ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Alma Mater Studiorum · Università di Bologna

Dipartimento di Informatica — Scienza e Ingegneria
Corso di Laurea Triennale in Informatica

Analyzing CLIP's Limits: Mitigating Its Problems and Defining Its Structural Boundaries

Relatore:
Chiar.mo
Prof. Andrea Asperti

Presentata da:
Luca Marangon

Sessione Luglio 2026
Anno Accademico 2025/2026

To my family,
which always supported me with a smile, regardless of anything.

Contents

Introduction

CLIP (Contrastive Language-Image Pre-training) is a vision-language model first developed by OpenAI in 2021. It immediately gained great popularity thanks to its zero-shot capabilities and structural simplicity, becoming the most widespread tool for image-text association. As the model's architecture is open source, countless variations have been created over the years, and many papers have focused on studying the capabilities of this instrument. In this Thesis, the objective is to analyze the limitations and weaknesses of this incredible tool, as well as some of the attempts made to improve it.

First, we will describe some of the most commonly encountered limitations of CLIP, such as its difficulty in recognizing relationships between objects, misinterpretation of negation in prompts, inability to count with precision, and difficulty in capturing the global structure (such as recognizing the image's style).

Then the Thesis will explore some of the possible solutions developed in the years, citing a few of the existing Open models implementing them.

On a final note, we will discuss the possible structural limitation intrinsic to CLIP, noted by recent articles, that propose the necessity of considerable structural changes in the model.

Chapter 1

Introduction to CLIP — Architecture and training

1.1 Architecture

As mentioned, CLIP has gained great popularity; as such, there are many variants and open models. While some of them attempt to exploit the weights of OpenAI's model by adding a simple finetuning, others focus on training the model from scratch with open source datasets (since the massive dataset OpenAI used for training is not publicly available). To start, we will quickly describe the original architecture solution explored by OpenAI's model in 2021.[16]

The architecture is divided into an image encoder and a text encoder, both sharing the same encoding space. The model associates each image with the most "similar" text caption based on a cosine similarity function of the respective encodings.

For the visual encoder, the original CLIP of 2022 used a modified Vit model with an attention pooling layer instead of the global average pooling layer (called ViT-L/14@336px). While for the text encoder, it used a transformer created ad hoc: a 12-layer 512-wide model with 8 attention heads.

Today, dozens of model alterations have been experimented, ranging from slight adjustments in the encoders or the similarity function, to more innovative solutions with a trade off between training complexity and model accuracy.

1.2 Dataset

The original OpenAI model was trained on a new database of 400 million image-text pairs sourced from the internet, especially created to train CLIP. To create it, after deciding on 500,000 different queries, the web was searched for image-caption pairs whose caption contained such query as a substring (for every query up to a maximum of 20,000 images were sampled, to ensure a balanced dataset).[16] This training set has been called WIT for WebImageText, unluckily it is not openly available but limited to OpenAI's internal use.

1.3 Training

The training technique chosen was contrastive learning: the dataset is divided into sets of N images, then the model is trained to maximize cosine similarity in the N correct image-text matches while maximizing the distance of all other possible pairings (NxN -N "wrong" pairs). The original loss function was a symmetric cross-entropy loss.

The decision of using contrastive learning comes from scalability reasons: experimenting, the bag of words contrastive approach took half the computation time and learned twelve times faster compared to training a Transformer Language Model to predict and generate the exact words of the caption.[16]

1.4 Zero shot transfer

As we mentioned, contrastive learning is much faster to train, but it can only associate already existing captions with images, and not generate ones from scratch. So how can CLIP be used effectively for other tasks? To apply the model to new tasks without fine-tuning (what is called zero-shot transfer), it is sufficient to use simple prompt engineering. For example, to perform class recognition tasks, it is sufficient to prepare a series of captions like "a photo of [CLASS]", CLIP then associates each image to one of such captions, effectively performing classification.

1.5 Results

When using comparison datasets, Zero-shot CLIP beats finetuned ResNet-50 on 16 out of 27 cases, demonstrating its incredible generalization capabilities.[16] However, its limits in complex tasks are also recognized in the original introductory article. Examples are: counting task, tumor recognition, and self driving tasks. As we will see such problems can be eased with finetuning using appropriate datasets, but complex image analysis remains a great limit for CLIP.

Chapter 2

Main limitations and problems

2.1 Introduction

Many articles explored not only CLIP's potential, but also its limitations: the main ones we will focus on are:

- Compositional reasoning
- Negation
- Fine grained recognition and spatial perception
- Object hallucination
- Counting
- Parroting
- Biases
- Human-like perception

2.2 Compositional reasoning

Studies show that CLIP cannot reliably distinguish the relationships between objects. For example, it is unable to reliably associate a color to the correct item in multi-object scenarios[17, 8], or recognize whether "the dog is behind the tree' or "the tree is behind the dog" [20].

A Study[8] compares zero-shot CLIP, finetuned CLIP, and a variant using compositional soft prompting designed to improve compositionality. The new model fine-tunes the embeddings of tokens representing adjective or concepts, while keeping other parameters of the encoders frozen. The tests were performed on a new benchmark divided in three datasets: single-object(various shapes and colors), two-object, and relational(shapes and relations like "in front or behind"), all generated with Blender.

- Single Adjective-Noun Composition: frozen CLIP outperforms all other models (even fine-tuned CLIP, likely due to overfitting) with a 92% accuracy.

- Two-Object Adjective-Noun Binding: performance drop significantly for every model, but frozen CLIP still outperforms the others with 31% accuracy on 5 options (random chance at 20%).
- Relational Composition: the model has to recognize the spatial relationship between objects, all CLIP models show disastrous results, obtaining results worse than random chance and close to 0% accuracy (These results hints at a training bias too, where only relationships encountered in training are considered[17]).

Error analysis on relational composition shows that the mistakes are almost exclusively caused by the model recognizing the object but not the correct spatial relationship, confirming that the problem is related to spatial recognition.

2.2.1 Bag of words

CLIP's difficulty in recognizing attribute binding and spatial recognition is one of its most pressing limitations. Its tendency to treat text captions as unordered collections of concepts suggests a behavior similar to bag-of-words (BoW) [20, 6]. To test this hypothesis, an experiment was performed perturbing captions (words are shuffled) and images (divided into 4 or 9 parts and shuffled). The resulting performance loss was only marginal, confirming that high performances can be obtained without order or composition information, thus suggesting that CLIP's contrastive pretraining does not incentivize their encoding.[20]

2.2.2 CAB

Another study tests CLIP on color-object association, and shows the same results: almost perfect accuracy for single object images, but performances drop even below random chance when a second object is introduced[17]. To explain these results, it is theorized again that CLIP does not correctly bind objects and attributes, processing them in a "bag of concepts" approach[20, 17].

When observing results on a "two-object dataset", most mistakes share a common source: when asked for the color of object A, CLIP systematically predicts the color of object B instead. This is confirmed by testing with a new evaluation condition, "Two objects*", where accuracy is measured by considering the color of the unqueried object B as the ground truth. Under this condition, accuracy is close to that of the single-object case, revealing that CLIP's errors are not random but are driven by a consistent bias.

Thus, a "Concept Association Bias" (CAB) is identified: CLIP creates strong associations between certain objects and attributes, confusing the two as the same thing. For example, a strong association was discovered between eggplant and purple: when the model was presented with an image of a lemon and an eggplant, it associated the caption "purple lemon" (as "purple" can apparently substitute for the concept of an "eggplant"). It is further confirmed by a test containing only unnaturally colored objects: in this case, CLIP predicts the color of the object choosing randomly between the two colors in the image. This highlights CLIP's lack of object-attribute binding, which is probably the underlying cause for its confusion between objects and attributes, treating them as interchangeable within their concept bag. Notably, CAB is found in real-world datasets

like VQA-v2 (where 50 out of 100 mistakes were attributable to CAB), and in attributes different from color (as similar results were obtained with part-whole relationships such as trees and leaves).[17].

2.3 Negation

CLIP often fails to process linguistic negation. When presented with a prompt such as 'a park with no benches', CLIP retrieves images of parks with benches, likely due to the suspected "bag of concept" approach [20, 6], as the dominant signal is the noun 'bench' rather than its negation.

This failure seems to mainly originate from a training dataset bias: for example in the LAION-400M dataset (used to train OpenCLIP) only about 0.7% of captions contain negation terms, many not corresponding to recognizable content in the image (for example: "St Louis Style Ribs - Perfect every time. You will NOT fail with this technique.") [14].

Interestingly other studies point out that this problem also stems from CLIP's structural limits, caused by the geometry of the encoding space using cosine similarity. As the encoding space seems to be unable to consider more problems at once, finetuning to better recognize negations may achieve improvements in that area but negatively impact others, such as attribute binding or spatial locations and relationships recognition[5]. We will further describe this structural limit later, but its interesting to take note of.

2.4 Fine grained recognition and spatial perception

CLIP and the MLLM's that use it seem to have big limitations when facing questions about simple details of an image, The theory is that the problem stems from encoding space's nature: two similar images may share an almost identical encoding, so extracting detailed information from the encoding may prove futile [5, 18].

A study especially explores the possibility, purposively selecting such "similar" pairs (called blind-pairs) and testing various CLIP models on nine different categories ("Orientation and Direction","Presence of Specific Features", "State and Condition", "Quantity and Count", "Positional and Relational Context", "Color and Appearance", "Structural Characteristics", "A Text", "Viewpoint and Perspective"). The results show how all CLIP models perform poorly in all categories, with the sole exception of "color and appearance" and "state and condition" [18].

While these results confirm CLIP's encoding limits, it has been proven that the encoding space is able to store some level of fine-grained information: experiments that substitute cosine similarity with different similarity functions obtain better results in fine-grained recognition, but in exchange, performances drop for coarse grained ones [4].

2.5 Object hallucination

In an attempt to understand the causes of hallucinations in LVLMs integrating CLIP, a new OHD benchmark (Object Hallucination Detection) has been created by adding incorrect captions to multiple datasets: either mentioning "hallucinatory objects" or removing objects from the caption; the idea is to train the model through negative examples. Such hallucinatory objects are divided into: "random objects", "popular objects" (popular in the dataset) and "adversarial objects" (objects most frequently paired with the actual image object). The results obtained testing various CLIP variants on the benchmark (NegCLIP, CeCLIP, EVA CLIP, MetaCLIP, CLIP ConvNext, etc.) show that all variants perform better than vanilla CLIP, but are still unsatisfactory, all reaching below 40% accuracy[11].

The cause of such hallucinations is plausibly a bias caused by the large web crawled training datasets, and the study shows that solving such problem cannot be achieved simply by finetuning tasks.

2.6 Human-like perception

CLIP seems to perceive images in a different way than humans; critical examples are the blindness to generative artifacts and the inability to accurately recognize or produce stylistic nuances. Although various text-to-image models can generate images able to fool humans as real paintings (in an experiment 28% of AI-generated images were mistaken as real[3]), such models still struggle avoiding errors obvious to the human eye. Examples are artifacts (abnormal number of fingers, misplaced limbs, etc.), distortions (distorted or missing parts, sketched figures, etc.), tendency to hyperrealism (regardless of the prompted style) and anachronisms (such as the insertion of modern objects in images generated in older styles) [3].

A recent study focuses exactly on investigating this stylistic limitations, testing various zero-shot CLIP models (models without finetuning) on four different datasets collectively containing both real paintings and AI-generated ones (National Gallery of Art Dataset, AI-ArtBench, WikiArt, and AI-Pastiche). [2]

2.6.1 Style Recognition

When three CLIP models were tasked to associate images with style based prompts ("An image in [CLASS] style"), results of show zero-shot recognition are modest and unstable across different datasets. Furthermore, a three dimensional projection of the high dimensional embeddings is able to visually reveals the distinct separation between image encodings and stylistic text encodings (Figure 1).

To further explore the reason for the disappointing results, the successive experiment used CLIP's image encodings to train a linear classifier. Such classifier shows better accuracy, but its performance drops significantly when it is used on datasets which were not part of the training. This means that the problem does not solely lie in image-text encoding alignment (as suggested by the clustered encodings) but also in CLIP's apparent inability to really generalize abstract concepts (such as brushstrokes, palette,

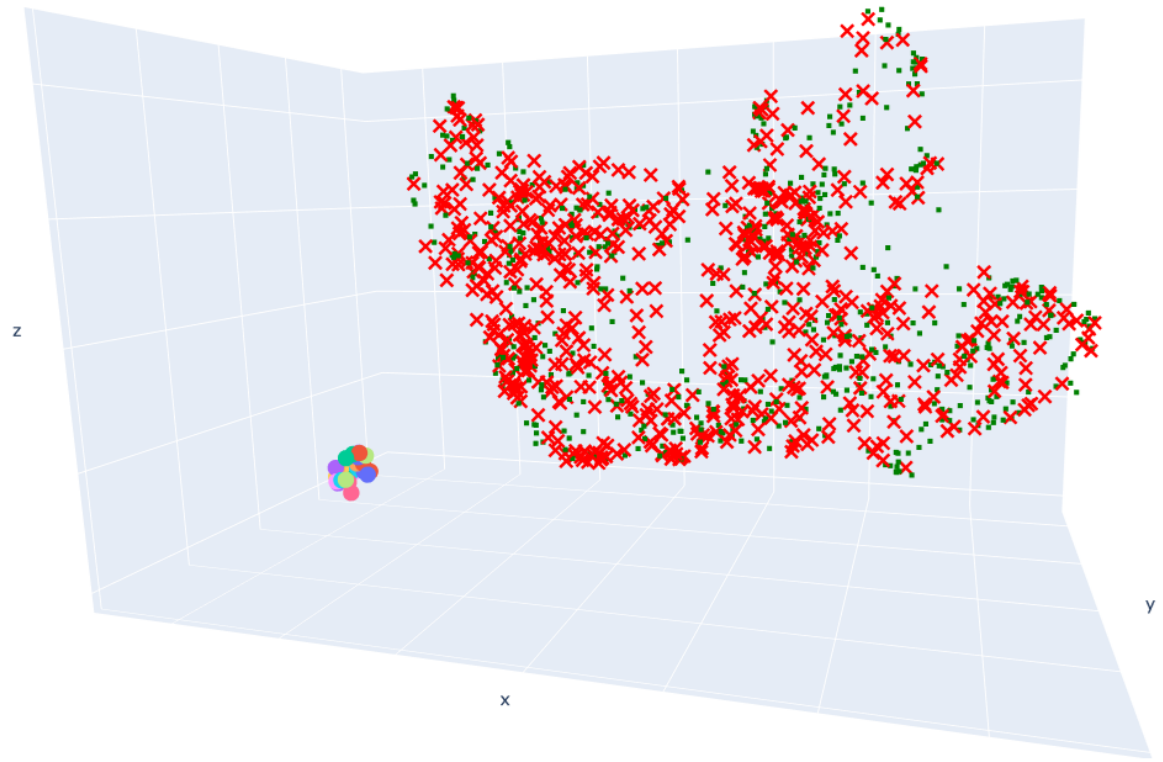

Figure 1: text encodings are clustered on the left, image encodings are the clustered on the right, clearly separated

and composition) beyond the quirks of a single dataset.

2.6.2 Human-like perception and artifacts recognition

But is CLIP just bad at recognizing styles or has a different perception than humans? To find out, CLIP's similarity scores between images and prompts were calculated, then it was compared to human given ratings using cosine similarity. The best CLIP models manage a meager 0.437 cosine similarity between their scores and human evaluations, confirming the difference between human and machine criteria for prompt adherence [2]. In particular, CLIP has been shown to be almost completely blind to low-level structural flaws easily recognized by humans, such as incorrectly placed limbs, and other artifacts of image generation.

2.6.3 Conclusions on image perception

In conclusion, although CLIP shows excellent performance in recognizing semantic content and objects, its perception is very different from human one. This may be caused by a lack of adequate datasets for proper training, often lacking many human concepts, as even the four datasets especially selected for the style recognition study show considerable limitations in their context (such as wikiArt's prompts being often very similar to each other). However, it may also lie in the inherent limits in CLIP's encoding space, as other studies revealed possible problems in such space [5] and in CLIP's ability to align image and text encodings [6].

2.7 Dataset biases

2.7.1 Negation

As already expressed, the failure to process negation is directly linked to a bias in the datasets used for training: as they are usually large collections of caption-image pairs obtained from the web with little filtering, there is a severe lack of negation examples (rarely a caption will express what is *not* present in the image)[14].

Furthermore, the datasets' lack of quality also causes many other biases carried over from the web, here we analyze some of them.

2.7.2 Parroting

CLIP disproportionately relies on text appearing on the image to associate captions, a possible cause was found in the dataset bias: captions partially or completely identical to the text embedded into the corresponding image (parrot captions) make up a considerable portion of internet examples, and are thus also found in datasets like LAION-2B (used for most OpenCLIP training). Tests have shown that CLIP's score reduces drastically when tested on such parroting images with the embedded text removed, confirming its over-reliance on such embeddings. To demonstrate that such reliance on parrot captioning is a negative shortcut that stops CLIP from learning meaningful attributes, experiments training CLIP with parroting-dense datasets were performed: while it improves text reading capabilities, it reduces zero-shot generalization and visual task performance considerably. Instead, removing the embedded text shows a performance improvement[10].

2.7.3 Counting

VLMs utilizing CLIP show significant limitations when tasked with precise counting, generating an image with an incorrect number of items, or even an image with a number embedded (case attributable to a parroting bias[10]). It has been observed that the limit does not stems from CLIP's architecture, but from a dataset bias in image captions. Humans rarely count with precision when more than four objects appear, so such numbers seldom appear in the captions; moreover, the numbers present in captions may refer to non-descriptive information (e.g. age, date, etc.) resulting useless or confusing for the training. Finetuning with a dataset of selected examples has shown improvement in counting capabilities, but the model is still held back by the difficulty to retrieve a sizable number of captions containing numbers above six [13].

2.7.4 Objects' relationships

As already mentioned, CLIP struggles when tasked with identifying multiple objects and their relationships accurately. Testing on (semantically) equivalent but (structurally) varied captions and analyzing the mistakes, biases can be found in both of CLIP's encoders:

- CLIP's image encoder prefers larger objects.

- CLIP's text encoder favors both the first object mentioned, and the visually larger ones (e.g. "house" has more weight than "bag").

These findings are derived by generating embeddings for captions/images containing multiple objects and comparing them against embeddings of each individual object present in the image; the single-object embedding most similar to the full image embedding is taken as the most influential object. [1].

The visual encoder bias can be attributed to the ViT architecture: larger objects that occupy more image space exert more influence, as it is a common problem in all ViT models.

The text encoder biases are more difficult to explain, but experiments find an association between bigger objects and first mentioned objects: analyzing LAION and COCO datasets it is notable that in most cases, the largest object in an image is mentioned earlier in its caption. Thus, the bias could be explained as the visual encoder bias "carrying over" from the image encoder during contrastive learning.

2.7.5 Bias Conclusion

One of CLIP's strengths lies in its intensive training over numerous example images (400,000 millions for OpenAI's CLIP) that allow it to have incredible zero-shot performance. However, such datasets are mainly composed of unfiltered captions and images from the internet. This creates strong biases not only in a prospective of gender, race, and culture, but also for a series of other instances such as counting, negation, etc.

In this respect, both negative hard samples and fine-tuning have shown promising results in addressing this issue, and their combined use may help achieve a less biased CLIP model.

Chapter 3

Improvements and fixes

How to improve on CLIP's weaknesses? Many different paths have been explored, achieving gains over the base model in different areas. Common approaches include improving dataset quality, increasing model complexity, or refining the loss function to produce better encodings. Below, we examine some of the most notable solutions.

3.1 CAB (Concept Association Bias)

Mitigating CAB: Deepening modality interaction to mitigate CAB showed limited effectiveness [17]. Applying an additional transformer to CLIP and subsequent finetuning it on VQA-v2 does reduce CAB for naturally colored objects (meaning the model gets better at identifying an object's own color when another object is present), but also reduces accuracy for unnaturally colored ones, suggesting the model learns to exploit statistical associations rather than developing genuine object-attribute binding.

3.2 Dataset filtering/enriching

Simple dataset filtering can address many of the problems of CLIP's Biases, for example, removing images with text embeddings[10], or adding finetuning with captions rich in negations[14], or explicit numbering of objects[13] significantly improves performance on these problems. This results suggest that CLIP's capabilities may actually be limited by the great unsupervised training datasets of web images, and improving the dataset quality may influence the performance more directly than simply increasing the training dataset size.

3.2.1 Visually Enriched Captions

What if we change the training direction to one more focused on data quality? As web images have shown numerous biases this may be an interesting option, one that VeCLIP explores[7]. Noticing how web-crawled images have many noisy or uninformative descriptions (like the address as caption for a house) the idea is that of Visually Enriched Captions(VeCap): first a multimodal model (such as LLava) is used to extract text information from the images, secondly using an LLM (such as Vicuna-1.1) the original

captions are improved and fused it with the new information extracted from the image. A limitation lies in the consistent style of LLMs; to enhance data diversity, the enriched captions are alternated to the originals during training. Training results are promising, as VeCLIP (trained using VeCap) learns faster and shows better performances than base Open CLIP models trained with the same number of pairs. In the experiments performed with 3M, 12M, 100M and 200M pairs (of the Vicuna-1.1-13B dataset) VeCLIP always outperforms CLIP, especially for ItoT (Image to Text) and TtoI (Text to Image) tasks [7]. Still, it must be noted that VeCLIP does not reach OpenAI's CLIP likely due to the poor quality of Vicuna-1.1-13B when compared to the private OpenAI's dataset (on ImageNet and ImageNetV2, VeCLIP at 200M pair reaches 64.6% and 57.7% against OpenAI's 76.2% and 70.1%).[16, 7]

3.2.2 Text token length

CLIP's OpenAI model has a limit of 77 tokens for text captions, and tests discovered that the actual number of tokens the model manages to consider is even lower: just about 20 [22]. To unlock the possibility of using longer captions while avoiding training a new model from scratch, a useful method is interpolating the model's positional embeddings: CLIP's original model contains 77 positional embedding vectors, each associated to one text token position, and to increase the number of acceptable tokens we would need more positional embedding vectors, altering the model's structure.

Without altering the vector number, using interpolation we can calculate new vectors from the existing 77, and each token position is assigned an embedding computed as a weighted average of multiple vectors (out of those 77). To avoid retraining the model, Long-CLIP makes an intelligent use of this interpolation: starting from pre-trained CLIP, and having observed how the first 20 positional embeddings are already well-trained, only interpolates from position 21 onward, "stretching" the 57 embeddings to cover up to 228 positions (reaching a total of 20+228=248 positions)[22].

As the embeddings from the 21th position are altered, Long-CLIP can't avoid finetuning. During such training, to avoid degrading the already good short-text capabilities, each image is represented by two feature vectors: a fine-grained one aligned to the long caption, and a coarse-grained one (obtained by retaining only the most important components of the fine-grained vector) aligned to a shorter summary caption. This allows the model to learn both detailed and key-element representations.

LongCLIP significantly outperforms the baseline for fine-grained tasks (i.e. short or long text-image retrieval) while suffering only minor degradation for coarse-grained classification ones [22].

3.3 Training

In this section we will mention some of the ideas that are based on altering CLIP's training to improve its performances.

3.3.1 Dataset

OpenAI's CLIP architecture is freely available, yet the 400 million image-caption pairs used to train it are not. This poses some limitations in understanding and improving CLIP's training. Many OpenCLIP variations may struggle to choose a dataset, and while LAION has obtained moderate popularity, it has shown its bias and limitations [14, 10, 1]. To address this problem, a promising solution may lie in MetaCLIP: a research aiming to create a transparent and performing dataset for CLIP training [19]. The idea is to emulate the process described to generate CLIP's dataset[16] rather than relying on filters and CLIP as a teacher model to create the dataset

1. Generating 500,000-queries as metadata (such queries are needed to search the internet for significant captions).
2. Searching for web images whose captions contain at least a metadata query as a substring (extracting from CommonCrawl, 1.6B pairs were found).
3. It was noted that most pairs matched only with a limited number of queries (with more than 100,000 queries without any match). Thus to balance distribution following CLIP's example, at most 20,000 pairs are kept for every query (after confirming 20,000 to be excellent threshold).
4. 400M pairs remain after such filtering, which are used to train the CLIP model from zero.
5. Lastly, the same process is repeated with a bigger pool of 10.7B pairs, obtaining two datasets: one of of 1B maintaining the 20k threshold), and another one of 2.5B using an increased threshold of 170k).

Results show that all MetaCLIP models slightly outperform OpenAI CLIP(+1% accuracy compared to OpenAI). And the models trained on the 1B and 2.5B datasets show further, albeit limited, improvement (a further +1% accuracy)[19]. This solution underlines the importance of good training dataset quality other than its size, and finds a more transparent alternative to OpenAI's CLIP.

3.3.2 Self supervised learning

Self supervised learning works by training the model on data without human intervention: for example, the model may be trained to encode images in such a way that it remains similar to the encoding of its cropped version; obtaining an encoder capable to extract the most important features of an image.

Observations point out that CLIP's encoders may be biased towards improving similarity between caption and image encodings, even at the cost of feature extraction [1]. A possible solution may reside in self-supervised pre-training, as it is capable of leading to improvements on challenging visual tasks. SLIP experiments on this idea, altering the loss function to consider both the standard CLIP Loss and an SSL Loss (a loss trying to minimize encoding differences between an image and its slightly altered version)[12]. The model must process approximately 3x activations due to the increased number of images processed, slowing down training. However, it shows considerable improvements compared to a CLIP model trained on the same YFCC15M dataset (it scores 42.8%

accuracy vs CLIP's 37.6% on ImageNet). Not only the results show accuracy improvements, but the average cosine similarity between correct image-embedding pairs is much higher for SLIP, confirming an improvement in features extraction.

3.3.3 Training speed — how can training get more accessible

Many promising solutions require training CLIP's model from scratch (like the self supervised training we just mentioned), as such they have a big drawback: the cost in time and computational power. Fortunately some experiments to improve on this aspect have been performed already:

Reducing computation time — inverse scaling law in CLIP's training

As mentioned, training models require a lot of resources. A first idea may be to just train using data with smaller size to reduce computational costs, so using images and captions with smaller token length, but is the data quality loss worth it? A Study actually reveals what can be called an "inverse scaling law" for CLIP models: the larger the model is, the less it is sensitive to the length of image/text tokens used in training[9].

The study purposefully reduces the size of images and captions used during training, and observes only a small reduction in model accuracy. In particular the best results were obtained with 'image resizing' (instead of masking) and text 'syntax masking' (where certain words, like nouns and adjectives, are prioritized excluding the rest of the caption).[9].

Results on text token length reduction are particularly interesting: not only accuracy loss is extremely low up to 16 tokens, but can also improve accuracy in larger models. Still it must be noted that the results are likely influenced by low quality, noisy captions in the training dataset of LAION-400M[9], and even if they do match with observations on how only the first 20 tokens are fully considered by the model [22], this may mean that reducing token text length in better-quality datasets may degrade performances more.

To fully test the theory, many variations of a new ACLIP model were trained with reduced pairs, then finetuned with the full token length. The results show performances comparable, if not superior, to OpenCLIP, while requiring only from 1/10 to 1/33 of GPU hours for training (depending on model size)[9].

Alternative training method

But what about changing the training method itself? CLIP's training uses a softmax as a loss function. For every pair, it tunes the encoders' parameter with the objective of maximizing cosine similarity of all correct pairing, while simultaneously minimizing that of all incorrect ones; to do so it must globally normalize all similarity vectors in the batch, and requires big batches to get good results (to obtain a generalized solution that does not depend on the limited examples of a small batch). A possible alternative is to substitute the softmax with a sigmoid-based Loss, processed independently for every caption-image pair (basically the problem becomes a binary classification for each possible combination of caption and image). Testing shows that a Sigmoid Loss Function performs better than the softmax baseline, especially for small batches, and has even

better accuracy than OpenAI's CLIP.[21]. The main benefits found using sigmoid loss compared to softmax:

- better performance for small batch sizes, also optimal batch size is smaller (at 32k instead of softmax's 98k).
- training requires less memory (compared to same-batch-size softmax)
- interestingly, sigmoid function is less sensitive to noise in the dataset, a common problem in the big datasets used for CLIP training (usually web-crawled ones). This is likely a consequence of the new training concept: with softmax each result is dependent on every pair in the batch, with sigmoid every pair is considered separately.

3.4 Hard negatives

CLIP's encoders may focus excessively on matching text and image encodings rather than extracting complete information. NegCLIP attempts to enhance the model's ability to encode features through an improvement in the training dataset. In particular, NegCLIP is the first to introduce the idea of hard negatives.[20]:

Hard negatives are textual prompts generated starting from existing ones, altering only minor attributes. During training, they are added to the dataset as additional incorrect options, with the intent of forcing the encoder to pay attention to details and learn to distinguish them.

Its greatest strengths are:

- The possibility of easily generating good hard negative captions with LLMs, using little time and resources and favoring scalability.
- The ease of use, as it is sufficient to finetune already existing models using hard negatives.
- Hard negatives can be generated for different kinds of problem, and are a flexible solution.

3.4.1 Compositional understanding

NegCLIP has been the first CLIP model to use hard negatives in training[20]. The intent is to improve CLIP's understandings in compositional relationships, attributes binding, and words order; limiting the "bag of words"-like behavior.

- For every caption, a negative one is generated swapping different linguistic elements (noun phrases, nouns, adjectives, adverbs, verb phrases).
- For every image in the training batch, a "strong alternative" is selected and added to the batch choosing between the K=3 most similar images in the dataset (after computing the similarity score for every pair of images). The intent is to force the model to distinguish the finer details of the image.

NegCLIP is generated finetuning Vit-B/32 with these hard negatives. During training the cosine similarity is calculated between the N images and 2N captions (half of which are hard negatives) without computing the loss for hard negative captions, as they do not have a corresponding image.

Tests on tasks requiring sensitivity to order or compositionality show great improvement, while the model maintains comparable results for other downstream tasks. These results show the efficacy and ease of use of hard negative samples even just for finetuning. However, it still does not fully exploit the hard negatives' characteristics: contrastive learning does neither maximize the distance between hard negatives and the corresponding image, nor takes into consideration the "partial correctness" of such captions.

An evolution that addresses this problem is the technique used in CeCLIP[23], where:

- An added intra-modal contrastive loss is added to promote the model's ability to distinguish correct captions from the similar hard negatives (cross-entropy that maximizes the distance between the caption's encoding and the ones corresponding to the hard negatives)
- To ensure the model learns the "similarity" between correct captions and its hard negatives other than simply distinguishing them, instead of a standard contrastive loss (that would simply maximize the distance) a new loss using a threshold is used: the loss only ensures the similarity between the image and the correct caption is greater than the one with the hard negative by a certain threshold. To do so a hinge loss is applied.

The final loss combines the standard image-text contrastive loss $\mathcal{L}_{\text{itc}}$, the text-negative contrastive loss $\mathcal{L}_{\text{tnc}}$, and the image-negative rank loss $\mathcal{L}_{\text{inr}}$, weighted by hyperparameters α and β:

$$\mathcal{L} = \mathcal{L}_{\text{itc}} + \alpha \cdot \mathcal{L}_{\text{tnc}} + \beta \cdot \mathcal{L}_{\text{inr}} \tag{3.1}$$

Results show further improvements compared to NegClip; But at the same time, there is an increase in model complexity and, possibly, training time.

3.4.2 Images as hard negatives

Another experiment aimed at improving the model's compositionality recognition beyond that of NegCLIP is TripletCLIP: the idea is adding both hard negative captions and corresponding images generated ad-hoc, forming pairs to train the model[15].

Firstly, improved negative captions were generated using LLMs (compared to NegCLIP, which shuffles nouns and attributes, all the captions have a semantically correct meaning), then images were generated starting from such captions using a text-to-image model. Experiments show that, unlike captions, images alone cannot improve the model significantly, but they can still help regularize the training. For that the loss is calculated on two triplets (Im,T,T') and (Im',T,T') where Im' and T' is the generated image text pair hard negative for (Im,T). Training LaCLIP, NegCLIP and Tripletclip from scratch shows how Tripletclip considerably outperforms the other models especially for compositional understanding. However, when using this method to finetune OpenAI's

CLIP, while the compositional understanding improves, zero-shot performance drops, this result is likely due to CLIP's structural limitations: finetuning for compositional understanding generality is lost[5].

3.4.3 Reducing hallucinations

To reduce CLIP's hallucinations, an attempt was made using captions containing various kind of hallucinatory objects as hard negatives: The loss function was modified to maximize the distance between images and these hard negatives, with an added margin loss ensuring the similarity to the correct caption always exceeds that of any hallucination-containing one by a fixed threshold. Results obtained finetuning CLIP ViT/32-B show significant improvements compared to vanilla CLIP in hallucination recognition (from 14.3% to 82.5%) while maintaining comparable zero-shot performance (65.6% vs CLIP's 66.0%) [11].

It is interesting to note that applying the same fine-tuning to NegCLIP and CECLIP moderately improves hallucination recognition but significantly decreases zero-shot performance [11]. This may be caused by the fundamental geometric constraints of CLIP's joint embedding framework, which has been formally shown to be incapable of simultaneously satisfying multiple semantic conditions using cosine similarity as its scoring objective[5].

3.4.4 Final notes

More than one type of hard negatives can be applied at the same time for the caption, like in CeCLIP (where 4 are used together, calculating the Loss accordingly)[23]. This means that if CLIP's structural limits of cosine similarity are solved, this solution can potentially address more fine-grained problems at once.

Chapter 4

Structural limits

So CLIP has many limits that can be improved on, but where do they stem from? various solutions like fine-tuning, dataset cleaning, and even modifications to its structure have been attempted with varying degree of success, facing a single problem at a time. In this section we analyze the reasons that may be intrinsically behind such limitations.

4.1 Text representation — limits of similarity

CLIP's training method is based on maximizing the cosine similarity of the encoded text vector and the encoded image vector, but is this objective ideal? Analyzing in a formal, mathematical way the cosine similarity between two n-dimensional random vectors, a study on this topic makes two observations: [1]

1. Minimizing cosine similarity for negative pairs: for vectors of n dimension, as n increases, the similarity between two random vectors tends to zero. This means that CLIP's encoders may need little to no modification to maximize the distance between unrelated images and captions, making it useless to focus on elements in the image that do not appear in the captions (and that may help distinguish differences in the images).

2. Maximizing cosine similarity for positive pairs: given an image encoder $i_\theta()$ and text encoder $t_\omega()$, maximizing cosine similarity we want to obtain $i_\theta(I) = z$ and $t_\omega(T) = z$ where z is a complete latent representation of the caption content. But a maximized cosine similarity may as well be obtained by $i_{\theta'}(I) = z_l$ and $t_{\omega'}(T) = z_l$ where z_l is only a limited subset of all caption content. In other words, as long as the text and image encores are aligned, they do not need to extract all information to maximize cosine similarity.

While it should be noted that this limit is merely theoretical, and does not necessarily represent the behavior of the encoders, the results obtained by methods using hard negatives to force the encoders to compare between similar captions or images, have shown considerable success, managing to both enhance general understanding[20] and reducing hallucinations[11].

4.2 Encoding space — single vector limitations

A recent study[5] has analyzed CLIP's latent space geometry in a mathematical, formal way, and demonstrated that (even for a subset of the real world objects, attributes and compositional concepts) an 'ideal CLIP space' cannot be obtained.

For 'ideal CLIP space' it is intended a CLIP-like encoder generating single vectors capable of precisely representing similarity (as closedness) and differences (as distance) for these four fields simultaneously:

- content (object or concepts)
- Attributes (and correct binding).
- Spatial relationships.
- Negation

It is demonstrated that no more than one condition can be completely fulfilled at a time[5]. Is this an insurmountable problem that limits CLIP to a coarse-grained tool? Apparently not necessarily: while it seems that retraining, finetuning, re-projecting the embedding, or even applying a new scoring mechanism cannot overcome CLIP's limits alone, new methods to substitute CLIP's representation of text and images as single vectors can be explored.

This theory explains the outcomes obtained in other experiments:

- Attempts to address the encoding space limitations by substituting the cosine similarity function with more complex ones, do manage to improve fine-grained understanding, but degrade the coarse grained one[4].
- when using a series of hard negative examples to finetune CLIP to both improve object attributes binding *and* reduce hallucination (finetuning NegCLIP for hallucination recognition), general performances degrade[11].

In the article an attempt to substitute single-vector encoding is made using Dense cosine similarity maps: big tables calculating the cosine similarity for every image patch and text token encoded by CLIP (instead of doing it just for the single image and text vectors). The general structure is shown on figure 2.

To exploit CLIP's initial training, the model takes the already trained CLIP's encoders, removing the CLS token (the single summary vector for images) and the EOS token (the single summary vector for text), then adds only a lightweight 2-layer CNN to substitute the cosine similarity function.

Instead of calculating the image-text distance with a single cosine similarity, it generates a table comparing the distance of every token and every patch, then the CNN interprets it. To resolve the problem of functional words not directly corresponding to patches in the images (like concepts of "right of", "above", etc.) the authors simply substitute the entire row with constant vectors with random entries (called Functional Rows), to teach the model to ignore them instead of counting on them.

The CNN can be trained with very small batches and little data because its task is not contrastive — it does not need to compare examples against each other to get a

meaningful learning signal. Each DCSM matrix is an independent input, and the light CNN simply learns to recognize structural patterns in it. This allowed training with batches of just 8 samples (CLIP's original batch size is 32,768) and only trained on 20,000 images total.

The solution significantly outperforms both base CLIP and common variants (OpenCLIP, NegCLIP, SigLIP, BLIP, CoCa) on many datasets (COCO, CLEVR, NCD, etc.), likely also thanks to the increased dimensionality of the scoring representation.

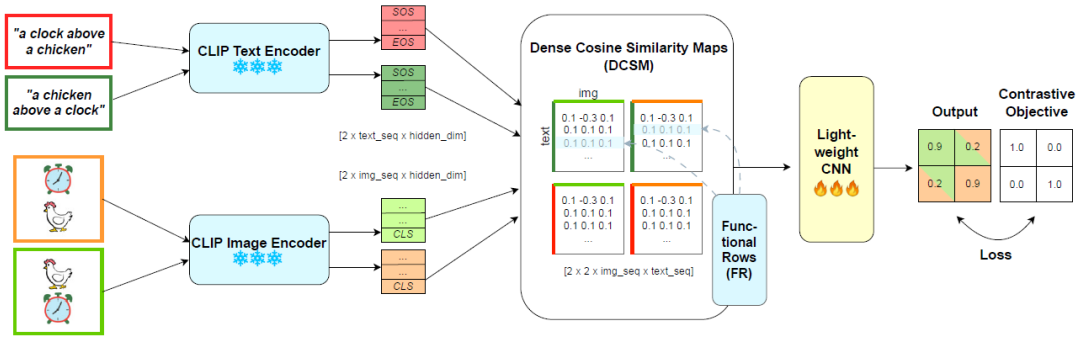

Figure 2: Schematics of our proposed pipeline and training of its scoring function

The model is still far from perfect, and does not necessarily represent the definitive solution. Some of its problems are:

- It is currently unable to handle functional words which must be discarded rather than interpreted.
- As it does not 'summarize' the encoding in a single vector, the encodings for long and rich captions (that may originally reach 77 tokens) become too complex for a limited 2 layer CNN to interpret (the best solution proposed by the paper is to reduce the longer captions to more concise ones using an LLM before passing them to the model).

It is also to note that as such model uses CLIP's frozen encoders, it does not bypass the limitations of the training using cosine similarity, which may limit the encoders' ability to extract information (and was pointed out in the previous section). Still the theoretical demonstration and practical results are incredibly promising.

4.3 Aligning encodings

We talked on how CLIP shows a bag-of-word approach and has great difficulties binding attributes and objects. But where does this limitation stems from? Just from the Uni-modal encoders (for captions and image respectively) or also from the cross-modal step that matches the two encodings? A study demonstrates that the encodings actually contain useful binding information, but matching fails to align such info[6]. In fact, using a simple linear probing on the already encoded images and text, the study shows that it is possible to extract such info, but when CLIP is faced with perturbations in the captions, its accuracy is comparable to random chance for binding. As the hypothesis is that

CLIP fails only when aligning the two encodings, the solution proposed with LABCLIP is simply to add a transformation Matrix to one of the two embeddings (like the text one) and freeze CLIP's weight while training the matrix. The use of hard negatives for this training helps further, and the results significantly improve zero-shot performances (while remaining slightly below finetuned CLIP), demonstrating that the information is already encoded, and no extensive computation is required for binding improvements. These results, while optimistic, also underline a problem in CLIP's cross-modal binding, as it is unable to naturally align information between text and image unless specifically trained to do so.

Conclusions

CLIP is an extremely powerful tool, and as the most widespread model for image-text matching its important to fully understand its strengths, limits and possible improvements.

As we saw it still has many shortcomings, and the many solutions and model variants can only partially solve them most of the time.

Many of its problems stem from the dataset's quality, as the massive quantities of image-text pairs are difficult to filter, often carry strong biases, and lack sufficiently varied examples for many case studies. This point is perhaps one of the most pressing, as almost all of the problems cited, like CAB, negation, hallucination, style recognition, parroting, counting, and even the recognition of relationships between objects, can benefit from better quality datasets and hard negatives. As such the direction to further improve CLIP lies not in further increasing its dataset size, but rather improving its quality.

We also noted how some of the most pressing problems such as attribute binding, spatial relationships, and negation all apparently stem from a limit in CLIP's encoding space. As such the solution can not simply be to lightly alter the encoders or training functions, but requires a deeper evolution, adequately considering not only efficacy, but also the tradeoff between performances, training time, and simplicity (which are currently CLIP's strengths).

In the end, the Stronger points of CLIP: its vast web-crawled dataset, and its model simplicity, are not only the attributes that allowed for its success and incredible performances, but may also be what is holding it off from evolving further and overcoming its limitations.

Bibliography

[1] Reza Abbasi, Ali Nazari, Aminreza Sefid, Mohammadali Banayeeanzade, Mohammad Hossein Rohban, and Mahdieh Soleymani Baghshah. Clip under the microscope: A fine-grained analysis of multi-object representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9308–9317, June 2025.

[2] Andrea Asperti, Leonardo Dessi, Maria Chiara Tonetti, and Nico Wu. Does clip perceive art the same way we do? In *2025 International Conference on Content-Based Multimedia Indexing (CBMI)*, pages 1–8, 2025.

[3] Andrea Asperti, Franky George, Tiberio Marras, Razvan Ciprian Stricescu, and Fabio Zanotti. A critical assessment of modern generative models' ability to replicate artistic styles. *Big Data and Cognitive Computing*, 9(9), 2025.

[4] Lorenzo Bianchi, Fabio Carrara, Nicola Messina, and Fabrizio Falchi. Is clip the main roadblock for fine-grained open-world perception? In *2024 International Conference on Content-Based Multimedia Indexing (CBMI)*, pages 1–8, 2024.

[5] Raphi Kang, Yue Song, Georgia Gkioxari, and Pietro Perona. Is clip ideal? no. can we fix it? yes! In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22436–22446, October 2025.

[6] Darina Koishigarina, Arnas Uselis, and Seong Joon Oh. CLIP behaves like a bag-of-words model cross-modally but not uni-modally. In *The Fourteenth International Conference on Learning Representations*, 2026.

[7] Zhengfeng Lai, Haotian Zhang, Bowen Zhang, Wentao Wu, Haoping Bai, Aleksei Timofeev, Xianzhi Du, Zhe Gan, Jiulong Shan, Chen-Nee Chuah, Yinfei Yang, and Meng Cao. Veclip: Improving clip training via visual-enriched captions. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, *Computer Vision – ECCV 2024*, pages 111–127, Cham, 2025. Springer Nature Switzerland.

[8] Martha Lewis, Nihal Nayak, Peilin Yu, Jack Merullo, Qinan Yu, Stephen Bach, and Ellie Pavlick. Does CLIP bind concepts? probing compositionality in large image models. In Yvette Graham and Matthew Purver, editors, *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1487–1500, St. Julian's, Malta, March 2024. Association for Computational Linguistics.

[9] Xianhang Li, Zeyu Wang, and Cihang Xie. An inverse scaling law for clip training. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors,

Advances in Neural Information Processing Systems, volume 36, pages 49068–49087. Curran Associates, Inc., 2023.

[10] Yiqi Lin, Conghui He, Alex Jinpeng Wang, Bin Wang, Weijia Li, and Mike Zheng Shou. Parrot captions teach clip to spot text. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, *Computer Vision – ECCV 2024*, pages 368–385, Cham, 2025. Springer Nature Switzerland.

[11] Yufang Liu, Tao Ji, Changzhi Sun, Yuanbin Wu, and Aimin Zhou. Investigating and mitigating object hallucinations in pretrained vision-language (CLIP) models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18288–18301, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[12] Norman Mu, Alexander Kirillov, David Wagner, and Saining Xie. Slip: Self-supervision meets language-image pre-training. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 529–544, Cham, 2022. Springer Nature Switzerland.

[13] Roni Paiss, Ariel Ephrat, Omer Tov, Shiran Zada, Inbar Mosseri, Michal Irani, and Tali Dekel. Teaching clip to count to ten. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3147–3157, 2023.

[14] Junsung Park, Jungbeom Lee, Jongyoon Song, Sangwon Yu, Dahuin Jung, and Sungroh Yoon. Know "no" better: A data-driven approach for enhancing negation awareness in clip. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2825–2835, October 2025.

[15] Maitreya Patel, Naga Sai Abhiram kusumba, Sheng Cheng, Changhoon Kim, Tejas Gokhale, Chitta Baral, and Yezhou Yang. TripletCLIP: Improving compositional reasoning of CLIP via synthetic vision-language negatives. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[16] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021.

[17] Yingtian Tang, Yutaro Yamada, Yoyo Zhang, and Ilker Yildirim. When are lemons purple? the concept association bias of vision-language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14333–14348, Singapore, December 2023. Association for Computational Linguistics.

[18] Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9568–9578, June 2024.

[19] Hu Xu, Saining Xie, Xiaoqing Tan, Po-Yao Huang, Russell Howes, Vasu Sharma, Shang-Wen Li, Gargi Ghosh, Luke Zettlemoyer, and Christoph Feichtenhofer. Demystifying clip data. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun, editors, *International Conference on Learning Representations*, volume 2024, pages 47812–47831, 2024.

[20] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? In *The Eleventh International Conference on Learning Representations*, 2023.

[21] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11941–11952, 2023.

[22] Beichen Zhang, Pan Zhang, Xiaoyi Dong, Yuhang Zang, and Jiaqi Wang. Long-clip: Unlocking the long-text capability of clip. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part LI*, page 310–325, Berlin, Heidelberg, 2024. Springer-Verlag.

[23] Le Zhang, Rabiul Awal, and Aishwarya Agrawal. Contrasting Intra-Modal and Ranking Cross-Modal Hard Negatives to Enhance Visio-Linguistic Compositional Understanding . In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13774–13784, Los Alamitos, CA, USA, June 2024. IEEE Computer Society.