Bias in CLIP Encoders: A Study of Encoder Bias and Object Representation in Multi-Object Scenarios

Anonymous authors
Paper under double-blind review

Abstract

Contrastive Language-Image Pre-training (CLIP) models have demonstrated remarkable performance in zero-shot classification tasks, yet their efficacy in handling complex multi-object scenarios remains challenging. This study presents a comprehensive analysis of CLIP's performance limitations in multi-object contexts through controlled experiments. We present a specialized dataset, ComCO, crafted to thoroughly assess the performance of CLIP's encoders in diverse multi-object scenarios. Our findings reveal significant biases in both encoders, with the text encoder showing a tendency to prioritize objects that are mentioned first in the prompt, and the image encoder exhibiting a bias toward larger objects. Through meticulous experiments, including both retrieval-based and classification-based tasks, we quantify these biases across multiple CLIP variants, we quantify these biases across multiple CLIP variants. We hypothesize that these biases originate from CLIP's training process and provide substantiating evidence through detailed analyses of the LAION dataset and CLIP's training progression. Our image-text matching experiments demonstrate substantial performance drops when manipulating object sizes in the images and/or object tokens order in the prompt, highlighting the CLIP's unstable performance when given rephrased yet semantically similar captions. We extend this analysis to longer, more complex captions and text-to-image generative models such as Stable Diffusion, revealing how CLIP's text encoder bias influences object prominence in generated images based on the prompt's token order. This work provides crucial insights into CLIP's behavior in complex visual-linguistic contexts, offering a robust evaluation methodology and identifying key areas for improving future vision-language models in multi-object scenarios.

1 Introduction

The convergence of vision and language in artificial intelligence has led to the development of Vision-Language Models (VLMs) that can interpret and generate multimodal content. Among these, OpenAI's Contrastive Language-Image Pre-training (CLIP) model Radford et al. (2021) has been particularly influential, demonstrating remarkable capabilities in zero-shot image classification and setting new standards for multimodal understanding Cherti et al. (2023); Gadre et al. (2023); Schuhmann et al. (2021); Thrush et al. (2022). The success of CLIP has catalyzed a wide array of applications—from image retrieval and visual question answering to text-to-image generation—signifying a paradigm shift in how models perceive and relate visual and linguistic information.

Visual Language Models like CLIP face significant challenges in understanding and reasoning about complex scenes with multiple objects and intricate relationships. CLIP struggles to identify distinct objects and model their relationships accurately, especially when captions contain the same objects but differ in their relationships. This results in difficulty distinguishing between similar captions with different object relationships. Several benchmark datasets have been introduced to elucidate the limitations of existing models in capturing subtle relational nuances. Notably, Winoground Thrush et al. (2022), VL-CheckList Zhao et al. (2022), ARO Yuksekgonul et al. (2023), and CREPE Ma et al. (2023) have been instrumental in evaluating models' capacities to accurately match images with semantically appropriate captions.

Numerous efforts have been made to address compositionality challenges in the multi-object setting. These studies have predominantly employed end-to-end methodologies, including fine-tuning techniques with hard-negative samples Yuksekgonul et al. (2023), to enhance model performance. The efficacy of these approaches have been criticized and improved recently, SUGARCREPE Hsieh et al. (2024), and Sahin et al. (2024). Specifically, a common methodology in these works involves the generation of negative captions through minor structural alterations, or through LLMs, to the original positive ones, emphasizing the identification of semantic disparities between captions that share structural similarities but differ conceptually. Hence, these approaches helped in capturing nuances in the text domain that is necessary in the compositional multi-object scenarios.

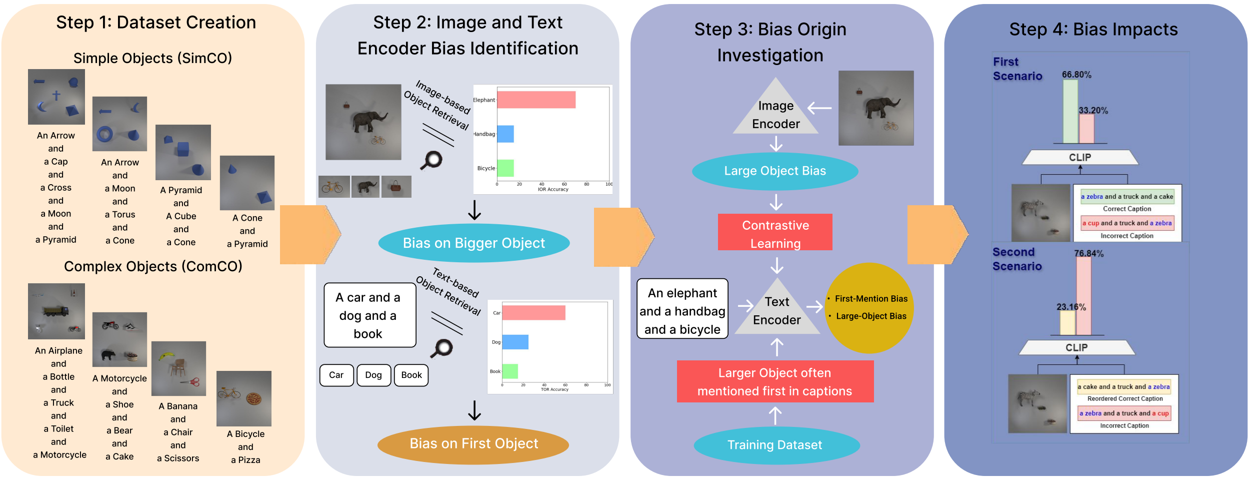

Step 1: Dataset Creation
Simple Objects (SimCO)
An Arrow and a Cap and a Cross and a Moon and a Pyramid
An Arrow and a Moon and a Torus and a Cone
A Pyramid and A Cube and a Cone
A Cone and a Pyramid
Complex Objects (ComCO)
An Airplane and a Bottle and a Truck and a Toilet and a Motorcycle
A Motorcycle and a Shoe and a Bear and a Cake
A Banana and a Chair and a Scissors
A Bicycle and a Pizza
Step 2: Image and Text Encoder Bias Identification
Image-based Object Retrieval
Elephant
Handbag
Bicycle
0
20
40
60
80
100
IOR Accuracy
Bias on Bigger Object
A car and a dog and a book
Text-based Object Retrieval
Car
Dog
Book
TOR Accuracy
Bias on First Object
Step 3: Bias Origin Investigation
Image Encoder
Large Object Bias
Contrastive Learning
An elephant and a handbag and a bicycle
Text Encoder
First-Mention Bias
Large-Object Bias
Larger Object often mentioned first in captions
Training Dataset
Step 4: Bias Impacts
First Scenario
66.80%
33.20%
CLIP
a zebra and a truck and a cake
Correct Caption
a cup and a truck and a zebra
Incorrect Caption
Second Scenario
76.84%
23.16%
CLIP
a cake and a truck and a zebra
Reordered Correct Caption
a zebra and a truck and a cup
Incorrect Caption

Figure 1: Overview of our key contributions. Step 1: We create ComCO dataset for controlled multi-object experiments. Step 2: We identify biases in CLIP's image encoder (favoring larger objects) and text encoder (prioritizing first-mentioned objects). Step 3: We investigate the origin of these biases, finding a connection to training data characteristics. Step 4: We demonstrate the practical impacts of these biases on image-text matching task, showing how they affect model performance in multi-object scenarios.

While these efforts have primarily focused on assessing CLIP's ability to differentiate between captions with minor structural variations but significant conceptual divergences, there remains a paucity of research examining CLIP's performance on captions that are semantically equivalent but structurally distinct. The work of Dumpala et al. Dumpala et al. (2024) represents one of the few forays into this domain. However, while such studies have introduced novel benchmarks, they have not comprehensively explored the underlying mechanisms that contributes to the CLIP unstable performance when given semantically equivalent prompts.

While previous studies have made significant strides in understanding CLIP's limitations, our work distinguishes itself in several key aspects. Firstly, we shift the focus from evaluating CLIP's ability to differentiate between conceptually distinct captions to examining its performance with semantically equivalent but structurally varied captions. This approach allows us to probe deeper into the model's understanding of language and visual content beyond surface-level differences. Here, model systematic mistakes give an indication the potential baises. Secondly, unlike many previous works that primarily introduced benchmarks or proposed end-to-end solutions, we conduct a thorough investigation into the underlying causes of CLIP's behavior. Our study delves into the internal mechanisms of both the image and text encoders, providing insights into why the model is biased and lacks invariance to certain types of linguistic and visual variations.

To facilitate this in-depth analysis, we introduce the **ComCO** dataset, specifically designed to isolate and examine different aspects of CLIP's performance in *controlled* multi-object scenarios. Furthermore, our research spans multiple versions of CLIP trained on various datasets and architectures, ensuring the broad applicability and generalizability of our findings. By focusing on these underexplored areas and employing a more comprehensive analytical approach, our work aims to provide a deeper understanding of CLIP's limitations and pave the way for more robust and versatile vision-language models. It is important to note that such an analysis not only benefits the improvement of CLIP but also has significant implications for related models, such as text-to-image (T2I) generative models and multimodal large language models (MLLMs). Understanding the intricacies of CLIP's encoding process can inform and enhance the development of these technologies, potentially leading to advancements across various domains of artificial intelligence. As shown in Figure 1, our key contributions are as follows:

- **Development of Novel Dataset**: We introduce *ComCO*, a specialized dataset specifically designed to create *controlled* multi-object scenarios. Here, unlike previous benchmarks, we can control the object size in the image, and their ordering in the caption. Hence, this dataset enables precise, fine-grained analysis of model performance across a spectrum of compositional challenges, facilitating a deeper understanding of VLMs' strengths and weaknesses.
- **Comprehensive Encoder Analysis**: We perform an in-depth examination of both the image and text encoders in CLIP when processing multi-object scenes and descriptions. This includes text-based, and object-based

image retrievals, that reveal each text and image encoder weaknesses in preserving the information necessary to discern various objects. By analyzing the embedding space, we identify the stages at which compositional information is lost or distorted, providing insights into the internal mechanisms of the model.

- **Identification of Specific Biases**: Our research uncovers significant biases in CLIP models. The image encoder prefers larger objects in multi-object images, while the text encoder favors first-mentioned objects and also objects that are usually visually larger in real-world. These biases reveal the complex interplay between visual and linguistic information processing in CLIP, influencing its interpretation of multi-object scenarios.
- **Investigation of the Bias Origin** : We explore the origins of observed biases in CLIP's performance, particularly in various multi-object scenarios. Our investigation delves into both the image and text encoders. We hypothesize that the visually larger objects are mostly mentioned earlier in the caption in CLIP training datasets. But it is evident that the image encoding naturally favors such objects in the embedding due to the abundance of their visual tokens. Therefore, the text encoder may get biased towards such objects, and consequently earlier mentioned text tokens. We provide evidence for these biases through analyses of the LAION dataset and CLIP's training progression, revealing a consistent trend where larger objects tend to be mentioned earlier in image captions.
- **Practical impacts of encoder biases**: We demonstrate how the identified biases in CLIP's image and text encoders significantly impact performance in multi-object analysis/synthesis scenarios. Using our ComCO dataset, we show substantial drops in image-text matching accuracy when manipulating object sizes and caption order. We further reveal how these biases propagate to text-to-image generation models like Stable Diffusion, influencing the prominence and likelihood of object appearance in generated images based on prompt order.

These observations highlight how biases in both the text and image encoders lead to a substantial decrease in CLIP's performance in multi-object scenarios. Our findings underscore the importance of addressing these biases to improve the robustness and versatility of vision-language models in complex visual environments. This work contributes valuable insights into CLIP's behavior in multi-object contexts and opens up new avenues for enhancing the performance of vision-language models in real-world applications.

2 Methodology

2.1 Dataset Design

To thoroughly evaluate the performance of CLIP models in multi-object scenarios under controlled conditions, we constructed the **ComCO** (Complex COCO Objects) dataset. Utilizing Blender software allowed us precise control over the number, location, and dimensions of objects in the images (see Appendix A.1). The **ComCO** dataset comprises 72 objects derived from the COCO dataset Lin et al. (2015). We generated images containing 2, 3, 4, and 5 objects. Each image is paired with a specific caption that accurately describes the objects present. This approach ensures high control over the dataset and minimizes confounding factors, providing a robust platform for evaluating the CLIP models.

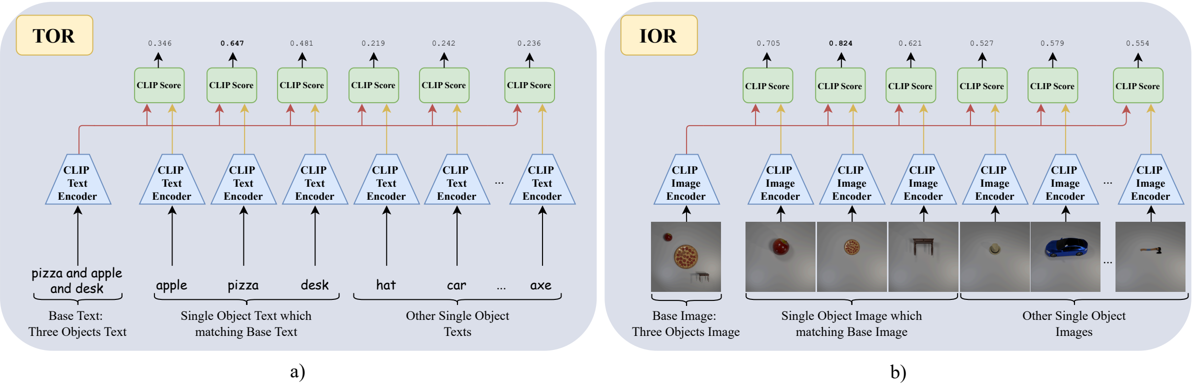
We deliberately chose not to use text-to-image models for generating these datasets due to two main reasons. First, these models often lack the capability to produce high-quality, fully controlled multi-object images. Second, since CLIP is used in many of these models, utilizing them could introduce unwanted biases into our evaluations.

2.2 Experimental Framework for Encoder Analysis

The main goal of this study is to evaluate the performance of CLIP's text and image encoders separately in multi-object scenarios. We aim to analyze the impact and contribution of each object in the final output of the encoders. To achieve this, we conducted experiments using our designed ComCO dataset, with images and captions containing two to five objects. To ensure the generalizability of our findings, we also validated our results on the widely-used COCO dataset Lin et al. (2014). We designed two sets of experiments: retrieval-based experiments and classification-based experiments. Given the consistency of the results in both types of experiments, we have included the classification results in the appendix A.2 and A.5 and explain the retrieval-based experiments bellow.

2.2.1 TEXT-BASED OBJECT RETRIEVAL (TOR)

The Text-based Object Retrieval task evaluates how well CLIP's text encoder can identify individual objects within multi-object captions. As illustrated in Figure 2a, this experiment involves several steps: First, we use CLIP's text


TOR
0.346
0.647
0.481
0.219
0.242
0.236
CLIP Score
CLIP Text Encoder
...
pizza and apple and desk
apple
pizza
desk
hat
car
...
axe
Base Text: Three Objects Text
Single Object Text which matching Base Text
Other Single Object Texts
IOR
0.705
0.824
0.621
0.527
0.579
0.554
CLIP Score
CLIP Image Encoder
...
Base Image: Three Objects Image
Single Object Image which matching Base Image
Other Single Object Images

a)

b)

Figure 2: Experimental setup for Text-based Object Retrieval (TOR) and Image-based Object Retrieval (IOR) tasks. a) TOR: The CLIP text encoder generates embeddings for multi-object and single-object texts. Cosine similarity scores are calculated between the base text embedding and single-object text embeddings to identify the most similar object. b) IOR: The CLIP image encoder generates embeddings for multi-object and single-object images. Cosine similarity scores are calculated between the base image embedding and single-object image embeddings to identify the most similar object.

encoder to create embeddings for both multi-object captions and single-object captions. We then measure the similarity between each multi-object caption embedding and all single-object caption embeddings. The single-object caption with the highest similarity score is considered the ”retrieved” object. To assess performance, we calculate retrieval accuracy for each object position in the multi-object captions. This helps us identify any biases related to an object’s position within a caption, such as favoring objects mentioned first or last.

2.2.2 IMAGE-BASED OBJECT RETRIEVAL (IOR)

The Image-based Object Retrieval task is similar to TOR but focuses on CLIP’s image encoder. As shown in Figure 2b, this experiment involves several steps: We begin by using CLIP’s image encoder to generate embeddings for multi-object images and single-object images. We then compute similarity scores between each multi-object image embedding and all single-object image embeddings. The single-object image with the highest similarity score is considered the ”retrieved” object. To evaluate performance, we calculate retrieval accuracy for different object size categories (e.g., large, small) within the multi-object images. This allows us to determine if the image encoder shows any preference for objects of a particular size.

We also experimented with a variation of ComCO, called SimCO, where objects were replaced with simple geometric shapes from the CLEVR dataset. This was done to confirm that bias persists even with non-natural, geometric objects. Further details are provided in Appendix A.1.

3 RESULTS AND ANALYSIS

Our experiments revealed significant biases in both the text and image encoders of the CLIP model. This section presents our findings, organized by encoder type and focusing on retrieval tasks.

3.1 TEXT ENCODER BIASES

We observed a consistent bias in the text encoder towards the first object mentioned in descriptions. In the TOR experiment, the retrieval accuracy (as shown in Table 1) was highest for the first object, indicating its dominant influence on the overall text representation. This suggests that the text encoder prioritizes the initial object, leading to its more accurate retrieval compared to subsequent objects. The detailed results for the scenarios involving 2, 3, and 5 objects can be found in the appendix A.3, and experiments on longer caption templates are in Appendix A.7 and A.8.

3.2 IMAGE ENCODER BIASES

In multi-object images, the image encoder exhibited a strong bias towards larger objects. The Image-based Object Retrieval IOR experiment, detailed in Table 2, shows that larger objects were more frequently and accurately retrieved during single-object image searches. This finding highlights the image encoder's bias towards larger objects, which receive disproportionate emphasis in the final image representation. Further detailed results, specifically for scenarios with 2, 3, and 5 objects, are provided in the appendix A.6.

Table 1: Performance on TOR for ComCO datasets

Task	Model	First Obj	Second Obj	Third Obj	Fourth Obj
TOR	*CLIP LAION*	**63.96**	21.59	10.68	3.76
	CLIP Datacomp	**71.13**	16.26	8.74	3.87
	CLIP Roberta	**44.03**	23.73	18.07	14.18
	SIGLIP	**58.11**	21.16	10.99	9.73
	CLIP openAI	**50.31**	20.74	14.45	6.79
	NegCLIP	**51.63**	28.92	14.86	4.59
	SugarCrepe	**44.29**	30.32	18.73	6.66

Table 2: Performance on IOR for ComCO datasets

Task	Model	Large Object	Small Obj 1	Small Obj 2	Small Obj 3
IOR	*CLIP LAION*	**85.45**	6.36	5.45	2.73
	CLIP Datacomp	**85.16**	5.65	4.95	4.24
	CLIP Roberta	**87.40**	8.66	2.36	1.57
	SIGLIP	**77.66**	10.11	6.38	5.85
	CLIP openAI	**65.22**	17.39	8.70	8.70
	NegCLIP	**61.67**	15.00	13.33	10.00
	SugarCrepe	**60.0**	18.38	16.85	4.7

3.3 COCO Dataset Experiments

To validate the generalizability of our findings from the synthetic dataset, we conducted similar experiments on the COCO dataset, which comprises real images with accompanying captions. This real-world dataset allowed us to investigate whether the previously observed biases persist in more naturalistic settings.

Due to the absence of single-object images for COCO objects, we approached the IOR experiment in two ways. First, we used single-object images from the DomainNet dataset Peng et al. (2019) as retrieval targets. Second, we introduced an alternative approach called Image-to-Text Object Retrieval (I2TOR). In I2TOR, we used the textual names of COCO objects instead of single-object images. These object names were embedded using CLIP's text encoder, allowing us to perform a retrieval task consistent with the IOR methodology while adapting to the constraints of the COCO dataset.

Table 3: Performance on TOR for coco dataset

Task	Model	First Obj	Second Obj	Third Obj	Fourth Obj
TOR	*CLIP openAI*	**35.24**	21.90	20.48	22.38
	CLIP LAION	**67.89**	13.76	8.26	10.09
	CLIP Datacomp	**57.68**	17.68	12.75	11.88
	CLIP Roberta	**40.78**	23.30	20.39	15.53
	SIGLIP	**49.47**	26.84	12.11	11.58
	NegCLIP	**38.69**	22.11	17.09	22.11

Table 4: Performance on IOR for coco dataset

Task	Model	Large Object	Small Obj 1	Small Obj 2	Small Obj 3
IOR	*CLIP openAI*	**43.02**	28.82	17.13	11.03
	CLIP LAION	**39.44**	28.45	17.70	14.41
	CLIP Datacomp	**36.71**	29.55	19.13	14.61
	CLIP Roberta	**36.71**	28.61	19.82	14.86
	SIGLIP	**36.63**	28.29	20.02	15.06
	NegCLIP	**44.04**	28.86	16.48	10.62
I2TOR	*CLIP openAI*	**51.49**	24.87	13.68	9.97
	CLIP LAION	**45.50**	27.02	15.91	11.56
	CLIP Datacomp	**46.64**	26.82	14.53	12.01
	CLIP Roberta	**44.69**	26.98	16.04	12.29
	SIGLIP	**47.09**	27.07	15.10	10.74
	NegCLIP	**49.04**	27.07	14.08	9.81

Tables 3 and 4 present the results of our COCO dataset experiments. In TOR, the first-mentioned object in COCO captions was retrieved with higher accuracy, which aligns with our earlier findings of bias in the text encoder. Similarly, in IOR, larger objects in COCO images were retrieved more accurately, consistent with the trends observed in our synthetic dataset experiments. The I2TOR results further confirmed this bias, demonstrating that even when using textual object representations, the bias towards larger objects persists.

Our experiments reveal two significant biases in the CLIP model: the text encoder shows a strong preference for the first mentioned object in textual descriptions, while the image encoder exhibits greater sensitivity to larger objects in images. These biases can significantly impact the overall system performance in various vision-language tasks, particularly in multi-object scenarios.

4 Origin of Bias in CLIP Models

In this section, we investigate the potential origins of the biases observed in CLIP models and provide evidence supporting our hypotheses.

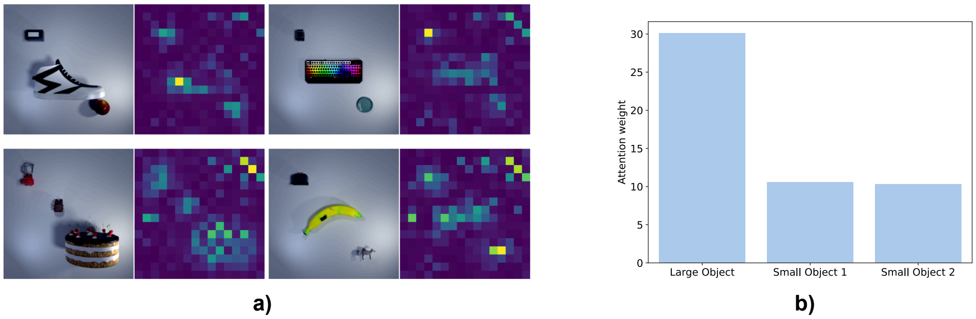


Figure 3: Attention allocation from the CLS token to objects of different sizes in the ComCO dataset. a) Qualitative results showing the CLS token's attention to each object. b) Quantitative analysis of attention distribution across 8,000 images, with each image containing one large and two small objects. The bar chart shows the average attention allocated to the large object versus the smaller ones, demonstrating a bias towards larger objects.

Table 5: Performance on TOC and TOR for ComCO datasets

Task	Model	First Obj	Second Obj	Third Obj	Fourth Obj
TOR	*CLIP*	**56.28**	22.71	13.17	7.48
	SBERT	29.02	19.80	17.50	**33.57**
	SimCSE Gao et al. (2021)	27.59	19.07	17.76	**34.83**

4.1 Bias in the Image Encoder

The observed bias favoring larger objects within the image domain can be attributed to the architectural characteristics of Vision Transformers (ViT) Alexey (2020) utilized in CLIP's image encoder. Our hypothesis is that larger objects, which occupy a greater number of patches in the ViT's patch-based image representation, exert a more significant influence on the final class (CLS) token representation. This bias is not exclusive to CLIP; it appears to be a consistent feature across ViT models, as demonstrated by our experiments detailed in the appendix.

To substantiate this hypothesis, we designed an experiment to quantify the attention allocated by the CLS token to each image patch. By calculating the cumulative attention received by each object from the CLS token, we could assess the influence of object size on attention allocation. We applied this analysis to our three-object ComCO dataset, and the results are illustrated in Figure 3. The findings confirm our hypothesis: larger objects indeed receive more attention from the CLS token.

4.2 Bias in the Text Encoder

We explore the bias present in the text encoder from two perspectives: the attention mechanism in the model structure and the model's training method.

4.2.1 Impact of Attention Mechanism

Text encoder models can be categorized based on their attention mechanisms: uni-directional (causal) attention and bi-directional attention. In models with causal attention, each token attends only to preceding tokens, whereas in bi-directional models, each token attends to all tokens in the sequence.

When OpenAI introduced the CLIP model, its text encoder employed causal attention, meaning each token could only attend to tokens before it and itself. This differs from typical self-attention mechanisms, where tokens attend to all other tokens. Most CLIP models use causal self-attention, with the exception of the variant using the XLM-Roberta text encoder, which also employs self-attention. However, as shown in Table 1, even this model exhibits the mentioned bias. This indicates that the bias does not originate from the attention mechanism itself.

4.2.2 Role of Training Method

To determine whether the observed bias is specific to CLIP models, we compared CLIP's text encoder with two other models designed to embed sentences into a meaningful semantic space: Sentence-BERT (SBERT) Reimers (2019) and SimCSE Gao et al. (2021). The primary distinction is that CLIP's embedding space is shared between images and text, whereas SBERT and SimCSE operate solely in the text domain.

We conducted the TOR experiment on our dataset using these models. As presented in Table 5, the bias observed in CLIP differs from that in the other models. This suggests that CLIP's unique training method, which aligns images and text in a shared embedding space through contrastive learning, contributes to the bias. Therefore, to uncover the root cause of the bias, we focus on the specifics of CLIP's training procedure.

4.3 Hypothesized Origin of Text-Side Bias in CLIP

We hypothesize that the text-side bias in CLIP, which favors objects mentioned earlier in text descriptions, originates from the image-side bias toward larger objects and is transferred to the text encoder during contrastive training. We present evidence supporting this hypothesis through two key claims and an analysis of the training progression.

Claim 1: Larger Objects Have More Influence on Text Embeddings. Building upon the established image-side bias discussed earlier, we posit that objects with larger physical sizes exert more influence on CLIP's text embeddings due to the alignment enforced during contrastive training. To test this, we categorized objects in the DomainNet dataset into large, medium, and small groups based on their relative physical sizes in real-world (with the full list of objects provided in the appendix A.11). Specifically, objects smaller than a school bag were categorized as small, objects sized between a school bag and a medium-sized car were classified as medium, and objects larger than a car—up to significantly larger items—were considered large. We then constructed two sets of sentences, each containing four objects: one set with a large object mentioned first followed by three medium-sized objects, and another with a small object mentioned first followed by three medium-sized objects.

Figure 4.a compares the TOR accuracy for the first object in these two groups. The higher TOR accuracy for sentences beginning with large objects supports our hypothesis that larger objects, when mentioned first, have a more significant impact on the text embeddings due to the cross-modal alignment with their prominent representation in images.

Claim 2: Caption Bias in Training Datasets. To investigate potential biases in CLIP's training data, we analyzed both the LAION Schuhmann et al. (2022) and COCO datasets. Due to limited computational resources and the large size of the LAION dataset, which contains over 2 billion image-text pairs, we randomly selected a subset of 200,000 samples for our analysis. Using the Llama3 model, we extracted objects from the image captions and employed the Language Segment-Anything tool to generate object masks in the corresponding images, calculating their areas based on these masks. A detailed description of our LAION dataset analysis methodology can be found in Appendix A.9.

Figure4.b shows the position of the largest object within each caption. The results indicate that, in the majority of cases, the largest object in an image is mentioned earlier in its caption. The same experiment was conducted on the COCO dataset, with detailed results and the distribution for two to five object scenarios provided in Appendix A.10. This demonstrates a consistent bias in the training data, where larger objects are not only more visually prominent but are also described earlier in text annotations.

Analysis of Bias Development During Training. To further validate our hypothesis, we examined the progression of text-side bias during CLIP's training. We utilized model checkpoints from the LAION dataset at five training stages, corresponding to exposure to 2, 4, 6, 8, and 10 billion samples. We conducted TOR experiments at each stage, focusing on the retrieval accuracy for the first object mentioned in text descriptions.

Figure4.c depicts the evolution of the TOR rate across different training stages for scenarios with varying numbers of objects (from 3 to 8). The consistent upward trend in the TOR rate as the model is exposed to more training data suggests that the text-side bias strengthens over time, likely due to the cumulative effect of the image-side bias being transferred to the text encoder through contrastive learning.

Incomplete Text Representation of CLIP Here we want to theoretically highlight why the CLIP text encoder could learn an incomplete representation of the text. Let $\mathbf{z}$ and $\mathbf{w}$ represent a latent representation of an image content and style, respectively. For example, $\mathbf{z}$ represents the fact that an image contains "a horse that is eating the grass." In this case, $\mathbf{w}$ might represent other details in the image, like the "horse color," "where the horse is located," etc. We assume a data generative process as follows:

$$I := g(\mathbf{z}, \mathbf{w})$$
$$T := h(\mathbf{z}),$$

where I is the image, and T is its corresponding caption.

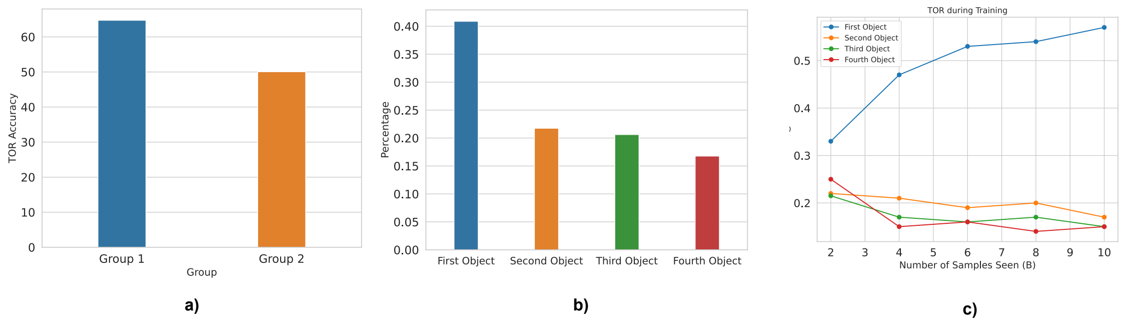


Figure 4: a) Top-1 Object Retrieval accuracy comparison for sentences where the first object is either large or small. The higher TOR accuracy for sentences beginning with large objects supports the hypothesis that larger objects, when mentioned first, exert a stronger influence on text embeddings due to cross-modal alignment with their prominent visual representation in images. b) Distribution of the position of the largest object within image captions from the LAION datasets. The results show a consistent bias where larger objects tend to be mentioned earlier in text descriptions. c) Progression of TOR rates across different training stages, indicating that text-side bias strengthens as the model is exposed to more data, suggesting the cumulative effect of image-side bias being transferred to the text encoder through contrastive learning.

Now we want to learn a joint embedding of the image and text through the CLIP. Here, we assume that f_θ and i_ω as learnable functions that map the image and text into the joint embedding space, respectively.

Theorem 1 *Let elements of z be independent and zero-mean. The contrastive loss for the ideal text encoder, $i_\omega = \mathbf{z}$ converges to that of a non-ideal incomplete one, i.e. $i_\omega = \mathbf{z}_s$, where $\mathbf{z}_s$ is the first $d-k$ dimensions of $\mathbf{z}$, with k being a constant, and $d \to \infty$.*

Proof: The contrastive loss in making this learning happen can be written as:

$$\mathbb{E}_{\mathbf{z},\mathbf{z}',\mathbf{w}} \left\{ \frac{\exp\left\{S(f_\theta(g(\mathbf{z},\mathbf{w}), i_\omega(h(\mathbf{z}))\right\}}{\exp\left\{S(f_\theta(g(\mathbf{z},\mathbf{w}), i_\omega(h(\mathbf{z}))\right\} + \exp\left\{S(f_\theta(g(\mathbf{z},\mathbf{w}), i_\omega(h(\mathbf{z}'))\right\}} \right\}, \tag{1}$$

were $\mathbf{z}$ and $\mathbf{z}'$ are two independent samples of the content in the representation space, and S is some normalized similarity metric, e.g. cosine similarity. We assume that elements of $\mathbf{z}$, denoted as z_i's are independent and zero mean. We further assume that the dimensionality of $\mathbf{z}$, denoted as d, goes to infinity.

It is well-known that under such conditions, $\|\mathbf{z}\| \xrightarrow{p} \sqrt{d}$, when d is large. Therefore, for two independent copies of $\mathbf{z}$, $\mathbf{z}'$, we have $S(\mathbf{z},\mathbf{z}') = \mathbf{z}^\top \mathbf{z}'/(\|\mathbf{z}\|\|\mathbf{z}'\|) \xrightarrow{p} 0$.

It is evident that in the ideal case, $f_\theta(g(\mathbf{z},\mathbf{w})) = \mathbf{z}$ and also $i_\omega(h(\mathbf{z})) = \mathbf{z}$, so the contrastive loss would converge to $e/(e+1)$, as the numerator is e, and the second term in the denominator converges to $\exp(0) = 1$, according to the Mann-Wald's theorem.

However, we show that other learning of this representation could achieve the same amount of loss. For instance, let $\mathbf{z}_s$ be the first $d-k$ elements of $\mathbf{z}$, with k being a *constant*. We show that if $f = \mathbf{z}_s$ and $i = \mathbf{z}_s$, the same loss would be achieved in the limit of large d. To see this, note that the numerator stays the same, i.e. e, while the second term in the denominator still converges to $\exp(0) = 1$.

This means that even if the image and text encoder of the CLIP only partially recover the content embedding, they reach an excellent loss. But such possible incomplete representations of $\mathbf{z}$ are combinatorially large, making convergence of the CLIP to such local minima pretty likely. This makes the text encoding of CLIP be far from ideal. Furthermore, the text encoder would become *biased*, depending on which of such local minima it converges to. Based on this explanation, we would expect a text encoder that has learned a complete representation to exhibit such biases to a lesser degree. As mentioned earlier, the subject of learning text representations in VLMs that are discriminative of hard negatives (e.g. NegCLIP) has been around for few years. We tested one of strongest such models, Hsieh et al. (2024), in our benchmark to validate the hypothesis that an incomplete text representation is one of the causes of the bias in the VLMs. We noticed that this model shows lower bias based on our benchmark (see the SugarCrepe model in tables 1 and 2).

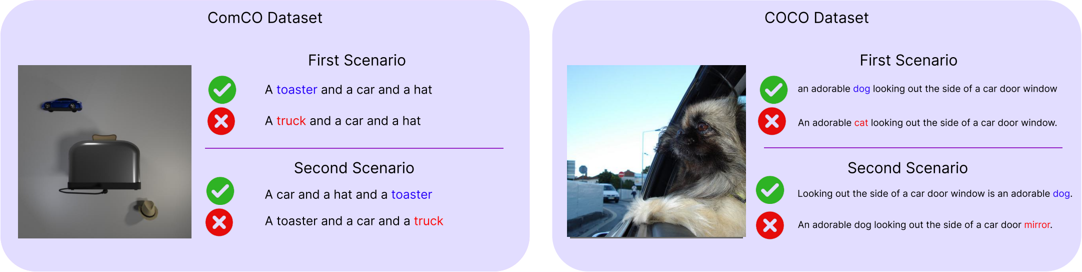

ComCO Dataset
First Scenario
✓ A toaster and a car and a hat
✗ A truck and a car and a hat
Second Scenario
✓ A car and a hat and a toaster
✗ A toaster and a car and a truck
COCO Dataset
First Scenario
✓ an adorable dog looking out the side of a car door window
✗ An adorable cat looking out the side of a car door window.
Second Scenario
✓ Looking out the side of a car door window is an adorable dog.
✗ An adorable dog looking out the side of a car door mirror.

Figure 5: An example of the correct and incorrect caption structures in the first and second scenarios.

5 Practical Impacts of Encoder Biases

The biases we observed in CLIP's image and text encoders have significant implications for the model's performance in real-world applications. This section explores how these biases manifest in practical scenarios, focusing on two key areas: image-text matching and text-to-image generation. By examining these applications, we aim to demonstrate the tangible effects of encoder biases on CLIP's functionality and highlight the importance of addressing these issues for improved model performance.

Our analysis in this section serves two primary purposes. First, it provides concrete evidence of how these theoretical biases can translate into practical limitations. Second, it offers insights into potential areas for improvement in vision-language models, particularly in handling complex, multi-object scenarios. Through a series of carefully designed experiments, we illustrate how the biases in both text and image encoders can lead to unexpected or suboptimal results in tasks that are crucial for many downstream applications.

5.1 Image-Text Matching

Building upon our findings of biases in CLIP's image and text encoders, we now demonstrate how these biases tangibly affect the model's performance in image-caption matching tasks. We designed two experimental scenarios, conducted on both the ComCO and COCO datasets, to evaluate these biases. The results of these experiments are summarized in Table 6. To better illustrate the differences between these two scenarios, an example of the caption structures is shown in Figure 5. In each scenario, we created incorrect captions by switching one object in the caption with an object that is not present in the image. Additionally, GPT-4O Achiam et al. (2023) was used to rewrite the captions in the COCO dataset.

First Scenario In the first scenario, biases assist the model in distinguishing between the correct and incorrect captions. In the correct captions, the largest object in the image is placed at the beginning, aligning with the model's bias towards prioritizing first-mentioned objects and larger objects. For the incorrect captions, the non-existent object is deliberately placed at the beginning, which helps the model recognize the difference between the correct and incorrect captions more effectively. This positioning emphasizes the discrepancy early on, allowing the model to better detect the mismatch between the caption and the image. The performance of different models in this scenario can be seen in Table 6 under the ”First Scenario” column.

Second Scenario In the second scenario, biases lead the model to make errors. The correct captions place the largest object at the end of the sentence, disrupting the model's bias towards objects mentioned earlier and its preference for larger objects. In the incorrect captions, the non-existent object is placed at the end, making it more difficult for the model to differentiate between correct and incorrect captions as its attention is drawn away from the critical discrepancies. The performance of different models in this scenario is shown in Table 6 under the ”Second Scenario” column.

By comparing these two scenarios, we demonstrate that biases in CLIP can either help or hinder the model's performance depending on how captions are structured. The experimental results, particularly with the use of GPT-4O for caption rephrasing in the COCO dataset, reveal how such biases can influence the accuracy of image-text matching tasks. These biases must be addressed to improve CLIP's robustness in real-world multi-object scenarios.

Table 6: Performance Comparison on Image-Text matching for ComCO and COCO Datasets

Model	**ComCO**		**COCO**	
	First Scenario	**Second Scenario**	**First Scenario**	**Second Scenario**
CLIP Datacomp Gadre et al. (2024)	**99.99**	67.50	71.2	54.2
CLIP Roberta	**99.98**	64.75	72.2	54.1
SIGLIP Zhai et al. (2023)	**99.49**	72.36	64.8	39.5
CLIP openAI	**99.59**	52.23	63.5	26.4
NegCLIP	**96.82**	46.94	72	28.7
SugarCrepe	**98.55**	60.43	80.0	40.9

5.2 Text to image geneation

The biases observed in CLIP's encoders have significant implications beyond image-text matching, particularly for text-to-image generation models that incorporate CLIP components. To investigate this impact, we focused on Stable Diffusion, a popular text-to-image generation model that utilizes CLIP's text encoder in its pipeline. Stable Diffusion employs CLIP's text encoder to process input prompts, creating text embeddings that guide the image generation process. Given our identification of biases in CLIP's text encoder, especially the preference for objects mentioned earlier in text descriptions, we hypothesized that these biases would manifest in the generated images. To test this hypothesis, we designed an experiment using prompts containing multiple objects from the COCO dataset. Our goal was to observe whether the order of objects in the text prompt influences their prominence or likelihood of appearance in the generated images.

Our experimental methodology consisted of three main steps. First, we created 1,000 multi-object prompts, each containing four distinct objects from the COCO dataset. Second, we used these prompts to generate images using three versions of Stable Diffusion: v1.4 Rombach et al. (2022), v2, and SD-XL Podell et al. (2023). Finally, to evaluate the presence of objects in the generated images, we employed YOLO v8 Reis et al. (2023), a state-of-the-art object detection model. We configured YOLO v8 with a detection threshold of 0.25 and used it to validate which objects from the original prompt were present in the generated image.

This approach allowed us to quantitatively assess how CLIP's text encoder biases propagate through the Stable Diffusion pipeline and manifest in the generated images. By comparing the frequency of object detection with their position in the input prompt, we could directly observe the impact of the text-side bias on the image generation process.

Table 7: Object presence in Stable Diffusion-generated images

Model	**First Obj**	**Second Obj**	**Third Obj**	**Fourth Obj**
SD v1.4	57.7	44.7	38.1	35.4
SD V2	62.5	49.7	47.5	42.2
SD-XL	79.2	69.3	59.4	64.0

Our findings, presented in Table 7, demonstrate a clear correlation between an object's position in the text prompt and its likelihood of appearing in the generated image. This correlation aligns with our earlier observations of CLIP's text encoder bias, suggesting that these biases significantly influence the output of text-to-image generation models.

6 Conclusion

Our study reveals significant biases in CLIP's image and text encoders, favoring larger objects and first-mentioned items respectively. These biases, demonstrated through our ComCO dataset, substantially impact CLIP's performance in multi-object scenarios. The observed performance drops when manipulating object sizes and mention order underscore CLIP's limitations in handling complex visual environments. These findings highlight the need for more balanced training approaches in vision-language models to mitigate such biases. Future work should focus on developing techniques to address these limitations, advancing the field towards more robust and versatile AI systems capable of accurately interpreting multi-faceted real-world information.

References

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

Dosovitskiy Alexey. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv: 2010.11929*, 2020.

Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2023. doi: 10.1109/cvpr52729.2023.00276. URL `http://dx.doi.org/10.1109/CVPR52729.2023.00276`.

Sri Harsha Dumpala, Aman Jaiswal, Chandramouli Sastry, Evangelos Milios, Sageev Oore, and Hassan Sajjad. Sugarcrepe++ dataset: Vision-language model sensitivity to semantic and lexical alterations. *arXiv preprint arXiv:2406.11171*, 2024.

Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, Eyal Orgad, Rahim Entezari, Giannis Daras, Sarah Pratt, Vivek Ramanujan, Yonatan Bitton, Kalyani Marathe, Stephen Mussmann, Richard Vencu, Mehdi Cherti, Ranjay Krishna, Pang Wei Koh, Olga Saukh, Alexander Ratner, Shuran Song, Hannaneh Hajishirzi, Ali Farhadi, Romain Beaumont, Sewoong Oh, Alex Dimakis, Jenia Jitsev, Yair Carmon, Vaishaal Shankar, and Ludwig Schmidt. Datacomp: In search of the next generation of multimodal datasets, 2023. URL `https://arxiv.org/abs/2304.14108`.

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

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

Cheng-Yu Hsieh, Jieyu Zhang, Zixian Ma, Aniruddha Kembhavi, and Ranjay Krishna. Sugarcrepe: Fixing hackable benchmarks for vision-language compositionality. *Advances in neural information processing systems*, 36, 2024.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755. Springer, 2014.

Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015. URL `https://arxiv.org/abs/1405.0312`.

Zixian Ma, Jerry Hong, Mustafa Omer Gul, Mona Gandhi, Irena Gao, and Ranjay Krishna. Crepe: Can vision-language foundation models reason compositionally? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10910–10921, 2023.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1406–1415, 2019.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

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, 2021. URL `https://arxiv.org/abs/2103.00020`.

N Reimers. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

Dillon Reis, Jordan Kupec, Jacqueline Hong, and Ahmad Daoudi. Real-time flying object detection with yolov8. *arXiv preprint arXiv:2305.09972*, 2023.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, June 2022.

Ugur Sahin, Hang Li, Qadeer Khan, Daniel Cremers, and Volker Tresp. Enhancing multimodal compositional reasoning of visual language models with generative negative mining. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 5563–5573, 2024.

Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs, 2021. URL `https://arxiv.org/abs/2111.02114`.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022.

Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. Winoground: Probing vision and language models for visio-linguistic compositionality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5238–5248, 2022.

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.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11975–11986, 2023.

Tiancheng Zhao, Tianqi Zhang, Mingwei Zhu, Haozhan Shen, Kyusong Lee, Xiaopeng Lu, and Jianwei Yin. Vl-checklist: Evaluating pre-trained vision-language models with objects, attributes and relations. *arXiv preprint arXiv:2207.00221*, 2022.

A APPENDIX

A.1 THE SIMCO AND COMCO DATASETS

A.1.1 THE SIMCO DATASET

The SIMCO dataset comprises 17 objects. These 17 objects are:

Cube
Mug
Cone
Moon
Leaf
Torus
Sphere
Pentagon
Pyramid
Cross
Arrow
Pot
Cylinder
Heart
Diamond
Snowflake
Star

Using Blender software, a collection of images containing 2 to 5 objects has been created from these 17 objects. The total number of images in this dataset is approximately 85,000. Examples of these images can be seen in Figure 6.

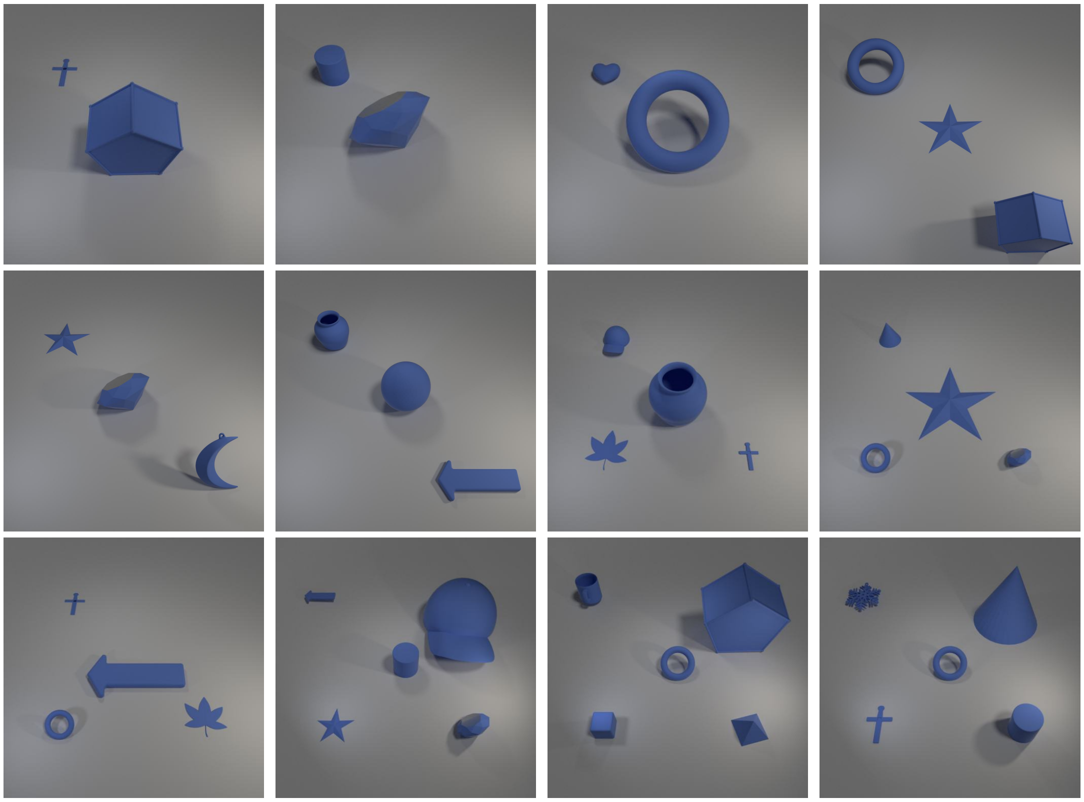


Figure 6: Examples of the SimCO dataset

A.1.2 The ComCO Dataset

The ComCO dataset contains 72 objects, as listed below:

person	bicycle	car	motorcycle	airplane	bus
train	truck	boat	traffic light	fire hydrant	street sign
stop sign	parking meter	bench	bird	cat	dog
horse	sheep	cow	dining table	cell phone	elephant
bear	zebra	giraffe	hat	backpack	umbrella
shoe	eye glasses	handbag	tie	suitcase	frisbee
skis	snowboard	kite	baseball bat	baseball glove	tennis racket
wine glass	hot dog	potted plant	teddy bear	hair drier	hair brush
skateboard	surfboard	bottle	plate	cup	fork
knife	spoon	bowl	banana	apple	sandwich
orange	broccoli	carrot	pizza	donut	cake
chair	couch	bed	mirror	window	desk
toilet	door	tv	laptop	mouse	remote
keyboard	microwave	oven	toaster	sink	refrigerator
blender	book	clock	vase	scissors	toothbrush

In this dataset, a collection of images containing 2 to 5 different objects has also been generated. The total number of images in this dataset is approximately 190,000. Various examples from this dataset can be seen in Figure 12.

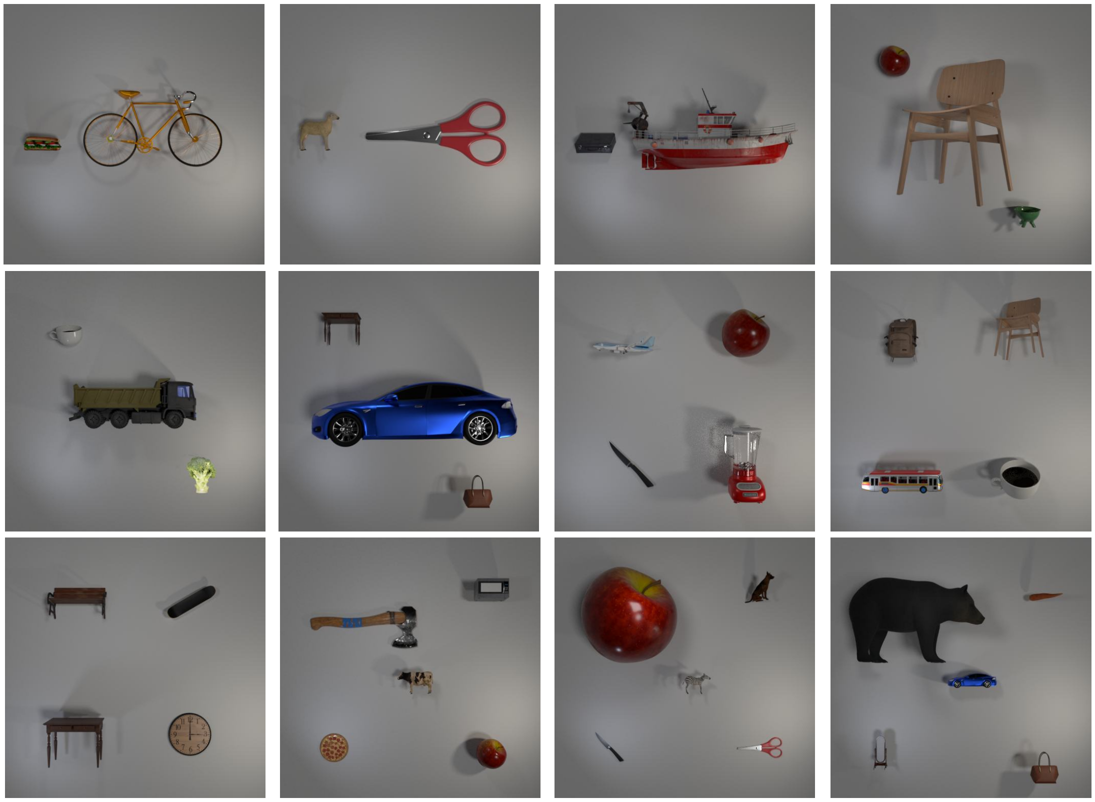


Figure 7: Examples of the ComCO dataset

A.2 Text-based Object Classification

A.2.1 Objective

The Text-based Object Classification experiment was designed to evaluate CLIP's text encoder's ability to represent individual objects within multi-object captions. Our goal was to quantify any potential bias in the representation of objects based on their position in the text.

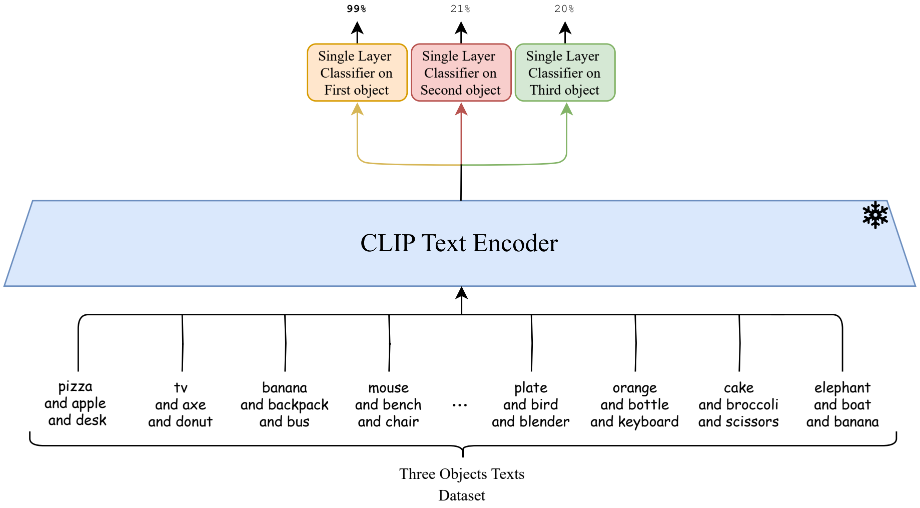

99%
21%
20%
Single Layer Classifier on First object
Single Layer Classifier on Second object
Single Layer Classifier on Third object
CLIP Text Encoder
pizza and apple and desk
tv and axe and donut
banana and backpack and bus
mouse and bench and chair
...
plate and bird and blender
orange and bottle and keyboard
cake and broccoli and scissors
elephant and boat and banana
Three Objects Texts Dataset

Figure 8: Illustration of the Text-based Object Classification experiment. The figure demonstrates how embeddings are calculated for multi-object captions using CLIP's text encoder. A single-layer classifier is then trained on these embeddings to classify individual objects.

A.2.2 Methodology

1. **Dataset Preparation**:
   - We used both the SimCO and ComCO datasets, which contain captions describing scenes with 2 to 5 objects.
   - Each caption in the dataset follows a consistent format: "Object1 and Object2 and ... and ObjectN".
2. **Text Embedding Generation**:
   - For each multi-object caption, we used CLIP's text encoder to generate a text embedding.
   - This embedding is a high-dimensional vector representation of the entire caption.
3. **Classifier Training**:
   - For each object position (1st, 2nd, 3rd, etc.), we trained a separate single-layer classifier.
   - Input: The text embedding of the multi-object caption.
   - Output: The predicted object class for that specific position.
4. **Evaluation**:
   - We tested each classifier on a held-out portion of the dataset.
   - For each caption, we recorded whether the classifier correctly identified the object at its respective position.
   - We calculated the classification accuracy for each object position across all test captions.

We conducted the TOC experiment on various models under different scenarios, and the results are presented in Table 8. This experiment was repeated on both the SIMCO and ComCO datasets.

Table 8: Text-based Object Classification

Number of Objects	Dataset	Model	**First Object**	**Second Object**	**Third Object**	**Fourth Object**	**Fifth Object**
n = 2	SimCO	*ViT-H-14 (DFN)*	99.86	97.09	-	-	-
		ViT-SO400M-SigLIP	98.67	91.29	-	-	-
		ViT-L-14 (datacomp)	99.76	96.77	-	-	-
		xlm-roberta-large-ViT-H-14	99.03	89.87	-	-	-
		ViT-L-14 (laion2b)	99.70	97.57	-	-	-
		ViT-L-14 (openai)	97.62	91.30	-	-	-
		ViT-B-32 (openai)	96.85	73.00	-	-	-
		NegCLIP	98.19	84.43	-	-	-
	ComCO	*ViT-H-14 (DFN)*	99.90	96.56	-	-	-
		ViT-SO400M-SigLIP	98.47	93.18	-	-	-
		ViT-L-14 (datacomp)	99.74	96.86	-	-	-
		xlm-roberta-large-ViT-H-14	99.16	91.57	-	-	-
		ViT-L-14 (laion2b)	99.72	96.24	-	-	-
		ViT-L-14 (openai)	97.93	96.69	-	-	-
		ViT-B-32 (openai)	96.86	85.42	-	-	-
		NegCLIP	99.30	92.09	-	-	-
n = 3	SimCO	*ViT-H-14 (DFN)*	99.46	60.47	76.99	-	-
		ViT-SO400M-SigLIP	98.23	71.42	45.80	-	-
		ViT-L-14 (datacomp)	99.49	45.80	78.66	-	-
		xlm-roberta-large-ViT-H-14	99.26	49.08	64.07	-	-
		ViT-L-14 (laion2b)	98.93	56.87	72.37	-	-
		ViT-L-14 (openai)	91.87	50.75	68.38	-	-
		ViT-B-32 (openai)	92.55	38.61	52.94	-	-
		NegCLIP	95.80	44.70	59.11	-	-
	ComCO	*ViT-H-14 (DFN)*	99.73	59.80	73.63	-	-
		ViT-SO400M-SigLIP	96.94	70.26	29.28	-	-
		ViT-L-14 (datacomp)	99.53	45.13	74.15	-	-
		xlm-roberta-large-ViT-H-14	99.20	53.34	57.15	-	-
		ViT-L-14 (laion2b)	99.26	58.58	64.74	-	-
		ViT-L-14 (openai)	90.86	49.67	83.49	-	-
		ViT-B-32 (openai)	87.97	45.77	63.13	-	-
		NegCLIP	56.94	98.03	56.66	-	-
n = 4	SimCO	*ViT-H-14 (DFN)*	99.46	34.57	36.73	62.35	-
		ViT-SO400M-SigLIP	98.23	69.91	26.10	6.54	-
		ViT-L-14 (datacomp)	99.00	23.76	35.55	60.55	-
		xlm-roberta-large-ViT-H-14	99.26	27.97	28.84	48.34	-
		ViT-L-14 (laion2b)	98.82	34.21	31.41	54.73	-
		ViT-L-14 (openai)	90.48	35.19	30.50	59.29	-
		ViT-B-32 (openai)	90.76	22.77	25.36	40.45	-
		NegCLIP	96.50	9.33	4.79	15.58	-
	ComCO	*ViT-H-14 (DFN)*	99.76	31.74	35.29	54.82	-
		ViT-SO400M-SigLIP	97.27	72.51	33.25	5.79	-
		ViT-L-14 (datacomp)	99.46	22.82	32.93	58.18	-
		xlm-roberta-large-ViT-H-14	99.60	26.27	26.20	36.51	-
		ViT-L-14 (laion2b)	98.89	31.64	20.90	47.76	-
		ViT-L-14 (openai)	87.17	30.60	31.69	74.49	-
		ViT-B-32 (openai)	88.24	24.23	28.30	49.82	-
		NegCLIP	98.73	28.05	30.83	43.82	-
n = 5	SimCO	*ViT-H-14 (DFN)*	99.00	24.30	22.33	27.23	53.03
		ViT-SO400M-SigLIP	97.79	71.67	27.41	6.29	6.48
		ViT-L-14 (datacomp)	98.89	16.51	21.29	26.92	48.52
		xlm-roberta-large-ViT-H-14	99.46	17.15	16.63	20.18	35.64
		ViT-L-14 (laion2b)	98.43	25.51	19.81	23.15	41.07
		ViT-L-14 (openai)	89.79	26.33	20.74	24.69	50.29
		ViT-B-32 (openai)	92.73	15.67	17.03	19.58	33.62
		NegCLIP	96.83	15.50	17.54	22.58	36.40
	ComCO	*ViT-H-14 (DFN)*	99.80	19.44	20.79	24.86	42.38
		ViT-SO400M-SigLIP	97.63	70.57	32.34	5.42	5.72
		ViT-L-14 (datacomp)	99.13	14.75	19.89	25.72	47.11
		xlm-roberta-large-ViT-H-14	99.40	18.21	15.47	18.05	26.12
		ViT-L-14 (laion2b)	98.76	20.91	18.11	20.77	33.54
		ViT-L-14 (openai)	86.13	22.11	19.43	28.03	68.37
		ViT-B-32 (openai)	91.20	15.56	13.31	19.66	39.39
		NegCLIP	99.03	16.69	16.51	22.26	34.29

A.3 Text-based Object Retrieval

A.3.1 Objective

The Text-based Object Retrieval (TOR) experiment was designed to assess CLIP's text encoder's ability to retrieve individual objects from multi-object captions. This experiment aimed to investigate potential biases in object retrieval based on the object's position within the caption.

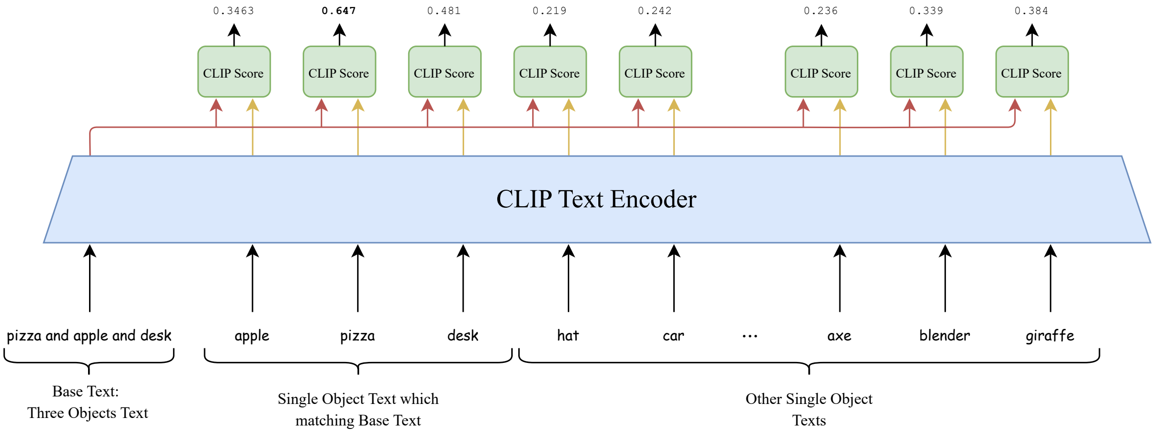


Figure 9: Visualization of the Text-based Object Retrieval experiment. This diagram illustrates the process of retrieving single-object texts based on multi-object captions using CLIP's text encoder.

A.4 Methodology

1. **Dataset Preparation**:
    - We utilized both the SimCO and ComCO datasets, containing captions describing scenes with 2 to 5 objects.
    - Each multi-object caption followed the format: "Object1 and Object2 and ... and ObjectN".
    - We also prepared a set of single-object captions for each object class in our datasets.
2. **Text Embedding Generation**:
    - We used CLIP's text encoder to generate embeddings for all multi-object captions.
    - Similarly, we generated embeddings for all single-object captions.
3. **Similarity Computation**:
    - For each multi-object caption, we computed the cosine similarity between its embedding and the embeddings of all single-object captions.
4. **Object Retrieval**:
    - For each multi-object caption, we identified the single-object caption with the highest similarity score.
    - We recorded which object from the multi-object caption (1st, 2nd, 3rd, etc.) matched this retrieved single-object caption.
5. **Evaluation**:
    - We calculated the percentage of times each object position (1st, 2nd, 3rd, etc.) was retrieved as the most similar.
    - This percentage represents the retrieval accuracy for each object position.

We repeated the TOR experiment on various models across scenarios with captions containing 2 to 5 objects. This was done to confirm the presence of the discovered bias. The complete results of this experiment, which was conducted on both the SIMCO and ComCO datasets, can be observed in Table 9.

Table 9: Text-based Object Retrieval

Number of Objects	Dataset	Model	**First Object**	**Second Object**	**Third Object**	**Fourth Object**	**Fifth Object**
n = 2	SimCO	*ViT-H-14 (DFN)*	69.18	30.82	-	-	-
		ViT-SO400M-SigLIP	68.87	31.13	-	-	-
		ViT-L-14 (datacomp)	69.93	30.07	-	-	-
		xlm-roberta-large-ViT-H-14	78.95	21.05	-	-	-
		ViT-L-14 (laion2b)	68.66	31.34	-	-	-
		ViT-L-14 (openai)	75.82	24.18	-	-	-
		ViT-B-32 (openai)	81.05	18.95	-	-	-
		NegCLIP	77.78	22.22	-	-	-
	ComCO	*ViT-H-14 (DFN)*	70.87	29.13	-	-	-
		ViT-SO400M-SigLIP	67.56	32.44	-	-	-
		ViT-L-14 (datacomp)	70.37	26.93	-	-	-
		xlm-roberta-large-ViT-H-14	59.15	40.85	-	-	-
		ViT-L-14 (laion2b)	70.84	29.16	-	-	-
		ViT-L-14 (openai)	66.03	33.97	-	-	-
		ViT-B-32 (openai)	61.62	38.38	-	-	-
		NegCLIP	64.13	35.87	-	-	-
n = 3	SimCO	*ViT-H-14 (DFN)*	62.05	18.07	19.88	-	-
		ViT-SO400M-SigLIP	58.05	20.50	21.46	-	-
		ViT-L-14 (datacomp)	61.68	20.35	17.96	-	-
		xlm-roberta-large-ViT-H-14	66.75	23.86	9.39	-	-
		ViT-L-14 (laion2b)	62.31	12.56	25.13	-	-
		ViT-L-14 (openai)	65.71	16.67	17.62	-	-
		ViT-B-32 (openai)	74.23	13.62	12.15	-	-
		NegCLIP	77.43	13.75	8.83	-	-
	ComCO	*ViT-H-14 (DFN)*	67.08	22.19	10.73	-	-
		ViT-SO400M-SigLIP	61.11	23.33	15.56	-	-
		ViT-L-14 (datacomp)	72.23	19.05	8.72	-	-
		xlm-roberta-large-ViT-H-14	43.60	31.36	25.05	-	-
		ViT-L-14 (laion2b)	66.85	23.52	9.63	-	-
		ViT-L-14 (openai)	57.66	26.75	15.59	-	-
		ViT-B-32 (openai)	55.73	28.28	15.98	-	-
		NegCLIP	57.56	29.45	12.99	-	-
n = 4	SimCO	*ViT-H-14 (DFN)*	60.06	12.77	12.03	15.14	-
		ViT-SO400M-SigLIP	53.54	14.76	11.43	20.27	-
		ViT-L-14 (datacomp)	62.16	15.99	10.41	11.44	-
		xlm-roberta-large-ViT-H-14	62.58	22.52	10.91	3.99	-
		ViT-L-14 (laion2b)	67.81	8.97	5.80	17.41	-
		ViT-L-14 (openai)	66.87	11.59	6.18	15.35	-
		ViT-B-32 (openai)	76.37	10.03	7.50	6.55	-
		NegCLIP	82.90	10.20	4.61	2.29	-
	ComCO	*ViT-H-14 (DFN)*	64.34	19.25	11.14	5.27	-
		ViT-SO400M-SigLIP	58.11	21.16	10.99	9.73	-
		ViT-L-14 (datacomp)	71.13	16.26	8.74	3.87	-
		xlm-roberta-large-ViT-H-14	44.03	23.73	18.07	14.18	-
		ViT-L-14 (laion2b)	63.96	21.59	10.68	3.76	-
		ViT-L-14 (openai)	48.20	26.01	10.74	8.74	-
		ViT-B-32 (openai)	50.31	20.74	15.45	6.79	-
		NegCLIP	51.63	28.92	14.86	4.59	-
n = 5	SimCO	*ViT-H-14 (DFN)*	60.80	10.61	8.35	9.02	11.22
		ViT-SO400M-SigLIP	49.47	13.32	3.39	11.97	21.25
		ViT-L-14 (datacomp)	66.43	16.12	6.59	4.99	5.87
		xlm-roberta-large-ViT-H-14	60.65	21.03	11.90	5.15	1.28
		ViT-L-14 (laion2b)	74.07	9.51	4.48	2.80	9.14
		ViT-L-14 (openai)	71.71	10.59	2.99	2.71	12.00
		ViT-B-32 (openai)	43.86	26.41	15.44	8.57	5.72
		NegCLIP	85.00	10.39	3.12	1.24	0.26
	ComCO	*ViT-H-14 (DFN)*	61.06	17.00	11.98	6.69	3.27
		ViT-SO400M-SigLIP	55.77	19.25	10.24	6.73	8.01
		ViT-L-14 (datacomp)	68.96	14.61	9.40	4.77	2.25
		xlm-roberta-large-ViT-H-14	28.86	26.87	19.42	14.61	10.24
		ViT-L-14 (laion2b)	61.93	19.10	11.65	5.11	2.21
		ViT-L-14 (openai)	38.40	24.80	18.79	11.04	6.68
		ViT-B-32 (openai)	44.71	26.69	16.44	8.37	3.79
		NegCLIP	45.70	27.56	17.03	7.57	2.15

A.5 Image-based Object Classification

A.5.1 Objective

The Image-based Object Classification (IOC) experiment was designed to evaluate CLIP's image encoder's ability to represent individual objects within multi-object images. This experiment aimed to investigate potential biases in object classification based on the object's size within the image.

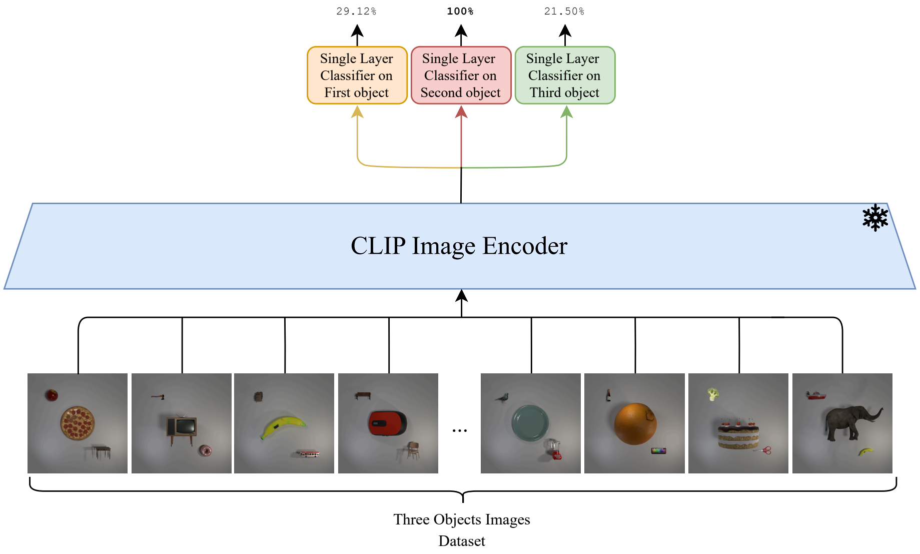

29.12%
100%
21.50%
Single Layer Classifier on First object
Single Layer Classifier on Second object
Single Layer Classifier on Third object
CLIP Image Encoder
...
Three Objects Images Dataset

Figure 10: Illustration of the Image-based Object Classification experiment with the ComCO dataset. The diagram shows the process of classifying individual objects in K-object images using CLIP's image encoder, with a single-layer classifier trained on the generated image embeddings

A.5.2 Methodology

1. **Dataset Preparation**:
   - We utilized both the SimCO and ComCO datasets, containing images with 2 to 5 objects.
   - In each image, one object was deliberately made larger than the others.
   - The position of the larger object was varied across images to avoid position-based biases.
2. **Image Embedding Generation**:
   - For each multi-object image, we used CLIP's image encoder to generate an image embedding.
   - This embedding is a high-dimensional vector representation of the entire image.
3. **Classifier Training**:
   - We trained separate single-layer classifiers for each object position (large object, small object 1, small object 2, etc.).
   - Input: The image embedding of the multi-object image.
   - Output: The predicted object class for that specific position/size.
4. **Evaluation**:
   - We tested each classifier on a held-out portion of the dataset.
   - For each image, we recorded whether the classifier correctly identified the object at its respective position/size.
   - We calculated the classification accuracy for each object position/size across all test images.

We conducted the IOC experiment on images from both datasets, focusing on scenarios with one significantly larger object in varying positions. The experiment was repeated across models, and the average results are shown in Table 10.

Table 10: Image-based Object Classification

Number of Objects	Dataset	Model	**Large Object**	**Small Obj 1**	**Small Obj 2**	**Small Obj 3**	**Small Obj 4**
n = 2	SimCO	*ViT-H-14 (DFN)*	88.1	14.29	-	-	-
		ViT-SO400M-SigLIP	97.62	16.67	-	-	-
		ViT-L-14 (datacomp)	83.33	11.9	-	-	-
		xlm-roberta-large-ViT-H-14	78.57	21.43	-	-	-
		ViT-L-14 (laion2b)	66.67	11.9	-	-	-
		ViT-L-14 (openai)	64.29	0.00	-	-	-
		ViT-B-32 (openai)	61.9	0.00	-	-	-
		NegCLIP	40.48	7.14	-	-	-
	ComCO	*ViT-H-14 (DFN)*	100.0	26.36	-	-	-
		ViT-SO400M-SigLIP	100.0	33.9	-	-	-
		ViT-L-14 (datacomp)	100.0	42.35	-	-	-
		xlm-roberta-large-ViT-H-14	100.0	40.85	-	-	-
		ViT-L-14 (laion2b)	100.0	31.29	-	-	-
		ViT-L-14 (openai)	99.8	41.29	-	-	-
		ViT-B-32 (openai)	99.8	35.81	-	-	-
		NegCLIP	99.6	41.95	-	-	-
n = 3	SimCO	*ViT-H-14 (DFN)*	100.0	35.65	41.57	-	-
		ViT-SO400M-SigLIP	99.8	42.8	49.03	-	-
		ViT-L-14 (datacomp)	100.0	39.94	51.28	-	-
		xlm-roberta-large-ViT-H-14	99.9	48.42	56.28	-	-
		ViT-L-14 (laion2b)	99.8	45.56	56.08	-	-
		ViT-L-14 (openai)	98.98	39.73	50.46	-	-
		ViT-B-32 (openai)	96.12	38.1	51.58	-	-
		NegCLIP	97.04	42.59	59.35	-	-
	ComCO	*ViT-H-14 (DFN)*	100.0	29.12	21.5	-	-
		ViT-SO400M-SigLIP	100.0	30.94	29.94	-	-
		ViT-L-14 (datacomp)	100.0	36.56	33.5	-	-
		xlm-roberta-large-ViT-H-14	100.0	33.69	32.31	-	-
		ViT-L-14 (laion2b)	100.0	35.44	30.31	-	-
		ViT-L-14 (openai)	99.94	33.31	34.31	-	-
		ViT-B-32 (openai)	99.94	29.0	32.94	-	-
		NegCLIP	99.81	33.88	43.0	-	-
n = 4	SimCO	*ViT-H-14 (DFN)*	100.0	40.06	34.06	41.31	-
		ViT-SO400M-SigLIP	100.0	47.0	38.5	41.06	-
		ViT-L-14 (datacomp)	100.0	48.94	38.38	45.06	-
		xlm-roberta-large-ViT-H-14	100.0	48.19	35.81	46.38	-
		ViT-L-14 (laion2b)	100.0	50.5	41.81	43.94	-
		ViT-L-14 (openai)	100.0	45.19	38.38	39.0	-
		ViT-B-32 (openai)	100.0	38.06	31.5	37.25	-
		NegCLIP	100.0	42.0	37.25	46.94	-
	ComCO	*ViT-H-14 (DFN)*	100.0	16.64	14.13	12.38	-
		ViT-SO400M-SigLIP	100.0	18.95	15.57	17.57	-
		ViT-L-14 (datacomp)	100.0	20.64	21.01	19.01	-
		xlm-roberta-large-ViT-H-14	100.0	20.45	18.45	16.51	-
		ViT-L-14 (laion2b)	100.0	19.76	17.57	18.89	-
		ViT-L-14 (openai)	99.94	19.32	21.89	22.39	-
		ViT-B-32 (openai)	100.0	21.58	21.83	22.26	-
		NegCLIP	100.0	21.89	23.64	31.33	-
n = 5	SimCO	*ViT-H-14 (DFN)*	100.0	34.0	30.0	30.38	21.62
		ViT-SO400M-SigLIP	100.0	38.5	34.7	27.38	25.62
		ViT-L-14 (datacomp)	100.0	40.38	36.12	32.0	24.75
		xlm-roberta-large-ViT-H-14	100.0	41.56	39.56	36.69	32.81
		ViT-L-14 (laion2b)	100.0	43.88	39.5	34.0	28.94
		ViT-L-14 (openai)	100.0	42.19	36.38	32.81	31.94
		ViT-B-32 (openai)	98.81	36.25	35.38	33.88	26.06
		NegCLIP	99.19	40.88	37.94	37.56	28.94
	ComCO	*ViT-H-14 (DFN)*	100.0	13.88	9.38	9.32	11.94
		ViT-SO400M-SigLIP	100.0	15.51	13.88	14.57	14.76
		ViT-L-14 (datacomp)	100.0	18.2	15.07	16.07	18.32
		xlm-roberta-large-ViT-H-14	99.94	15.38	14.88	15.26	19.14
		ViT-L-14 (laion2b)	100.0	15.51	12.32	14.13	17.95
		ViT-L-14 (openai)	100.0	15.38	14.76	16.76	20.01
		ViT-B-32 (openai)	99.87	17.76	18.64	19.2	23.14
		NegCLIP	100	18.89	16.57	23.51	28.77

A.6 Image-based Object Retrieval

A.6.1 Objective

The Image-based Object Retrieval (IOR) experiment was designed to assess CLIP's image encoder's ability to retrieve individual objects from multi-object images. This experiment aimed to investigate potential biases in object retrieval based on the object's size within the image.

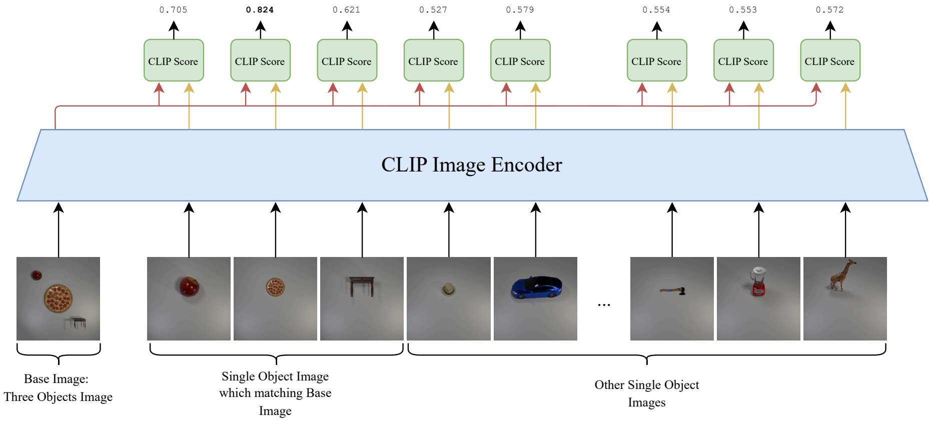


Figure 11: Visualization of the Image-based Object Retrieval experiment. This diagram illustrates the process of retrieving single-object images based on multi-object image inputs using CLIP's image encoder. The experiment employs a base image containing three objects of varying sizes. CLIP scores are computed between the embedding of this multi-object image and embeddings of various single-object images.

A.6.2 Methodology

1. **Dataset Preparation**:
    - We utilized both the SimCO and ComCO datasets, containing images with 2 to 5 objects.
    - In each multi-object image, one object was deliberately made larger than the others.
    - The position of the larger object was varied across images to avoid position-based biases.
    - We also prepared a set of single-object images for each object class in our datasets.
2. **Image Embedding Generation**:
    - We used CLIP's image encoder to generate embeddings for all multi-object images.
    - Similarly, we generated embeddings for all single-object images.
3. **Similarity Computation**:
    - For each multi-object image, we computed the cosine similarity between its embedding and the embeddings of all single-object images.
4. **Object Retrieval**:
    - For each multi-object image, we identified the single-object image with the highest similarity score.
    - We recorded whether the retrieved single-object image corresponded to the large object or one of the small objects in the multi-object image.
5. **Evaluation**:
    - We calculated the percentage of times the large object and each small object were retrieved as the most similar.
    - This percentage represents the retrieval accuracy for each object size category (large object, small object 1, small object 2, etc.).

We conducted the IOR experiment on images from the SimCO and ComCO datasets with 2 to 5 objects, varying the position of the larger object to avoid location-based biases. The results are shown in Table 11.

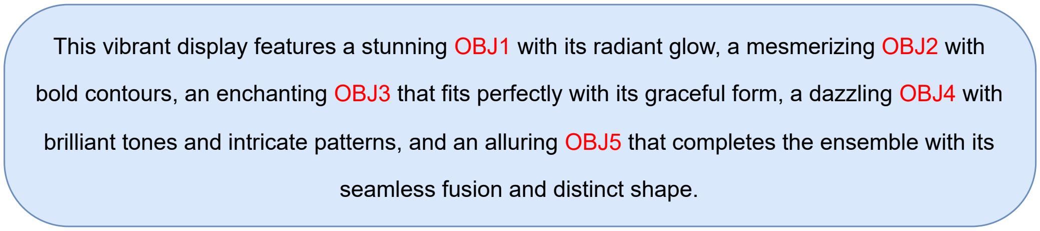
Table 11: Image-based Object Retrieval

Number of Objects	Dataset	Model	**Large Object**	**Small Obj 1**	**Small Obj 2**	**Small Obj 3**	**Small Obj 4**
n = 2	SimCO	*ViT-H-14 (DFN)*	99.11	0.89	-	-	-
		ViT-SO400M-SigLIP	91.67	8.33	-	-	-
		ViT-L-14 (datacomp)	91.96	8.04	-	-	-
		xlm-roberta-large-ViT-H-14	94.92	5.08	-	-	-
		ViT-L-14 (laion2b)	92.86	7.14	-	-	-
		ViT-L-14 (openai)	87.88	12.12	-	-	-
		ViT-B-32 (openai)	90.24	9.76	-	-	-
		NegCLIP	94.64	5.36	-	-	-
	ComCO	*ViT-H-14 (DFN)*	97.35	2.65	-	-	-
		ViT-SO400M-SigLIP	95.13	4.87	-	-	-
		ViT-L-14 (datacomp)	89.85	10.15	-	-	-
		xlm-roberta-large-ViT-H-14	93.89	6.11	-	-	-
		ViT-L-14 (laion2b)	94.84	5.16	-	-	-
		ViT-L-14 (openai)	83.7	16.30	-	-	-
		ViT-B-32 (openai)	86.86	13.14	-	-	-
		NegCLIP	83.3	16.7	-	-	-
n = 3	SimCO	*ViT-H-14 (DFN)*	93.80	0.65	5.55	-	-
		ViT-SO400M-SigLIP	83.27	5.61	11.12	-	-
		ViT-L-14 (datacomp)	77.16	5.81	17.04	-	-
		xlm-roberta-large-ViT-H-14	80.21	5.12	14.66	-	-
		ViT-L-14 (laion2b)	76.57	9.57	13.86	-	-
		ViT-L-14 (openai)	72.07	8.66	19.27	-	-
		ViT-B-32 (openai)	61.14	14.69	24.17	-	-
		NegCLIP	59.13	14.91	25.96	-	-
	ComCO	*ViT-H-14 (DFN)*	96.52	1.71	17.8	-	-
		ViT-SO400M-SigLIP	90.5	5.47	4.03	-	-
		ViT-L-14 (datacomp)	89.65	6.09	4.26	-	-
		xlm-roberta-large-ViT-H-14	91.39	4.92	3.69	-	-
		ViT-L-14 (laion2b)	91.26	3.28	5.46	-	-
		ViT-L-14 (openai)	74.2	12.79	13.01	-	-
		ViT-B-32 (openai)	80.6	5.22	14.18	-	-
		NegCLIP	76.36	10.47	13.18	-	-
n = 4	SimCO	*ViT-H-14 (DFN)*	99.5	0.0	0.0	0.5	-
		ViT-SO400M-SigLIP	91.03	1.28	2.99	4.7	-
		ViT-L-14 (datacomp)	89.71	3.43	3.61	3.25	-
		xlm-roberta-large-ViT-H-14	92.47	2.08	2.60	2.86	-
		ViT-L-14 (laion2b)	86.92	4.67	3.74	4.67	-
		ViT-L-14 (openai)	70.55	13.01	7.53	8.9	-
		ViT-B-32 (openai)	52.17	18.84	13.04	15.94	-
		NegCLIP	74.4	10.4	7.2	8.0	-
	ComCO	*ViT-H-14 (DFN)*	95.86	2.55	1.27	0.32	-
		ViT-SO400M-SigLIP	94.03	2.24	1.49	2.24	-
		ViT-L-14 (datacomp)	93.3	3.91	1.12	16.8	-
		xlm-roberta-large-ViT-H-14	90.91	2.02	5.05	2.02	-
		ViT-L-14 (laion2b)	91.78	5.48	2.74	0.0	-
		ViT-L-14 (openai)	67.86	14.29	7.14	10.71	-
		ViT-B-32 (openai)	85.0	0.0	5.0	10.0	-
		NegCLIP	79.55	0.0	2.27	18.19	-
n = 5	SimCO	*ViT-H-14 (DFN)*	100.0	0.0	0.0	0.0	0.0
		ViT-SO400M-SigLIP	94.92	3.39	1.69	0.0	0.0
		ViT-L-14 (datacomp)	91.3	5.59	1.24	1.24	0.62
		xlm-roberta-large-ViT-H-14	77.42	11.83	5.38	3.23	2.15
		ViT-L-14 (laion2b)	81.01	8.86	5.06	1.27	0.38
		ViT-L-14 (openai)	77.14	8.57	5.71	5.71	2.86
		ViT-B-32 (openai)	68.75	25.0	6.25	0.0	0.0
		NegCLIP	58.62	17.24	15.52	5.17	3.45
	ComCO	*ViT-H-14 (DFN)*	95.16	1.61	1.61	0.0	1.61
		ViT-SO400M-SigLIP	80.0	0.0	0.0	0.0	20.0
		ViT-L-14 (datacomp)	90.91	4.55	0.0	0.0	4.55
		xlm-roberta-large-ViT-H-14	100.0	0.0	0.0	0.0	0.0
		ViT-L-14 (laion2b)	100.0	0.0	0.0	0.0	0.0
		ViT-L-14 (openai)	100.0	0.0	0.0	0.0	0.0
		ViT-B-32 (openai)	100.0	0.0	0.0	0.0	0.
		NegCLIP	50.0	0.0	0.0	50.0	0.0

A.7 TEXT-BASED OBJECT CLASSIFICATION FOR LONG CAPTION

In this section, we revisited the IOC experiment with a significant modification to the caption structure. Our objective was to investigate whether the previously observed bias persists in longer, more elaborate captions. We achieved this by expanding the caption template, incorporating additional descriptive phrases between object mentions.

The extended caption template used in this experiment was as follows:



This vibrant display features a stunning OBJ1 with its radiant glow, a mesmerizing OBJ2 with bold contours, an enchanting OBJ3 that fits perfectly with its graceful form, a dazzling OBJ4 with brilliant tones and intricate patterns, and an alluring OBJ5 that completes the ensemble with its seamless fusion and distinct shape.

Figure 12: Format for Extended Caption Template

This template allowed us to maintain a consistent structure while significantly increasing the caption length and complexity.

The results of this modified IOC experiment are presented in Table 12. Notably, the observed pattern closely resembles that of the standard IOC experiment. This similarity suggests that the bias identified in shorter captions persists even in more elaborate textual descriptions.

A.8 TEXT-BASED OBJECT RETRIEVAL FOR LONG CAPTION

In this section, we aimed to examine the performance of various models in the IOR experiment when presented with longer caption formats. This approach mirrors our previous investigation, allowing us to draw comparisons between standard and extended caption scenarios.

We utilized the same extended caption template as in the previous section. The results of this experiment are presented in Table 13. Notably, the observed pattern closely aligns with that of the standard IOR experiment, suggesting a consistency in model behavior across different caption lengths.

Table 12: Text-based Object Classification on Long Captions

Number of Objects	Dataset	Model	**First Object**	**Second Object**	**Third Object**	**Fourth Object**	**Fifth Object**
n = 2	SimCO	*ViT-H-14 (DFN)*	100.0	89.01	-	-	-
		ViT-SO400M-SigLIP	100.0	93.83	-	-	-
		ViT-L-14 (datacomp)	100.0	63.22	-	-	-
		xlm-roberta-large-ViT-H-14	99.82	51.83	-	-	-
		ViT-L-14 (laion2b)	100.0	85.88	-	-	-
		ViT-L-14 (openai)	99.65	98.26	-	-	-
		ViT-B-32 (openai)	100.0	72.69	-	-	-
		NegCLIP	100	89.59	-	-	-
	ComCO	*ViT-H-14 (DFN)*	99.99	99.86	-	-	-
		ViT-SO400M-SigLIP	100	99.48	-	-	-
		ViT-L-14 (datacomp)	100	98.89	-	-	-
		xlm-roberta-large-ViT-H-14	99.95	92.84	-	-	-
		ViT-L-14 (laion2b)	100	99.03	-	-	-
		ViT-L-14 (openai)	99.99	99.99	-	-	-
		ViT-B-32 (openai)	99.59	99.45	-	-	-
		NegCLIP	99.94	98.99	-	-	-
n = 3	SimCO	*ViT-H-14 (DFN)*	99.34	43.49	89.66	-	-
		ViT-SO400M-SigLIP	100.0	65.26	49.76	-	-
		ViT-L-14 (datacomp)	100.0	30.47	37.20	-	-
		xlm-roberta-large-ViT-H-14	97.78	22.96	27.23	-	-
		ViT-L-14 (laion2b)	99.65	57.67	35.51	-	-
		ViT-L-14 (openai)	99.13	86.67	58.22	-	-
		ViT-B-32 (openai)	96.26	54.19	44.88	-	-
		NegCLIP	98.30	67.60	65.90	-	-
	ComCO	*ViT-H-14 (DFN)*	99.31	78.44	84.15	-	-
		ViT-SO400M-SigLIP	99.93	67.22	76.89	-	-
		ViT-L-14 (datacomp)	98.98	85.77	65.64	-	-
		xlm-roberta-large-ViT-H-14	99.21	38.60	60.10	-	-
		ViT-L-14 (laion2b)	98.81	82.72	74.31	-	-
		ViT-L-14 (openai)	99.41	96.44	82.18	-	-
		ViT-B-32 (openai)	95.59	81.91	76.09	-	-
		NegCLIP	98.62	74.29	81.70	-	-
n = 4	SimCO	*ViT-H-14 (DFN)*	99.17	24.74	67.00	41.46	-
		ViT-SO400M-SigLIP	100.0	46.75	24.40	20.93	-
		ViT-L-14 (datacomp)	100.0	15.27	17.79	43.03	-
		xlm-roberta-large-ViT-H-14	98.87	13.34	12.67	15.85	-
		ViT-L-14 (laion2b)	99.56	36.03	19.23	34.51	-
		ViT-L-14 (openai)	98.22	70.29	40.54	50.71	-
		ViT-B-32 (openai)	97.47	41.20	25.18	24.31	-
		NegCLIP	98.93	49.58	35.89	35.40	-
	ComCO	*ViT-H-14 (DFN)*	98.34	62.49	70.25	42.34	-
		ViT-SO400M-SigLIP	99.90	39.28	58.01	32.51	-
		ViT-L-14 (datacomp)	97.95	71.61	37.24	48.50	-
		xlm-roberta-large-ViT-H-14	99.34	20.38	21.45	25.08	-
		ViT-L-14 (laion2b)	98.41	66.90	51.43	38.87	-
		ViT-L-14 (openai)	96.39	88.74	62.87	75.1	-
		ViT-B-32 (openai)	96.81	62.50	59.19	22.93	-
		NegCLIP	98.50	45.93	40.11	68.58	-
n = 5	SimCO	*ViT-H-14 (DFN)*	97.44	18.82	53.68	26.08	47.45
		ViT-SO400M-SigLIP	100.0	20.35	19.30	12.57	18.40
		ViT-L-14 (datacomp)	99.74	17.57	19.29	41.34	23.67
		xlm-roberta-large-ViT-H-14	99.09	12.51	8.49	8.63	30.25
		ViT-L-14 (laion2b)	99.69	60.13	28.18	49.20	54.92
		ViT-L-14 (openai)	96.26	70.36	44.68	36.7	48.1
		ViT-B-32 (openai)	96.79	30.71	15.25	12.58	41.30
		NegCLIP	99.35	32.26	22.22	16.39	62.63
	ComCO	*ViT-H-14 (DFN)*	97.45	43.49	29.20	17.91	1.13
		ViT-SO400M-SigLIP	98.46	45.21	32.54	26.64	1.18
		ViT-L-14 (datacomp)	92.76	40.83	17.56	9.8	1.05
		xlm-roberta-large-ViT-H-14	99.84	13.18	11.02	8.26	45.38
		ViT-L-14 (laion2b)	97.39	41.48	19.5	9.4	1.26
		ViT-L-14 (openai)	92.81	68.46	31.85	9.8	1.24
		ViT-B-32 (openai)	95.85	42.62	22.24	9.18	0.9
		NegCLIP	99.16	27.60	19.78	21.80	69.08

Table 13: Text-based Object Retrieval For long template

Number of Objects	Dataset	Model	**Accuracy**	**First Object**	**Second Object**	**Third Object**	**Fourth Object**	**Fifth Object**
n = 2	SimCO	*ViT-H-14 (DFN)*	96.73	62.16	37.84	-	-	-
		ViT-SO400M-SigLIP	5.88	100.0	0.00	-	-	-
		ViT-L-14 (datacomp)	98.04	70.67	29.33	-	-	-
		xlm-roberta-large-ViT-H-14	98.69	76.82	23.18	-	-	-
		ViT-L-14 (laion2b)	51.63	62.03	37.97	-	-	-
		ViT-L-14 (openai)	96.08	39.46	60.54	-	-	-
		ViT-B-32 (openai)	79.74	45.90	54.10	-	-	-
		NegCLIP	99.35	38.82	61.18	-	-	-
	ComCO	*ViT-H-14 (DFN)*	92.38	71.03	28.97	-	-	-
		ViT-SO400M-SigLIP	3.42	100.0	0.00	-	-	-
		ViT-L-14 (datacomp)	84.32	62.63	37.37	-	-	-
		xlm-roberta-large-ViT-H-14	72.06	63.31	36.69	-	-	-
		ViT-L-14 (laion2b)	58.73	63.01	36.99	-	-	-
		ViT-L-14 (openai)	84.64	61.27	38.70	-	-	-
		ViT-B-32 (openai)	78.38	61.77	37.78	-	-	-
		NegCLIP	82.67	55.63	44.37	-	-	-
n = 3	SimCO	*ViT-H-14 (DFN)*	88.6	43.02	30.43	26.56	-	-
		ViT-SO400M-SigLIP	0.74	100.0	0.00	0.00	-	-
		ViT-L-14 (datacomp)	88.48	63.02	24.38	12.60	-	-
		xlm-roberta-large-ViT-H-14	89.83	61.66	22.10	16.23	-	-
		ViT-L-14 (laion2b)	31.86	56.54	26.15	17.31	-	-
		ViT-L-14 (openai)	69.73	24.08	39.89	36.03	-	-
		ViT-B-32 (openai)	38.24	25.96	39.10	34.94	-	-
		NegCLIP	72.30	23.39	52.71	23.90	-	-
	ComCO	*ViT-H-14 (DFN)*	76.75	50.43	22.45	27.12	-	-
		ViT-SO400M-SigLIP	0.07	100.0	0.00	0.00	-	-
		ViT-L-14 (datacomp)	56.14	47.80	34.17	18.03	-	-
		xlm-roberta-large-ViT-H-14	36.78	48.46	28.75	22.79	-	-
		ViT-L-14 (laion2b)	29.17	48.75	35.78	15.47	-	-
		ViT-L-14 (openai)	52.38	43.44	37.00	19.53	-	-
		ViT-B-32 (openai)	49.97	47.58	30.75	21.45	-	-
		NegCLIP	50.80	38.67	38.16	23.17	-	-
n = 4	SimCO	*ViT-H-14 (DFN)*	66.47	39.82	21.88	24.34	13.96	-
		ViT-SO400M-SigLIP	0.49	100.0	0.00	0.00	0.00	-
		ViT-L-14 (datacomp)	74.58	61.74	22.17	10.96	5.13	-
		xlm-roberta-large-ViT-H-14	65.95	53.96	21.36	19.33	5.35	-
		ViT-L-14 (laion2b)	22.42	66.76	17.78	11.22	4.23	-
		ViT-L-14 (openai)	58.73	16.30	32.78	26.49	24.37	-
		ViT-B-32 (openai)	18.43	35.64	37.77	14.18	12.41	-
		NegCLIP	50.78	26.25	49.94	16.73	7.08	-
	ComCO	*ViT-H-14 (DFN)*	52.87	47.87	20.54	22.72	8.87	-
		ViT-SO400M-SigLIP	0.01	100.0	0.00	0.00	0.00	-
		ViT-L-14 (datacomp)	31.36	39.21	30.74	20.94	9.11	-
		xlm-roberta-large-ViT-H-14	14.99	43.03	24.29	19.72	12.96	-
		ViT-L-14 (laion2b)	10.19	42.66	34.16	17.09	6.09	-
		ViT-L-14 (openai)	28.78	35.25	31.55	19.19	13.86	-
		ViT-B-32 (openai)	21.62	43.69	24.57	16.78	14.59	-
		NegCLIP	19.41	30.36	30.38	24.39	14.86	-
n = 5	SimCO	*ViT-H-14 (DFN)*	45.44	43.46	20.45	18.34	11.87	5.88
		ViT-SO400M-SigLIP	0.16	100.0	0.00	0.00	0.00	0.00
		ViT-L-14 (datacomp)	51.45	59.26	22.46	8.12	8.46	1.70
		xlm-roberta-large-ViT-H-14	52.92	54.87	13.81	19.30	8.16	3.86
		ViT-L-14 (laion2b)	12.34	75.40	10.31	8.42	4.26	1.61
		ViT-L-14 (openai)	29.39	8.98	29.39	28.44	15.97	17.20
		ViT-B-32 (openai)	6.69	32.11	38.57	12.22	8.55	8.55
		NegCLIP	17.54	23.15	41.18	24.48	7.65	3.53
	ComCO	*ViT-H-14 (DFN)*	23.56	36.07	19.21	22.65	11.90	10.17
		ViT-SO400M-SigLIP	0.00	100.0	0.00	0.00	0.00	0.00
		ViT-L-14 (datacomp)	12.49	32.55	27.84	23.76	12.73	3.11
		xlm-roberta-large-ViT-H-14	9.26	40.26	21.35	18.16	11.99	8.23
		ViT-L-14 (laion2b)	4.57	38.49	31.50	17.50	8.31	4.20
		ViT-L-14 (openai)	1.75	21.59	18.57	20.25	20.54	19.02
		ViT-B-32 (openai)	1.86	32.72	15.62	14.71	18.36	16.26
		NegCLIP	1.41	24.30	23.17	22.14	17.64	12.75

A.9 LAION Dataset Analysis

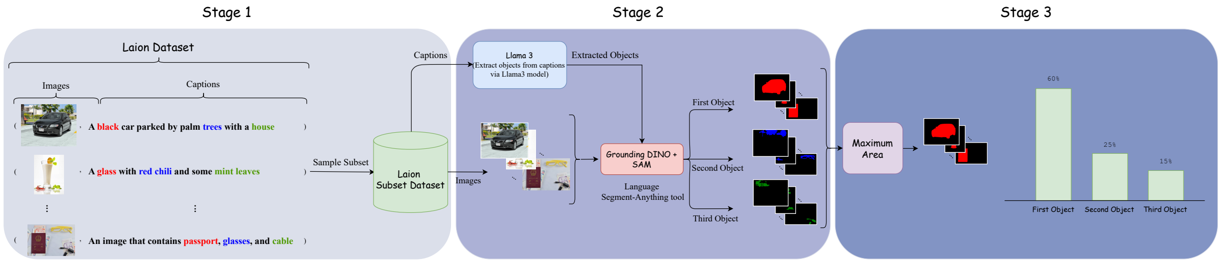


Figure 13: Process flow for LAION dataset analysis

To investigate the potential bias in CLIP's training data, as discussed in Section 4.3, Claim 2, we conducted an analysis of the LAION dataset. This process, illustrated in Figure 13, consisted of three main stages:

A.9.1 Stage 1: Dataset Sampling

Due to the vast size of the LAION dataset (over 2 billion image-text pairs), we randomly selected a subset of 200,000 samples for our analysis. This subset maintained the diversity of the original dataset while making the analysis computationally feasible.

A.9.2 Stage 2: Object Extraction

For each image-caption pair in our subset:

1. We used the Llama 3 model to extract object mentions from the captions. This step allowed us to identify the objects described in each text without relying on manual annotation.
2. We applied the Grounding DINO + SAM (Segment Anything Model) tool to generate object masks for the corresponding images. This process enabled us to identify and segment individual objects within each image.

A.9.3 Stage 3: Analysis

With the extracted data, we performed the following analysis:

1. **Object Order:** We recorded the order in which objects were mentioned in each caption.
2. **Object Size:** Using the generated masks, we calculated the area of each object in the corresponding image.
3. **Correlation:** We examined the relationship between an object's position in the caption and its size in the image.

AS shown in Figure 14 This distribution strongly suggests a bias in the LAION dataset where larger objects tend to be mentioned earlier in image captions. This finding supports our hypothesis about the origin of CLIP's text encoder bias, as discussed in Section 4.3 of the main paper.

A.10 COCO Dataset Analysis

In this section, we repeated the experiment conducted in Section 4.3 for different scenarios involving 2 to 5 objects. We divided the captions in the COCO dataset into four subsets: those mentioning 2 objects, 3 objects, 4 objects, and 5 objects. We then analyzed each subset to determine in what percentage of cases the largest object appeared in which position.

The results of this evaluation are presented in Figure 14. As can be observed, this trend is repeated across all scenarios: in most cases, the larger object appears earlier in the caption.

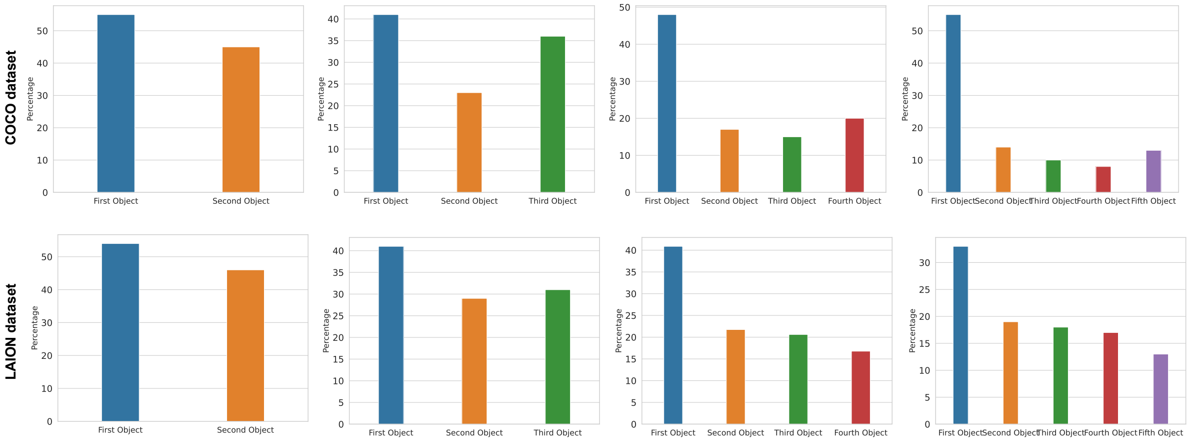


Figure 14: Distribution of larger object positions in captions for objects in COCO and LAION dataset

A.11 Object Categories from DomainNet

The DomainNet dataset objects were categorized into three groups based on their relative sizes: small, medium, and large. These categories were used to investigate potential bias in CLIP's text embeddings, as discussed in Section 4.3, Claim 1. The full list of objects used in each category is presented below:

A.11.1 Small Objects

ant anvil apple arm asparagus axe
banana bandage basket bat bee belt
binoculars bird blackberry blueberry book boomerang
bottlecap bowtie bracelet brain bread broccoli
broom bucket butterfly cactus cake calculator
calendar camera candle carrot cat clarinet
clock compass cookie crab backpack crown
cup dog donut drill duck dumbbell
ear envelope eraser eye eyeglasses feather
finger fork frog hammer hat headphones
hedgehog helmet hourglass jacket keyboard key
knife lantern laptop leaf lipstick lobster
lollipop mailbox marker megaphone microphone microwave
mosquito mouse mug mushroom necklace onion
owl paintbrush parrot peanut pear peas
pencil pillow pineapple pizza pliers popsicle
postcard potato purse rabbit raccoon radio
rake rhinoceros rifle sandwich saw saxophone
scissors scorpion shoe shovel skateboard skull
snail snake snorkel spider spoon squirrel
stethoscope strawberry swan sword syringe teapot
telephone toaster toothbrush trombone trumpet umbrella
violin watermelon wheel

A.11.2 Medium Objects

angel bathtub bear bed bench
bicycle camel cannon canoe cello
chair chandelier computer cooler couch
cow crocodile dishwasher dolphin door
dresser drums flamingo guitar horse
kangaroo ladder mermaid motorbike panda
penguin piano pig sheep stereo
stove table television tiger zebra

A.11.3 LARGE OBJECTS

aircraft carrier
airplane
ambulance
barn
bridge
bulldozer
bus
car
castle
church
cloud
cruise ship
dragon
elephant
firetruck
flying saucer
giraffe
helicopter
hospital
hot air balloon
house
moon
mountain
palm tree
parachute
pickup truck
police car
sailboat
school bus
skyscraper
speedboat
submarine
sun
tent
The Eiffel Tower
The Great Wall of China
tractor
train
tree
truck
van
whale
windmill