to train representation models: Inject fundamental knowledge through generative pretraining, then activate latent representations into a similarity-matching space.

My ongoing and future research agenda builds directly on the core principle I have developed and validated during above work: that *representation learning is the alignment of representational capabilities with pre-existing generative knowledge*. I outline two important directions below for extending this vision—first by

rethinking how we activate representations, and second by interrogating what kinds of generative processes give rise to the strongest representational potential. By investigating Reinforcement Learning for Representation Learning (RL for RL), we explore the possibility of the de facto usage of contrastive learning for post-hoc representation activation into RL-based. By assessing non-autoregressive backbones, we rethink whether the generation-representation relationship revealed in this thesis is architecture-specific or universally applies to models that use generative objectives.

**Reinforcement Learning for Representation Learning ("RL for RL")** Given the success of reinforcement learning (RL) in enhancing reasoning in LLMs (189; 190), I am exploring whether RL can serve as a more efficient and capability-preserving alternative to contrastive losses (e.g., InfoNCE) for activating representations.

Co-authors and I are actively exploring fundamental reinforcement learning techniques that are not only suitable for enhancing general reasoning capabilities but also applicable to activating representation capabilities built in pretrained language models. We developed VL-Cogito (254), a multimodal foundation model trained via progressive curriculum RL to perform complex reasoning. We are now adapting this framework to train reasoning-aware embedding models, where the model must reason about relevance before producing a representation. This includes RL-based training for embedding models, re-rankers and retrieval-specific query rewriters—moving beyond passive encoding toward active, goal-directed representation. Such systems would enable *Environment-aware AI Search*: retriever-specific, database-specific adaptation, facilitating use cases like corporate search, going beyond current deep research paradigm which predominantly overfits to web search.

**Representation Learning with Non-Autoregressive Backbones** While my work establishes that stronger generative capabilities lead to better representations, a critical question remains: Is this scaling law driven by the generative objective itself, or by architectural priors (e.g., decoder-only, autoregressive design)?

Interestingly, recent work shows that equipping decoder models with bidirectional attention during contrastive tuning can outperform their unidirectionally pretrained counterparts—suggesting architecture may not be destiny.

This motivates exploration of diffusion-based language models (255; 256; 257), which generate non-autoregressively, enabling self-correction and global context integration. Despite being largely unexplored for representation learning, diffusion models offer a compelling testbed: they are powerful generative systems trained without next-token prediction. If they too exhibit strong post-contrastive representational performance, it would suggest that generative capacity, not autoregressive training, is the key driver.

I believe this is a promising frontier for building more coherent, globally aware embedding models—an exciting direction I plan to pioneer.

# CHAPTER 10

# List of Publications

**[NeurIPS 2025]** Xiao, C., Chan, HP., Zhang, H., Xu, W., Aljunied, M., Rong, Y., 2025. Scaling Language-centric Omni-modal Representation Learning Advances in Neural Information Processing Systems.

**[EMNLP 2025]** Wang, Y., Xiao, C., Hsiao, C.Y., Chang, Z.Y., Chen, C.L., Loakman, T. and Lin, C., 2025. Drivel-ology: Challenging LLMs with Interpreting Nonsense with Depth. In Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics.

**[EMNLP 2025]** Sun, Y., Qian, X., Xu, W., Zhang, H., Xiao, C., Li, L., Rong, Y., Huang, W., Bai, Q. and Xu, T., 2025. ReasonMed: A 370K Multi-Agent Generated Dataset for Advancing Medical Reasoning. In Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics.

**[ICCV 2025]** Xiao, C., Chung, I., Kerboua, I., Stirling, J., Zhang, X., Kardos, M., Solomatin, R., Al Moubayed, N., Enevoldsen, K., Muennighoff, N., 2025. MIEB: Massive Image Embedding Benchmark. In Proceedings of the IEEE/CVF international conference on computer vision

**[Transactions of the Association for Computational Linguistics, TACL]**

Wang, Y., Xiao, C., Li, Y., Middleton, S.E., Al Moubayed, N. and Lin, C., 2025. Adversarial defence without adversarial defence: instance-level principal component removal for robust language models. Transactions of the Association for Computational Linguistics.

**[Transactions of the Association for Computational Linguistics, TACL; co-first*]** Hong, H. *, Xiao, C.*, Wang, Y., Liu, Y., Rong, W. and Lin, C., 2025. Beyond One-Size-Fits-All: Inversion Learning for Highly Effective NLG Evaluation Prompts. Transactions of the Association for Computational Linguistics.

**[ICCV 2025]** Hudson, G.T., Slack, D., Winterbottom, T., Sterling, J., Xiao, C., Shentu, J. and Moubayed, N.A., 2024. Everything is a video: Unifying modalities through next-frame prediction. In Proceedings of the IEEE/CVF international conference on computer vision

**[ACL 2025]** Xiao, C., Chan, HP., Zhang, H., Aljunied, M., Bing, L., Al Moubayed, N., Rong, Y., 2025. Analyzing LLMs' Knowledge Boundary Cognition across Languages through the Lens of Internal Representations. In Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 24099–24115, Vienna, Austria. Association for Computational Linguistics.

**[ICLR 2025; core managing authors*]** Enevoldsen, K.*, Chung, I.*, Kerboua, I.*, Kardos, M.*, . . . , Xiao, C.*, Adlakha, V. *, Weller, O.*, Reddy. S.*, Muennighoff, N.*, 2025. MMTEB: Massive Multilingual Text Embedding Benchmark, in The Thirteenth International Conference on Learning Representations.

**[NAACL 2025]** Ma, Y., Xiao, C., Yuan, C., Van Der Veer, S.N., Hassan, L., Lin, C. and Nenadic, G., 2025. CAST: Corpus-Aware Self-similarity Enhanced Topic modelling. In Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers) (pp. 7548-7561).

**[EMNLP 2024 findings; AC best paper nomination; co-first*]** James, J.*, Xiao, C*., Li, Y. and Lin, C., 2024. On the Rigour of Scientific Writing: Criteria, Analysis, and Insights. In Findings of the Association for Computational Linguistics: EMNLP 2024 (pp. 6523-6538).

**[ICLR 2024]** Yizhi, L.I., Yuan, R., Zhang, G., Ma, Y., Chen, X., Yin, H., Xiao, C., Lin, C., Ragni, A., Benetos, E. and Gyenge, N., 2023. MERT: Acoustic music understanding model with large-scale self-supervised training. In The Twelfth International Conference on Learning Representations.

**[ACL Findings 2024]** Wu, S., Li, Y., Zhu, K., Zhang, G., Liang, Y., Ma, K., Xiao, C., Zhang, R.H., Yang, B., Chen, W. and Huang, W., 2024, August. SciMMIR: Benchmarking Scientific Multi-modal Information Retrieval. In Findings of the Association for Computational Linguistics ACL 2024 (pp. 12560-12574).

**[MTEB integrated]** Xiao, C., Hudson, G., Al Moubayed, N., 2024. RAR-b: Reasoning as Retrieval Benchmark. [MIEB integrated] Xiao, C., Huang, Z., Chen, D., Hudson, G., Li, Y., Duan, H., Lin, C., Fu, J., Han, J., Al Moubayed, N., Pixel Sentence Representation Learning

**[EMNLP 2023]** Xiao, C., Li, Y., Hudson, G., Lin, C. and Al Moubayed, N., 2023. Length is a Curse and a Blessing for Document-level Semantics. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 1385–1396, Singapore. Association for Computational Linguistics.

**[EMNLP/CoNLL 2023 BabyLM Challenge; Loose Track Winner]** Xiao, C., Hudson, G., Al Moubayed, N., 2023. Towards more Human-like Language Models based on Contextualizer Pretraining Strategy. In Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning, pages 317–326, Singapore. Association for Computational Linguistics.

**[LREC-Coling 2024]** Yang, B., Tang, C., Zhao, K., Xiao, C. and Lin, C., 2024, May. Effective Distillation of Table-based Reasoning Ability from LLMs. In Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024) (pp. 5538-5550).

**[ICLR 2023 workshops]** Xiao, C., Ye, Z., Hudson, G.T., Sun, Z., Blunsom, P. and Al Moubayed, N., 2023. Can Text Encoders be Deceived by Length Attack?. in The Eleventh International Conference on Learning Representations, Tiny Papers Track.

**[ACL Findings 2023]** Xiao, C., Long, Y. and Al Moubayed, N., 2023, July. On Isotropy, Contextualization and Learning Dynamics of Contrastive-based Sentence Representation Learning. In Findings of the Association for Computational Linguis-

tics: ACL 2023 (pp. 12266-12283).

# Bibliography


[1] S. Tong, E. Brown, P. Wu, S. Woo, M. Middepogu, S. C. Akula, J. Yang, S. Yang, A. Iyer, X. Pan, *et al.*, "Cambrian-1: A fully open, vision-centric exploration of multimodal llms," *arXiv preprint arXiv:2406.16860*, 2024.

[2] X. Wu, C. Gao, Z. Lin, J. Han, Z. Wang, and S. Hu, "InfoCSE: Information-aggregated contrastive learning of sentence embeddings," in *Findings of the Association for Computational Linguistics: EMNLP 2022*, (Abu Dhabi, United Arab Emirates), pp. 3060–3070, Association for Computational Linguistics, Dec. 2022.

[3] T. Gao, X. Yao, and D. Chen, "Simcse: Simple contrastive learning of sentence embeddings," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6894–6910, 2021.

[4] Y. Yu, C. Xiong, S. Sun, C. Zhang, and A. Overwijk, "Coco-dr: Combating distribution shifts in zero-shot dense retrieval with contrastive and distributionally robust learning," 2022.

[5] J. Su, J. Cao, W. Liu, and Y. Ou, "Whitening sentence representations for better semantics and faster retrieval," *arXiv preprint arXiv:2103.15316*, 2021.

[6] B. Li, H. Zhou, J. He, M. Wang, Y. Yang, and L. Li, "On the sentence embeddings from pre-trained language models," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9119–9130, 2020.

[7] T. Jiang, M. Song, Z. Zhang, H. Huang, W. Deng, F. Sun, Q. Zhang, D. Wang, and F. Zhuang, "E5-v: Universal embeddings with multimodal large language models," *arXiv preprint arXiv:2407.12580*, 2024.

[8] E. Agirre, J. Bos, M. Diab, S. Manandhar, Y. Marton, and D. Yuret, "* sem 2012: The first joint conference on lexical and computational semantics–volume 1: Proceedings of the main conference and the shared task, and volume 2: Proceedings of the sixth international workshop on semantic evaluation

(semeval 2012)," in ** SEM 2012: The First Joint Conference on Lexical and Computational Semantics–Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, 2012.

[9] E. Agirre, D. Cer, M. Diab, A. Gonzalez-Agirre, and W. Guo, "*SEM 2013 shared task: Semantic textual similarity," in *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity* (M. Diab, T. Baldwin, and M. Baroni, eds.), (Atlanta, Georgia, USA), pp. 32–43, Association for Computational Linguistics, June 2013.

[10] E. Agirre, C. Banea, C. Cardie, D. Cer, M. Diab, A. Gonzalez-Agirre, W. Guo, R. Mihalcea, G. Rigau, and J. Wiebe, "Semeval-2014 task 10: Multilingual semantic textual similarity," in *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, pp. 81–91, 2014.

[11] E. Agirre, C. Banea, C. Cardie, D. Cer, M. Diab, A. Gonzalez-Agirre, W. Guo, I. Lopez-Gazpio, M. Maritxalar, R. Mihalcea, *et al.*, "Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability," in *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pp. 252–263, 2015.

[12] E. Agirre, C. Banea, D. Cer, M. Diab, A. Gonzalez Agirre, R. Mihalcea, G. Rigau Claramunt, and J. Wiebe, "Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation," in *SemEval-2016. 10th International Workshop on Semantic Evaluation; 2016 Jun 16-17; San Diego, CA. Stroudsburg (PA): ACL; 2016. p. 497-511.*, ACL (Association for Computational Linguistics), 2016.

[13] D. Cer, M. Diab, E. Agirre, I. Lopez-Gazpio, and L. Specia, "SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation," in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, (Vancouver, Canada), pp. 1–14, Association for Computational Linguistics, Aug. 2017.

[14] N. Thakur, N. Reimers, A. Rücklé, A. Srivastava, and I. Gurevych, "Beir: A heterogeneous benchmark for zero-shot evaluation of information retrieval models," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

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

[16] J. Devlin, M.-W. Chang, K. Lee, and K. 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)*, pp. 4171–4186, 2019.

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

[18] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," *arXiv preprint arXiv:1908.10084*, 2019.

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

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

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

[22] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*, pp. 1597–1607, PMLR, 2020.

[23] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, pp. 8748–8763, PMLR, 2021.

[24] S. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 632–642, 2015.

[25] A. Williams, N. Nangia, and S. Bowman, "A broad-coverage challenge corpus for sentence understanding through inference," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1112–1122, 2018.

[26] Y.-S. Chuang, R. Dangovski, H. Luo, Y. Zhang, S. Chang, M. Soljačić, S.-W. Li, W.-T. Yih, Y. Kim, and J. Glass, "Diffcse: Difference-based contrastive learning for sentence embeddings," in *Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2022.

[27] J. Giorgi, O. Nitski, B. Wang, and G. Bader, "Declutr: Deep contrastive learning for unsupervised textual representations," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 879–895, 2021.

[28] C. Xiao, Y. Long, and N. Al Moubayed, "On isotropy, contextualization and learning dynamics of contrastive-based sentence representation learning,"

in *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 12266–12283, 2023.

[29] G. Izacard, M. Caron, L. Hosseini, S. Riedel, P. Bojanowski, A. Joulin, and E. Grave, "Unsupervised dense information retrieval with contrastive learning," *Transactions on Machine Learning Research*, 2022.

[30] C. Xiao, Z. Ye, G. T. Hudson, Z. Sun, P. Blunsom, and N. Al Moubayed, "Can text encoders be deceived by length attack?," 2023.

[31] C. Xiao, Y. Li, G. Hudson, C. Lin, and N. Al Moubayed, "Length is a curse and a blessing for document-level semantics," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 1385–1396, 2023.

[32] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459–9474, 2020.

[33] H. Su, J. Kasai, Y. Wang, Y. Hu, M. Ostendorf, W.-t. Yih, N. A. Smith, L. Zettlemoyer, T. Yu, *et al.*, "One embedder, any task: Instruction-finetuned text embeddings," *arXiv preprint arXiv:2212.09741*, 2022.

[34] A. Asai, T. Schick, P. Lewis, X. Chen, G. Izacard, S. Riedel, H. Hajishirzi, and W.-t. Yih, "Task-aware retrieval with instructions," *arXiv preprint arXiv:2211.09260*, 2022.

[35] L. Wang, N. Yang, X. Huang, L. Yang, R. Majumder, and F. Wei, "Improving text embeddings with large language models," *arXiv preprint arXiv:2401.00368*, 2023.

[36] N. Muennighoff, H. Su, L. Wang, N. Yang, F. Wei, T. Yu, A. Singh, and D. Kiela, "Generative representational instruction tuning," *arXiv preprint arXiv:2402.09906*, 2024.

[37] C. Xiao, G. T. Hudson, and N. A. Moubayed, "Rar-b: Reasoning as retrieval benchmark," *arXiv preprint arXiv:2404.06347*, 2024.

[38] P. Rust, J. F. Lotz, E. Bugliarello, E. Salesky, M. de Lhoneux, and D. Elliott, "Language modelling with pixels," in *The Eleventh International Conference on Learning Representations*, 2023.

[39] M. Tschannen, B. Mustafa, and N. Houlsby, "Clippo: Image-and-language understanding from pixels only," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11006–11017, 2023.

[40] Y. Tai, X. Liao, A. Suglia, and A. Vergari, "Pixar: Auto-regressive language modeling in pixel space," *arXiv preprint arXiv:2401.03321*, 2024.

[41] T. Gao, Z. Wang, A. Bhaskar, and D. Chen, "Improving language understanding from screenshots," *arXiv preprint arXiv:2402.14073*, 2024.

[42] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, *et al.*, "Dinov2: Learning robust visual features without supervision," *Transactions on Machine Learning Research Journal*, pp. 1–31, 2024.

[43] M. Cherti, R. Beaumont, R. Wightman, M. Wortsman, G. Ilharco, C. Gordon, C. Schuhmann, L. Schmidt, and J. Jitsev, "Reproducible scaling laws for contrastive language-image learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2818–2829, 2023.

[44] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer, "Sigmoid loss for language image pre-training," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11975–11986, 2023.

[45] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. Le, Y.-H. Sung, Z. Li, and T. Duerig, "Scaling up visual and vision-language representation learning with noisy text supervision," in *International conference on machine learning*, pp. 4904–4916, PMLR, 2021.

[46] C. Xiao, Z. Huang, D. Chen, G. T. Hudson, Y. Li, H. Duan, C. Lin, J. Fu, J. Han, and N. A. Moubayed, "Pixel sentence representation learning," *arXiv preprint arXiv:2402.08183*, 2024.

[47] Y. Yan, R. Li, S. Wang, F. Zhang, W. Wu, and W. Xu, "Consert: A contrastive framework for self-supervised sentence representation transfer," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 5065–5075, 2021.

[48] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, (New Orleans, Louisiana), pp. 2227–2237, Association for Computational Linguistics, June 2018.

[49] K. Ethayarajh, "How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 55–65, 2019.

[50] J. Gao, D. He, X. Tan, T. Qin, L. Wang, and T. Liu, "Representation degeneration problem in training natural language generation models," in *International Conference on Learning Representations*, 2019.

[51] J. Mu and P. Viswanath, "All-but-the-top: Simple and effective postprocessing for word representations," in *International Conference on Learning Representations*, 2018.

[52] X. Cai, J. Huang, Y. Bian, and K. Church, "Isotropy in the contextual embedding space: Clusters and manifolds," in *International Conference on Learning Representations*, 2020.

[53] A. Merchant, E. Rahimtoroghi, E. Pavlick, and I. Tenney, "What happens to bert embeddings during fine-tuning?," in *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pp. 33–44, 2020.

[54] Y. Hao, L. Dong, F. Wei, and K. Xu, "Investigating learning dynamics of bert fine-tuning," in *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pp. 87–92, 2020.

[55] S. Rajaee and M. T. Pilehvar, "How does fine-tuning affect the geometry of embedding space: A case study on isotropy," in *Findings of the Association for Computational Linguistics: EMNLP 2021*, (Punta Cana, Dominican Republic), pp. 3042–3049, Association for Computational Linguistics, Nov. 2021.

[56] J. Huang, D. Tang, W. Zhong, S. Lu, L. Shou, M. Gong, D. Jiang, and N. Duan, "Whiteningbert: An easy unsupervised sentence embedding approach," in *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 238–244, 2021.

[57] W. Timkey and M. van Schijndel, "All bark and no bite: Rogue dimensions in transformer language models obscure representational quality," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 4527–4546, 2021.

[58] K. Song, X. Tan, T. Qin, J. Lu, and T.-Y. Liu, "Mpnet: Masked and permuted pre-training for language understanding," *Advances in Neural Information Processing Systems*, vol. 33, pp. 16857–16867, 2020.

[59] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, and M. Zhou, "Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers," *Advances in Neural Information Processing Systems*, vol. 33, pp. 5776–5788, 2020.

[60] T. Wang and P. Isola, "Understanding contrastive representation learning through alignment and uniformity on the hypersphere," in *International Conference on Machine Learning*, pp. 9929–9939, PMLR, 2020.

[61] F. Wang and H. Liu, "Understanding the behaviour of contrastive loss," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2495–2504, 2021.

[62] Y. Su, F. Liu, Z. Meng, T. Lan, L. Shu, E. Shareghi, and N. Collier, "TaCL: Improving BERT pre-training with token-aware contrastive learning," in *Findings of the Association for Computational Linguistics: NAACL 2022*, (Seattle, United States), pp. 2497–2507, Association for Computational Linguistics, July 2022.

[63] Y. Zhang, H. Zhu, Y. Wang, N. Xu, X. Li, and B. Zhao, "A contrastive framework for learning sentence representations from pairwise and triple-wise perspective in angular space," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4892–4903, 2022.

[64] K. Sparck Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of documentation*, vol. 28, no. 1, pp. 11–21, 1972.

[65] M. Yuksekgonul, F. Bianchi, P. Kalluri, D. Jurafsky, and J. 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.

[66] X. Wu, C. Gao, L. Zang, J. Han, Z. Wang, and S. Hu, "ESimCSE: Enhanced sample building method for contrastive learning of unsupervised sentence embedding," in *Proceedings of the 29th International Conference on Computational Linguistics*, (Gyeongju, Republic of Korea), International Committee on Computational Linguistics, Oct. 2022.

[67] T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, and L. Deng, "Ms marco: A human generated machine reading comprehension dataset," *choice*, vol. 2640, p. 660, 2016.

[68] N. Reimers, P. Beyer, and I. Gurevych, "Task-oriented intrinsic evaluation of semantic textual similarity," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 87–96, 2016.

[69] K. Wang, N. Reimers, and I. Gurevych, "Tsdae: Using transformer-based sequential denoising auto-encoderfor unsupervised sentence embedding learning," in *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 671–688, 2021.

[70] K. Abe, S. Yokoi, T. Kajiwara, and K. Inui, "Why is sentence similarity benchmark not predictive of application-oriented task performance?," in *Proceedings of the 3rd Workshop on Evaluation and Comparison of NLP Systems*, pp. 70–87, 2022.

[71] J. Devlin, M.-W. Chang, K. Lee, and K. 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)*, (Minneapolis, Minnesota), pp. 4171–4186, Association for Computational Linguistics, June 2019.

[72] E. Agirre, D. Cer, M. Diab, A. Gonzalez-Agirre, and W. Guo, "* sem 2013 shared task: Semantic textual similarity," in *Second joint conference on lexical and computational semantics (* SEM), volume 1: proceedings of the Main conference and the shared task: semantic textual similarity*, pp. 32–43, 2013.

[73] A. Neelakantan, T. Xu, R. Puri, A. Radford, J. M. Han, J. Tworek, Q. Yuan, N. Tezak, J. W. Kim, C. Hallacy, *et al.*, "Text and code embeddings by contrastive pre-training," *arXiv preprint arXiv:2201.10005*, 2022.

[74] N. Muennighoff, N. Tazi, L. Magne, and N. Reimers, "Mteb: Massive text embedding benchmark," in *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 2014–2037, 2023.

[75] A. Deshpande, C. E. Jimenez, H. Chen, V. Murahari, V. Graf, T. Rajpurohit, A. Kalyan, D. Chen, and K. Narasimhan, "Csts: Conditional semantic textual similarity," *arXiv preprint arXiv:2305.15093*, 2023.

[76] P. Xu, W. Ping, X. Wu, L. McAfee, C. Zhu, Z. Liu, S. Subramanian, E. Bakhturina, M. Shoeybi, and B. Catanzaro, "Retrieval meets long context large language models," *arXiv preprint arXiv:2310.03025*, 2023.

[77] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, and H. Wang, "Retrieval-augmented generation for large language models: A survey," *arXiv preprint arXiv:2312.10997*, 2023.

[78] OpenAI, "Gpt-4.," 2023.

[79] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[80] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung, "Survey of hallucination in natural language generation," *ACM Computing Surveys*, vol. 55, no. 12, pp. 1–38, 2023.

[81] T. Vu, M. Iyyer, X. Wang, N. Constant, J. Wei, J. Wei, C. Tar, Y.-H. Sung, D. Zhou, Q. Le, *et al.*, "Freshllms: Refreshing large language models with search engine augmentation," *arXiv preprint arXiv:2310.03214*, 2023.

[82] N. Kandpal, H. Deng, A. Roberts, E. Wallace, and C. Raffel, "Large language models struggle to learn long-tail knowledge," in *International Conference on Machine Learning*, pp. 15696–15707, PMLR, 2023.

[83] C. Malaviya, S. Lee, S. Chen, E. Sieber, M. Yatskar, and D. Roth, "Expertqa: Expert-curated questions and attributed answers," *arXiv preprint arXiv:2309.07852*, 2023.

[84] L. Berglund, M. Tong, M. Kaufmann, M. Balesni, A. C. Stickland, T. Korbak, and O. Evans, "The reversal curse: Llms trained on "a is b" fail to learn "b is a"," in *The Twelfth International Conference on Learning Representations*, 2023.

[85] P. BehnamGhader, S. Miret, and S. Reddy, "Can retriever-augmented language models reason? the blame game between the retriever and the language model," in *Findings of the Association for Computational Linguistics: EMNLP*

*2023* (H. Bouamor, J. Pino, and K. Bali, eds.), (Singapore), pp. 15492–15509, Association for Computational Linguistics, Dec. 2023.

[86] A. Asai, Z. Zhong, D. Chen, P. W. Koh, L. Zettlemoyer, H. Hajishirzi, and W.-t. Yih, "Reliable, adaptable, and attributable language models with retrieval," *arXiv preprint arXiv:2403.03187*, 2024.

[87] Y. Bai, X. Lv, J. Zhang, H. Lyu, J. Tang, Z. Huang, Z. Du, X. Liu, A. Zeng, L. Hou, *et al.*, "Longbench: A bilingual, multitask benchmark for long context understanding," *arXiv preprint arXiv:2308.14508*, 2023.

[88] P. Clark, O. Etzioni, T. Khot, A. Sabharwal, O. Tafjord, P. Turney, and D. Khashabi, "Combining retrieval, statistics, and inference to answer elementary science questions," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, 2016.

[89] A. Rogers, O. Kovaleva, M. Downey, and A. Rumshisky, "Getting closer to ai complete question answering: A set of prerequisite real tasks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, pp. 8722–8731, 2020.

[90] K. Sakaguchi, R. L. Bras, C. Bhagavatula, and Y. Choi, "Winogrande: An adversarial winograd schema challenge at scale," *Communications of the ACM*, vol. 64, no. 9, pp. 99–106, 2021.

[91] P. Clark, I. Cowhey, O. Etzioni, T. Khot, A. Sabharwal, C. Schoenick, and O. Tafjord, "Think you have solved question answering? try arc, the ai2 reasoning challenge," *arXiv preprint arXiv:1803.05457*, 2018.

[92] C. Bhagavatula, R. Le Bras, C. Malaviya, K. Sakaguchi, A. Holtzman, H. Rashkin, D. Downey, W.-t. Yih, and Y. Choi, "Abductive commonsense reasoning," in *International Conference on Learning Representations*, 2020.

[93] R. Zellers, A. Holtzman, Y. Bisk, A. Farhadi, and Y. Choi, "Hellaswag: Can a machine really finish your sentence?," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, 2019.

[94] Y. Bisk, R. Zellers, J. Gao, Y. Choi, *et al.*, "Piqa: Reasoning about physical commonsense in natural language," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, pp. 7432–7439, 2020.

[95] M. Sap, H. Rashkin, D. Chen, R. Le Bras, and Y. Choi, "Social iqa: Commonsense reasoning about social interactions," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4463–4473, 2019.

[96] Q. Tan, H. T. Ng, and L. Bing, "Towards benchmarking and improving the temporal reasoning capability of large language models," *arXiv preprint arXiv:2306.08952*, 2023.

[97] R. Mirzaee, H. R. Faghihi, Q. Ning, and P. Kordjamshidi, "Spartqa: A textual question answering benchmark for spatial reasoning," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4582–4598, 2021.

[98] D. Hendrycks, C. Burns, S. Kadavath, A. Arora, S. Basart, E. Tang, D. Song, and J. Steinhardt, "Measuring mathematical problem solving with the math dataset," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[99] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, *et al.*, "Training verifiers to solve math word problems," *arXiv preprint arXiv:2110.14168*, 2021.

[100] L. Yu, W. Jiang, H. Shi, J. Yu, Z. Liu, Y. Zhang, J. T. Kwok, Z. Li, A. Weller, and W. Liu, "Metamath: Bootstrap your own mathematical questions for large language models," *arXiv preprint arXiv:2309.12284*, 2023.

[101] H. de Swart, *Introduction to Natural Language Semantics*. Stanford, CA: CSLI Publications, 1998. Numerous exercises punctuate each chapter and an example exam based on the materials presented is included, making this volume a perfect textbook and resource for any undergraduate or graduate-level introductory course in semantics.

[102] N. Muennighoff, Q. Liu, A. Zebaze, Q. Zheng, B. Hui, T. Y. Zhuo, S. Singh, X. Tang, L. Von Werra, and S. Longpre, "Octopack: Instruction tuning code large language models," *arXiv preprint arXiv:2308.07124*, 2023.

[103] J. Austin, A. Odena, M. Nye, M. Bosma, H. Michalewski, D. Dohan, E. Jiang, C. Cai, M. Terry, Q. Le, *et al.*, "Program synthesis with large language models," *arXiv preprint arXiv:2108.07732*, 2021.

[104] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. d. O. Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, *et al.*, "Evaluating large language models trained on code," *arXiv preprint arXiv:2107.03374*, 2021.

[105] H. Husain, H.-H. Wu, T. Gazit, M. Allamanis, and M. Brockschmidt, "Codesearchnet challenge: Evaluating the state of semantic code search," *arXiv preprint arXiv:1909.09436*, 2019.

[106] X. Wang, Q. Yang, Y. Qiu, J. Liang, Q. He, Z. Gu, Y. Xiao, and W. Wang, "Knowledgpt: Enhancing large language models with retrieval and storage access on knowledge bases," *arXiv preprint arXiv:2308.11761*, 2023.

[107] X. Ma, Y. Gong, P. He, H. Zhao, and N. Duan, "Query rewriting for retrieval-augmented large language models," *arXiv preprint arXiv:2305.14283*, 2023.

[108] O. Yoran, T. Wolfson, B. Bogin, U. Katz, D. Deutch, and J. Berant, "Answering questions by meta-reasoning over multiple chains of thought," *arXiv preprint arXiv:2304.13007*, 2023.

[109] Z. Zhang, X. Zhang, Y. Ren, S. Shi, M. Han, Y. Wu, R. Lai, and Z. Cao, "Iag: Induction-augmented generation framework for answering reasoning questions," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 1–14, 2023.

[110] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, "Self-rag: Learning to retrieve, generate, and critique through self-reflection," *arXiv preprint arXiv:2310.11511*, 2023.

[111] S.-C. Lin, A. Asai, M. Li, B. Oguz, J. Lin, Y. Mehdad, W.-t. Yih, and X. Chen, "How to train your dragon: Diverse augmentation towards generalizable dense retrieval," *arXiv preprint arXiv:2302.07452*, 2023.

[112] J. Chen, S. Xiao, P. Zhang, K. Luo, D. Lian, and Z. Liu, "Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation," *arXiv preprint arXiv:2402.03216*, 2024.

[113] S. Xiao, Z. Liu, P. Zhang, and N. Muennighof, "C-pack: Packaged resources to advance general chinese embedding," *arXiv preprint arXiv:2309.07597*, 2023.

[114] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, *et al.*, "Mistral 7b," *arXiv preprint arXiv:2310.06825*, 2023.

[115] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 5485–5551, 2020.

[116] N. Lourie, R. Le Bras, C. Bhagavatula, and Y. Choi, "Unicorn on rainbow: A universal commonsense reasoning model on a new multitask benchmark," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 13480–13488, 2021.

[117] J. Wei, M. Bosma, V. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le, "Finetuned language models are zero-shot learners," in *International Conference on Learning Representations*, 2021.

[118] T. Wang, A. Roberts, D. Hesslow, T. Le Scao, H. W. Chung, I. Beltagy, J. Launay, and C. Raffel, "What language model architecture and pretraining objective works best for zero-shot generalization?," in *International Conference on Machine Learning*, pp. 22964–22984, PMLR, 2022.

[119] Y. Wang, S. Mishra, P. Alipoormolabashi, Y. Kordi, A. Mirzaei, A. Naik, A. Ashok, A. S. Dhanasekaran, A. Arunkumar, D. Stap, *et al.*, "Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 5085–5109, 2022.

[120] J. Ni, C. Qu, J. Lu, Z. Dai, G. H. Abrego, J. Ma, V. Zhao, Y. Luan, K. Hall, M.-W. Chang, *et al.*, "Large dual encoders are generalizable retrievers," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 9844–9855, 2022.

[121] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, "Scaling laws for neural language models," *arXiv preprint arXiv:2001.08361*, 2020.

[122] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. d. L. Casas, L. A. Hendricks, J. Welbl, A. Clark, *et al.*, "Training compute-optimal large language models," *arXiv preprint arXiv:2203.15556*, 2022.

[123] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, *et al.*, "Emergent abilities of large language models," *Transactions on Machine Learning Research*, 2022.

[124] N. Muennighoff, "Sgpt: Gpt sentence embeddings for semantic search," *arXiv preprint arXiv:2202.08904*, 2022.

[125] W. Sun, L. Yan, X. Ma, S. Wang, P. Ren, Z. Chen, D. Yin, and Z. Ren, "Is chatgpt good at search? investigating large language models as re-ranking agents," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 14918–14937, 2023.

[126] R. Pradeep, S. Sharifymoghaddam, and J. Lin, "Rankzephyr: Effective and robust zero-shot listwise reranking is a breeze!," *arXiv preprint arXiv:2312.02724*, 2023.

[127] H. Fang, S. Wang, M. Zhou, J. Ding, and P. Xie, "Cert: Contrastive self-supervised learning for language understanding," *arXiv preprint arXiv:2005.12766*, 2020.

[128] Z. Wu, S. Wang, J. Gu, M. Khabsa, F. Sun, and H. Ma, "Clear: Contrastive learning for sentence representation," *arXiv preprint arXiv:2012.15466*, 2020.

[129] Y. Meng, C. Xiong, P. Bajaj, P. Bennett, J. Han, X. Song, *et al.*, "Coco-lm: Correcting and contrasting text sequences for language model pretraining," *Advances in Neural Information Processing Systems*, vol. 34, pp. 23102–23114, 2021.

[130] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *54th Annual Meeting of the Association for Computational Linguistics*, pp. 1715–1725, Association for Computational Linguistics (ACL), 2016.

[131] G. Rawlinson, "The significance of letter position in word recognition," in *PhD thesis*, University of Nottingham, 1976.

[132] F. Ferreira and N. D. Patson, "The 'good enough'approach to language comprehension," *Language and linguistics compass*, vol. 1, no. 1-2, pp. 71–83, 2007.

[133] J. Grainger and J. C. Ziegler, "A dual-route approach to orthographic processing," *Frontiers in psychology*, vol. 2, p. 54, 2011.

[134] K. Rayner, A. Pollatsek, J. Ashby, and C. Clifton Jr, "Psychology of reading," 2012.

[135] P. May, "Machine translated multilingual sts benchmark dataset.," 2021.

[136] J. Li, S. Ji, T. Du, B. Li, and T. Wang, "Textbugger: Generating adversarial text against real-world applications," in *26th Annual Network and Distributed System Security Symposium*, 2019.

[137] J. X. Morris, E. Lifland, J. Y. Yoo, J. Grigsby, D. Jin, and Y. Qi, "Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp," *arXiv preprint arXiv:2005.05909*, 2020.

[138] K. Lee, M.-W. Chang, and K. Toutanova, "Latent retrieval for weakly supervised open domain question answering," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 6086–6096, 2019.

[139] J. Tiedemann, "Parallel data, tools and interfaces in opus.," in *Lrec*, vol. 2012, pp. 2214–2218, 2012.

[140] G. Lample, A. Conneau, M. Ranzato, L. Denoyer, and H. Jégou, "Word translation without parallel data," in *International Conference on Learning Representations*, 2018.

[141] L. Barrault, O. Bojar, M. R. Costa-Jussa, C. Federmann, M. Fishel, Y. Graham, B. Haddow, M. Huck, P. Koehn, S. Malmasi, *et al.*, "Findings of the 2019 conference on machine translation (wmt19)," ACL, 2019.

[142] N. Reimers and I. Gurevych, "Making monolingual sentence embeddings multilingual using knowledge distillation," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4512–4525, 2020.

[143] H. Schwenk, V. Chaudhary, S. Sun, H. Gong, and F. Guzmán, "Wikimatrix: Mining 135m parallel sentences in 1620 language pairs from wikipedia," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 1351–1361, 2021.

[144] M. Marelli, L. Bentivogli, M. Baroni, R. Bernardi, S. Menini, and R. Zamparelli, "Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment," in *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, pp. 1–8, 2014.

[145] T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee, *et al.*, "Natural questions: a benchmark for question answering research," *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 453–466, 2019.

[146] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. Cohen, R. Salakhutdinov, and C. D. Manning, "Hotpotqa: A dataset for diverse, explainable multi-hop question answering," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, 2018.

[147] A. Cohan, S. Feldman, I. Beltagy, D. Downey, and D. S. Weld, "Specter: Document-level representation learning using citation-informed transformers," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2270–2282, 2020.

[148] D. Wadden, S. Lin, K. Lo, L. L. Wang, M. van Zuylen, A. Cohan, and H. Hajishirzi, "Fact or fiction: Verifying scientific claims," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7534–7550, 2020.

[149] V. Boteva, D. Gholipour, A. Sokolov, and S. Riezler, "A full-text learning to rank dataset for medical information retrieval," in *Advances in Information Retrieval: 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20–23, 2016. Proceedings 38*, pp. 716–722, Springer, 2016.

[150] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

[151] O. Taylor, "Pango, an open-source unicode text layout engine," in *Proceedings of 25th Internationalization and Unicode Conference*, 2004.

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

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

[154] G. Izacard, M. Caron, L. Hosseini, S. Riedel, P. Bojanowski, A. Joulin, and E. Grave, "Unsupervised dense information retrieval with contrastive learning," *arXiv preprint arXiv:2112.09118*, 2021.

[155] J. H. Clark, D. Garrette, I. Turc, and J. Wieting, "Canine: Pre-training an efficient tokenization-free encoder for language representation," *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 73–91, 2022.

[156] L. Xue, A. Barua, N. Constant, R. Al-Rfou, S. Narang, M. Kale, A. Roberts, and C. Raffel, "Byt5: Towards a token-free future with pre-trained byte-to-byte

models," *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 291–306, 2022.

[157] S. Edunov, M. Ott, M. Auli, and D. Grangier, "Understanding back-translation at scale," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 489–500, 2018.

[158] D. Jin, Z. Jin, J. T. Zhou, and P. Szolovits, "Is bert really robust? a strong baseline for natural language attack on text classification and entailment," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, pp. 8018–8025, 2020.

[159] X. Geng, H. Zhang, J. Bian, and T.-S. Chua, "Learning image and user features for recommendation in social networks," in *Proceedings of the IEEE international conference on computer vision*, pp. 4274–4282, 2015.

[160] A. Zhai, H.-Y. Wu, E. Tzeng, D. H. Park, and C. Rosenberg, "Learning a unified embedding for visual search at pinterest," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, (New York, NY, USA), p. 2412–2420, Association for Computing Machinery, 2019.

[161] J.-T. Huang, A. Sharma, S. Sun, L. Xia, D. Zhang, P. Pronin, J. Padmanabhan, G. Ottaviano, and L. Yang, "Embedding-based retrieval in facebook search," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2553–2561, 2020.

[162] C. Wei, Y. Chen, H. Chen, H. Hu, G. Zhang, J. Fu, A. Ritter, and W. Chen, "UniIR: Training and benchmarking universal multimodal information retrievers," *arXiv preprint arXiv:2311.17136*, 2023.

[163] T. Weyand, A. Araujo, B. Cao, and J. Sim, "Google landmarks dataset v2 - a large-scale benchmark for instance-level recognition and retrieval," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[164] N.-A. Ypsilantis, N. Garcia, G. Han, S. Ibrahimi, N. Van Noord, and G. Tolias, "The Met dataset: Instance-level recognition for artworks," in *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[165] R. Datta, D. Joshi, J. Li, and J. Z. Wang, "Image retrieval: Ideas, influences, and trends of the new age," *ACM Computing Surveys (Csur)*, vol. 40, no. 2, pp. 1–60, 2008.

[166] M. Faysse, H. Sibille, T. Wu, B. Omrani, G. Viaud, C. Hudelot, and P. Colombo, "Colpali: Efficient document retrieval with vision language models," 2024.

[167] G. Alain and Y. Bengio, "Understanding intermediate layers using linear classifier probes," 2018.

[168] X. Chen, S. Xie, and K. He, "An empirical study of training self-supervised vision transformers," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9640–9649, 2021.

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

[170] A. Collignon, F. Maes, D. Delaere, D. Vandermeulen, P. Suetens, G. Marchal, *et al.*, "Automated multi-modality image registration based on information theory," in *Information processing in medical imaging*, vol. 3, pp. 263–274, Citeseer, 1995.

[171] C. Studholme, D. L. Hill, and D. J. Hawkes, "An overlap invariant entropy measure of 3d medical image alignment," *Pattern recognition*, vol. 32, no. 1, pp. 71–86, 1999.

[172] S. Kornblith, J. Shlens, and Q. V. Le, "Do better imagenet models transfer better?," 2019.

[173] A. Koukounas, G. Mastrapas, M. Günther, B. Wang, S. Martens, I. Mohr, S. Sturua, M. K. Akram, J. F. Martínez, S. Ognawala, *et al.*, "Jina clip: Your clip model is also your text retriever," *arXiv preprint arXiv:2405.20204*, 2024.

[174] S. Y. Gadre, G. Ilharco, A. Fang, J. Hayase, G. Smyrnis, T. Nguyen, R. Marten, M. Wortsman, D. Ghosh, J. Zhang, *et al.*, "Datacomp: In search of the next generation of multimodal datasets," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[175] Q. Sun, Y. Fang, L. Wu, X. Wang, and Y. Cao, "Eva-clip: Improved training techniques for clip at scale," *arXiv preprint arXiv:2303.15389*, 2023.

[176] J. Zhou, Z. Liu, S. Xiao, B. Zhao, and Y. Xiong, "Vista: Visualized text embedding for universal multi-modal retrieval," *arXiv preprint arXiv:2406.04292*, 2024.

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

[178] Z. Jiang, R. Meng, X. Yang, S. Yavuz, Y. Zhou, and W. Chen, "Vlm2vec: Training vision-language models for massive multimodal embedding tasks," *arXiv preprint arXiv:2410.05160*, 2024.

[179] "voyage-multimodal-3: all-in-one embedding model for interleaved text, images, and screenshots." `https://blog.voyageai.com/2024/11/12/voyage-multimodal-3/`.

[180] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," *Advances in neural information processing systems*, vol. 36, pp. 34892–34916, 2023.

[181] K. Enevoldsen, M. Kardos, N. Muennighoff, and K. L. Nielbo, "The scandinavian embedding benchmarks: Comprehensive assessment of multilingual and monolingual text embedding," *Advances in Neural Information Processing Systems*, vol. 37, pp. 40336–40358, 2024.

[182] A. V. Thapliyal, J. P. Tuset, X. Chen, and R. Soricut, "Crossmodal-3600: A massively multilingual multimodal evaluation dataset," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 715–729, 2022.

[183] E. Bugliarello, F. Liu, J. Pfeiffer, S. Reddy, D. Elliott, E. M. Ponti, and I. Vulić, "IGLUE: A benchmark for transfer learning across modalities, tasks, and languages," in *Proceedings of the 39th International Conference on Machine Learning* (K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, eds.), vol. 162 of *Proceedings of Machine Learning Research*, pp. 2370–2392, PMLR, 17–23 Jul 2022.

[184] H. Liu, C. Li, Y. Li, and Y. J. Lee, "Improved baselines with visual instruction tuning," 2023.

[185] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, *et al.*, "Lora: Low-rank adaptation of large language models.," *ICLR*, vol. 1, no. 2, p. 3, 2022.

[186] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (L. Màrquez, C. Callison-Burch, and J. Su, eds.), (Lisbon, Portugal), pp. 632–642, Association for Computational Linguistics, Sept. 2015.

[187] A. Williams, N. Nangia, and S. Bowman, "A broad-coverage challenge corpus for sentence understanding through inference," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (M. Walker, H. Ji, and A. Stent, eds.), (New Orleans, Louisiana), pp. 1112–1122, Association for Computational Linguistics, June 2018.

[188] T. Thrush, R. Jiang, M. Bartolo, A. Singh, A. Williams, D. Kiela, and C. Ross, "Winoground: Probing vision and language models for visio-linguistic compositionality," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5238–5248, June 2022.

[189] A. Jaech, A. Kalai, A. Lerer, A. Richardson, A. El-Kishky, A. Low, A. Helyar, A. Madry, A. Beutel, A. Carney, *et al.*, "Openai o1 system card," *arXiv preprint arXiv:2412.16720*, 2024.

[190] D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi, *et al.*, "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning," *arXiv preprint arXiv:2501.12948*, 2025.

[191] J. Li, D. Li, C. Xiong, and S. Hoi, "Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation," in *International conference on machine learning*, pp. 12888–12900, PMLR, 2022.

[192] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: bootstrapping language-image pre-training with frozen image encoders and large language models," in *Proceedings of the 40th International Conference on Machine Learning*, ICML'23, JMLR.org, 2023.

[193] S. Yang, B. Zhai, Q. You, J. Yuan, H. Yang, and C. Xu, "Law of vision representation in mllms," *arXiv preprint arXiv:2408.16357*, 2024.

[194] K. Enevoldsen, I. Chung, I. Kerboua, M. Kardos, A. Mathur, D. Stap, J. Gala, W. Siblini, D. Krzemiński, G. I. Winata, S. Sturua, S. Utpala, M. Ciancone, M. Schaeffer, G. Sequeira, D. Misra, S. Dhakal, J. Rystrøm, R. Solomatin, Ömer Çağatan, A. Kundu, M. Bernstorff, S. Xiao, A. Sukhlecha, B. Pahwa, R. Poświata, K. K. GV, S. Ashraf, D. Auras, B. Plüster, J. P. Harries, L. Magne, I. Mohr, M. Hendriksen, D. Zhu, H. Gisserot-Boukhlef, T. Aarsen, J. Kostkan, K. Wojtasik, T. Lee, M. Šuppa, C. Zhang, R. Rocca, M. Hamdy, A. Michail, J. Yang, M. Faysse, A. Vatolin, N. Thakur, M. Dey, D. Vasani, P. Chitale, S. Tedeschi, N. Tai, A. Snegirev, M. Günther, M. Xia, W. Shi, X. H. Lù, J. Clive, G. Krishnakumar, A. Maksimova, S. Wehrli, M. Tikhonova, H. Panchal, A. Abramov, M. Ostendorff, Z. Liu, S. Clematide, L. J. Miranda, A. Fenogenova, G. Song, R. B. Safi, W.-D. Li, A. Borghini, F. Cassano, H. Su, J. Lin, H. Yen, L. Hansen, S. Hooker, C. Xiao, V. Adlakha, O. Weller, S. Reddy, and N. Muennighoff, "Mmteb: Massive multilingual text embedding benchmark," 2025.

[195] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.

[196] L. McInnes, J. Healy, S. Astels, *et al.*, "hdbscan: Hierarchical density based clustering.," *J. Open Source Softw.*, vol. 2, no. 11, p. 205, 2017.

[197] N.-A. Ypsilantis, K. Chen, B. Cao, M. Lipovskỳ, P. Dogan-Schönberger, G. Makosa, B. Bluntschli, M. Seyedhosseini, O. Chum, and A. Araujo, "Towards universal image embeddings: A large-scale dataset and challenge for generic image representations," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11290–11301, 2023.

[198] Y. Jin, M. Choi, G. Verma, J. Wang, and S. Kumar, "Mm-soc: Benchmarking multimodal large language models in social media platforms," in *ACL*, 2024.

[199] Y. Liu, Z. Li, M. Huang, B. Yang, W. Yu, C. Li, X. Yin, C. lin Liu, L. Jin, and X. Bai, "Ocrbench: On the hidden mystery of ocr in large multimodal models," 2024.

[200] S.-C. Lin, C. Lee, M. Shoeybi, J. Lin, B. Catanzaro, and W. Ping, "MM-EMBED: UNIVERSAL MULTIMODAL RETRIEVAL WITH MULTIMODAL

LLMS," in *The Thirteenth International Conference on Learning Representations*, 2025.

[201] X. Zhang, Y. Zhang, W. Xie, M. Li, Z. Dai, D. Long, P. Xie, M. Zhang, W. Li, and M. Zhang, "Gme: Improving universal multimodal retrieval by multimodal llms," 2024.

[202] X. Fu, Y. Hu, B. Li, Y. Feng, H. Wang, X. Lin, D. Roth, N. A. Smith, W.-C. Ma, and R. Krishna, "Blink: Multimodal large language models can see but not perceive," *arXiv preprint arXiv:2404.12390*, 2024.

[203] Z. Liu, C. Rodriguez-Opazo, D. Teney, and S. Gould, "Image retrieval on real-life images with pre-trained vision-and-language models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2125–2134, 2021.

[204] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona, "Caltech-ucsd birds 200," 09 2010.

[205] S. Liu, W. Feng, T.-J. Fu, W. Chen, and W. Wang, "Edis: Entity-driven image search over multimodal web content," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4877–4894, 2023.

[206] X. Han, Z. Wu, P. X. Huang, X. Zhang, M. Zhu, Y. Li, Y. Zhao, and L. S. Davis, "Automatic spatially-aware fashion concept discovery," in *Proceedings of the IEEE international conference on computer vision*, pp. 1463–1471, 2017.

[207] H. Wu, Y. Gao, X. Guo, Z. Al-Halah, S. Rennie, K. Grauman, and R. Feris, "Fashion iq: A new dataset towards retrieving images by natural language feedback," in *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pp. 11307–11317, 2021.

[208] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier, "From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions," *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 67–78, 2014.

[209] P. Wu, S. Wang, K. D. Rosa, and D. H. Hu, "Forb: A flat object retrieval benchmark for universal image embedding," 2023.

[210] D. Kiela, H. Firooz, A. Mohan, V. Goswami, A. Singh, P. Ringshia, and D. Testuggine, "The hateful memes challenge: Detecting hate speech in multimodal memes," *Advances in neural information processing systems*, vol. 33, pp. 2611–2624, 2020.

[211] Y. Chen, H. Hu, Y. Luan, H. Sun, S. Changpinyo, A. Ritter, and M.-W. Chang, "Can pre-trained vision and language models answer visual information-seeking questions?," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 14948–14968, 2023.

[212] C. Sharma, D. Bhageria, W. Scott, S. Pykl, A. Das, T. Chakraborty, V. Pulabaigari, and B. Gambäck, "Semeval-2020 task 8: Memotion analysis-the visuo-lingual metaphor!," in *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pp. 759–773, 2020.

[213] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755, Springer, 2014.

[214] S. Fu, N. Tamir, S. Sundaram, L. Chai, R. Zhang, T. Dekel, and P. Isola, "Dreamsim: Learning new dimensions of human visual similarity using synthetic data," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[215] H. Hu, Y. Luan, Y. Chen, U. Khandelwal, M. Joshi, K. Lee, K. Toutanova, and M.-W. Chang, "Open-domain visual entity recognition: Towards recognizing millions of wikipedia entities," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12065–12075, 2023.

[216] F. Radenović, A. Iscen, G. Tolias, Y. Avrithis, and O. Chum, "Revisiting oxford and paris: Large-scale image retrieval benchmarking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[217] J. Peng, C. Xiao, and Y. Li, "Rp2k: A large-scale retail product dataset for fine-grained image classification," *arXiv preprint arXiv:2006.12634*, 2020.

[218] S. Wu, Y. Li, K. Zhu, G. Zhang, Y. Liang, K. Ma, C. Xiao, H. Zhang, B. Yang, W. Chen, W. Huang, N. A. Moubayed, J. Fu, and C. Lin, "Scimmir: Benchmarking scientific multi-modal information retrieval," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL), findings*, 2024.

[219] H. Oh Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep metric learning via lifted structured feature embedding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4004–4012, 2016.

[220] J. Krause, J. Deng, M. Stark, and L. Fei-Fei, "Collecting a large-scale dataset of fine-grained cars," 2013.

[221] M. Eitz, J. Hays, and M. Alexa, "How do humans sketch objects?," *ACM Transactions on graphics (TOG)*, vol. 31, no. 4, pp. 1–10, 2012.

[222] F. Liu, Y. Wang, T. Wang, and V. Ordonez, "Visual news: Benchmark and challenges in news image captioning," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6761–6771, 2021.

[223] D. Gurari, Q. Li, A. J. Stangl, A. Guo, C. Lin, K. Grauman, J. Luo, and J. P. Bigham, "Vizwiz grand challenge: Answering visual questions from blind people," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3608–3617, 2018.

[224] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh, "Making the v in vqa matter: Elevating the role of image understanding in visual question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[225] Y. Chang, M. Narang, H. Suzuki, G. Cao, J. Gao, and Y. Bisk, "Webqa: Multihop and multimodal qa," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16495–16504, 2022.

[226] T. Berg, J. Liu, S. Woo Lee, M. L. Alexander, D. W. Jacobs, and P. N. Belhumeur, "Birdsnap: Large-scale fine-grained visual categorization of birds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[227] L. 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*, pp. 178–178, 2004.

[228] A. Krizhevsky, "Learning multiple layers of features from tiny images," tech. rep., 2009.

[229] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi, "Describing textures in the wild," in *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[230] P. Helber, B. Bischke, A. Dengel, and D. 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*, vol. 12, no. 7, pp. 2217–2226, 2019.

[231] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2015.

[232] S. Maji, E. Rahtu, J. Kannala, M. Blaschko, and A. Vedaldi, "Fine-grained visual classification of aircraft," 2013.

[233] L. Bossard, M. Guillaumin, and L. Van Gool, "Food-101 – mining discriminative components with random forests," in *European Conference on Computer Vision*, 2014.

[234] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The german traffic sign recognition benchmark: A multi-class classification competition," in *The 2011 International Joint Conference on Neural Networks*, pp. 1453–1460, 2011.

[235] Y. LeCun, C. Cortes, and C. Burges, "Mnist handwritten digit database," *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, vol. 2, 2010.

[236] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pp. 722–729, 2008.

[237] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. V. Jawahar, "Cats and dogs," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3498–3505, 2012.

[238] B. S. Veeling, J. Linmans, J. Winkens, T. Cohen, and M. Welling, "Rotation equivariant cnns for digital pathology," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018* (A. F. Frangi, J. A. Schnabel, C. Davatzikos, C. Alberola-López, and G. Fichtinger, eds.), (Cham), pp. 210–218, Springer International Publishing, 2018.

[239] G. Cheng, J. Han, and X. Lu, "Remote sensing image scene classification: Benchmark and state of the art," *Proceedings of the IEEE*, vol. 105, no. 10, pp. 1865–1883, 2017.

[240] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics* (G. Gordon, D. Dunson, and M. Dudík, eds.), vol. 15 of *Proceedings of Machine Learning Research*, (Fort Lauderdale, FL, USA), pp. 215–223, PMLR, 11–13 Apr 2011.

[241] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "Sun database: Large-scale scene recognition from abbey to zoo," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3485–3492, 2010.

[242] K. Soomro, A. R. Zamir, and M. Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," 2012.

[243] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, pp. 303–338, June 2010.

[244] Y. Le and X. S. Yang, "Tiny imagenet visual recognition challenge," 2015.

[245] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. Lawrence Zitnick, and R. Girshick, "Clevr: A diagnostic dataset for compositional language and elementary visual reasoning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[246] M. Yuksekgonul, F. Bianchi, P. Kalluri, D. Jurafsky, and J. 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.

[247] C.-Y. Hsieh, J. Zhang, Z. Ma, A. Kembhavi, and R. Krishna, "Sugarcrepe: Fixing hackable benchmarks for vision-language compositionality," in *Thirty-Seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

[248] B. Krojer, V. Adlakha, V. Vineet, Y. Goyal, E. Ponti, and S. Reddy, "Image retrieval from contextual descriptions," *arXiv preprint arXiv:2203.15867*, 2022.

[249] Z. Nussbaum, J. X. Morris, B. Duderstadt, and A. Mulyar, "Nomic embed: Training a reproducible long context text embedder," 2024.

[250] "Nomic embed vision: Expanding the nomic latent space." https://www.nomic.ai/blog/posts/nomic-embed-vision.

[251] C. Xiao, I. Chung, I. Kerboua, J. Stirling, X. Zhang, M. Kardos, R. Solomatin, N. Al Moubayed, K. Enevoldsen, and N. Muennighoff, "Mieb: Massive image embedding benchmark," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22187–22198, 2025.

[252] C. Xiao, H. P. Chan, H. Zhang, M. Aljunied, L. Bing, N. Al Moubayed, and Y. Rong, "Analyzing LLMs' knowledge boundary cognition across languages through the lens of internal representations," in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar, eds.), (Vienna, Austria), pp. 24099–24115, Association for Computational Linguistics, July 2025.

[253] C. Xiao, H. P. Chan, H. Zhang, W. Xu, M. Aljunied, and Y. Rong, "Scaling language-centric omnimodal representation learning," *arXiv preprint arXiv:2510.11693*, 2025.

[254] R. Yuan, C. Xiao, S. Leng, J. Wang, L. Li, W. Xu, H. P. Chan, D. Zhao, T. Xu, Z. Wei, *et al.*, "Vl-cogito: Progressive curriculum reinforcement learning for advanced multimodal reasoning," *arXiv preprint arXiv:2507.22607*, 2025.

[255] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[256] S. Nie, F. Zhu, Z. You, X. Zhang, J. Ou, J. Hu, J. Zhou, Y. Lin, J.-R. Wen, and C. Li, "Large language diffusion models," *arXiv preprint arXiv:2502.09992*, 2025.

[257] X. Chen, Z. Liu, S. Xie, and K. He, "Deconstructing denoising diffusion models for self-supervised learning," in *The Thirteenth International Conference on Learning Representations*, 2025.