Object-centric Binding in Contrastive Language-Image Pretraining

Rim Assouel[1,2,3] **Florian Bordes**[1] **Pietro Astolfi**[1]
Michal Drozdzal[1] **Adriana Romero-Soriano**[1,2,4,5]
[1]FAIR at Meta [2]Mila – Québec AI Institute
[3]Université de Montréal [4]McGill University [5]Canada CIFAR AI Chair

Abstract

Recent advances in vision language models (VLM) have been driven by contrastive models such as CLIP, which learn to associate visual information with their corresponding text descriptions. However, these models have limitations in understanding complex compositional scenes involving multiple objects and their spatial relationships. To address these challenges, we propose a novel approach that diverges from commonly used strategies, which rely on the design of finegrained hard-negative augmentations. Instead, our work focuses on integrating inductive biases into CLIP-like models to improve their compositional understanding. To that end, we introduce a binding module that connects a scene graph, derived from a text description, with a slot-structured image representation, facilitating a structured similarity assessment between the two modalities. We also leverage relationships as text-conditioned visual constraints, thereby capturing the intricate interactions between objects and their contextual relationships more effectively. Our resulting model not only enhances the performance of CLIP-based models in multi-object compositional understanding but also paves the way towards more accurate and sample-efficient image-text matching of complex scenes.

1 Introduction

Recent advancements in multi-modal representation learning have primarily been enabled by the introduction of CLIP [Radford et al., 2021]. CLIP learns aligned image-text representations from Internet-scale data. Despite its success, CLIP exhibits limitations in understanding complex scenes composed of multiple objects [Kamath et al., 2023, Yuksekgonul et al., 2023a, Doveh et al., 2023b, Paiss et al., 2023]. For instance, while capable of recognizing individual objects, CLIP struggles with interpreting spatial relationships among objects in the scene(*e.g.*, "the cat is to the left of the mat" *vs.* "the cat is to the right of the mat") and adequately associating objects with their corresponding attributes (*e.g.*, "a red square and a blue circle" *vs.* "a blue square and a red circle"). The process of acquiring this compositional understanding of the world is known as the *binding problem* in the literature, and may be decomposed into *segregation*, *representation*, and *composition* problems [Greff et al., 2020b].

Efforts to improve the compositional understanding of CLIP-like models have largely relied on leveraging *hard negative examples*[1], either in the text space [Kalantidis et al., 2020, Yuksekgonul et al., 2023b, Zhang et al., 2024b, Doveh et al., 2023b, Paiss et al., 2023] – to improve sensitivity to the order of words and subtle textual differences – or the image space [Awal et al., 2024, Le et al., 2023, Zhang et al., 2024a] – to improve sensitivity to subtle visual differences. Although these methods have somewhat improved CLIP-like models' performance on scene compositionality benchmarks [Parcalabescu et al., 2022, Zhao et al., 2022, Yuksekgonul et al., 2023b, Hsieh et al., 2023b], they do not

[1]Hard-negatives are additional samples that either contain subtle visual changes in the image and/or subtle linguistic/semantic difference in the caption and are sampled as negatives in the same batch.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

explicitly address the binding problem as they focus mainly on enhancing the model's representation capabilities with additional data, hindering their generalization to unseen scene compositions.

Yet, the literature on object-centric representation learning [Eslami et al., 2016, Greff et al., 2020a, Locatello et al., 2020, Wu et al., 2023, Seitzer et al., 2023] has long focused on devising methods to address the segregation and representation problems as a way to facilitate the subsequent compositional processing of images. This has led to the development of inductive biases to segregate different objects in a scene into distinct representational *slots*, which have been shown to naturally scale to an increasing number of visual objects and relations [Locatello et al., 2020, Webb et al., 2023, Mondal et al., 2024, Elsayed et al., 2022]. To the best of our knowledge, advances in object-centric representation learning are yet to be explored in the vision-language domain.

Therefore, in this paper, we focus on enhancing the compositional scene understanding of CLIP-like models by leveraging advances from object-centric representation learning. In particular, we propose to endow CLIP-based vision-language architectures with segregation and composition capabilities. Our core idea is to adapt the slot-centric representation paradigm for CLIP architectures and dynamically align each representational slot with the object entities mentioned in the text. To do so, we design a binding module that connects a scene graph, derived from the textual description, with a slot-structured image representation. We utilize the scene graph's relationships as constraints to effectively capture the complex interactions among the visual entities represented as slots. Our enhanced model, which we refer to as Object-Centric CLIP (OC-CLIP), not only boosts CLIP's performance in understanding multi-object compositional scenes but also improves the sample efficiency of the model when trained from scratch.

Our contributions are summarized as follows:

- We introduce OC-CLIP, a *pretraining method* which endows CLIP-based architectures with inductive biases to address the binding problem.
- We evaluate the sample efficiency of our approach against methods leveraging hard negative augmentations in a controlled 3D environment and show the overall efficiency of OC-CLIP compared to both text and image based a hard-negative augmentations.
- We demonstrate that OC-CLIP significantly enhances the binding of object-centric attributes and spatial relationships across a representative set of challenging real-world compositional image-text matching benchmarks. Notably, we report an increase of **16.5%** accuracy in the challenging *swap-attribute* split of SugarCrepe compared to OpenCLIP [Ilharco et al., 2021] finetuned in-domain, and go from random chance to more than **89%** on COCO-spatial and **92%**on GQA-spatial from the Whatsup benchmark [Kamath et al., 2023].
- We show the scaling potential of OC-CLIP when trained from scratch on noisy data [Changpinyo et al., 2021, Sharma et al., 2018]. We report an increase of **12.7%** accuracy in zero-shot ImageNet classification compared to OpenCLIP.

2 Related Work

Contrastive Pretraining of VLMs. Vision-language models (VLMs) have made substantial strides in both the vision and multi-modal domains [Bordes et al., 2024]. Modern VLMs are pretrained on vast, diverse and oftentimes noisy multi-modal datasets [Changpinyo et al., 2021, Schuhmann et al., 2022, Ilharco et al., 2021, Zeng et al., 2022], and have shown substantial improvements when applied to various zero-shot tasks. CLIP [Radford et al., 2021] presented a contrastive learning approach used for pretraining, which involves training the model to differentiate between similar and dissimilar image-text pairs. This approach encourages the model to learn a shared representation space for images and text, where semantically similar pairs are close together and dissimilar pairs are far apart. Following CLIP's lead, image-text contrastive learning has become a prevalent strategy for VLM pretraining [Liu et al., 2023, Cai et al., 2024, Liu et al., 2024a, Dai et al., 2023, Zhai et al., 2022b, Chen et al., 2022, Beyer et al., 2024, Fini et al., 2023]. Contrastive vision-language pretraining spans numerous downstream applications, including zero-shot image classification [Zhai et al., 2022a, Radford et al., 2021, Metzen et al., 2024, Gao et al., 2021], text-to-image generation [Podell et al., 2023, Abdal et al., 2021, Ramesh et al., 2022, Saharia et al., 2022], as well as assessing text-image alignment [Moens et al., 2021, Cho et al., 2023]. In this work, we are particularly interested in the ability of CLIP-based models to evaluate compositional text-image alignment.

Compositional Understanding Benchmarks. Several benchmarks have been developed to assess the compositional understanding of VLMs. In this work, we focus on benchmarks structured as cross-modal retrieval tasks where the model needs to distinguish between correct and incorrect text descriptions given an image, and evaluations are based on accuracy metrics. The majority of these benchmarks [Zhao et al., 2022, Yuksekgonul et al., 2023a, Parcalabescu et al., 2022] rely on the rule-based construction of negative captions and the generation of their associated image counter-factuals [Zhang et al., 2024a, Awal et al., 2024]. Yet, many of these benchmarks may be solved by leveraging the language prior exclusively [Goyal et al., 2017, Lin et al., 2024], hence disregarding the information from the visual input. To address this, benchmarks such as SugarCrepe [Hsieh et al., 2023a] leverage large language models to generate plausible and linguistically correct hard negatives, and show that previously introduced text-based hard negative strategies are not always effective [Yuksekgonul et al., 2023b] – *e.g.*, when considering attribute and object swaps between textual descriptions. Other benchmarks focus on assessing the VLMs' spatial understanding [Kamath et al., 2023, Yuksekgonul et al., 2023b, Zhang et al., 2024a], and propose to finetune CLIP-based models on data containing a high proportion of spatial relationships since these relationships tend to be under-represented in commonly used pretraining datasets. Interestingly, Kamath et al. [2023] show that even when finetuning with in-domain data containing an over-representation of spatial relationships, state-of-the-art models still exhibit a close to random chance performance. In this work, we test the hypothesis that spatial relationship failures are due to the lack of composition in the similarity score computation used to train CLIP-like models.

Object-centric Binding Inductive Biases. CLIP has been shown [Yuksekgonul et al., 2023a] to be pushed to learn disentangled, bag-of-words-style representations from the contrastive loss and the easily distinguishable negatives typically used for pretraining. Although the learned representations might be effective for objects presented in isolation, they struggle with scenes containing multiple objects [Tang et al., 2023]. For example, consider a simple scene with a green apple and a yellow banana. In this case, the model must maintain and correctly link the attributes ("green", "yellow") to the objects ("apple", "banana"), without mixing the concepts – *e.g.*, "yellow apple" or 'green banana". This exemplifies the importance of devising robust mechanisms within the CLIP architecture and/or training to accurately handle multiple objects, while preventing feature interferences. In this work, we focus on equipping CLIP with object-centric binding inductive biases and take inspiration from the architectures proposed in the unsupervised object-centric visual representation learning literature [Locatello et al., 2020, Wu et al., 2023, Seitzer et al., 2023, Assouel et al., 2022]. Many recent image-only approaches follow a simple inductive bias introduced by slot attention [Locatello et al., 2020], where an image – encoded as a set of input tokens – is soft partitioned into K slots. In particular, attention maps are computed via an **inverted cross attention** mechanism [Wu et al.], where the softmax is applied along the query dimension in order to induce a competition between the slots to explain different groups of input tokens. In this work, we extend these inductive biases to define text-conditioned visual slots from the input image.

3 Method

Our goal is to enhance CLIP-based architectures with object-centric binding and composition capabilities. Our method starts by extracting representations of distinct open-ended objects and relationships in a textual description, as well as representations of patches in an image. Next, a binding module matches the text representation of objects to the relevant image patches, producing a slot-centric representation of the image. Finally, a structured similarity score compares the slot-centric representation with the textual representations of different objects, and leverages the extracted relationships as constraints applied to the visual slots. Our key contributions lie in the design of the *binding module*[2] and the proposal of the *structured similarity score*, which we detail in sections 3.1 and 3.2, respectively. Figure 1 presents an overview of the proposed approach. Our approach relies on a scene-graph representation of the text modality. We assume the parser is given and orthogonal to our approach and discuss the choice of the parsing method in Appendix A.4.

Notation. We denote as $\mathbf{x}$ an image of shape $\mathbb{R}^{h\times w\times 3}$ and as $\bar{\mathbf{x}} = [\bar{\mathbf{x}}^1, ..., \bar{\mathbf{x}}^N] = E_\phi(\mathbf{x}) \in \mathbb{R}^{N\times d}$ its patch-level encoding, where E_ϕ is an image encoder – typically a pre-trained ViT [Dosovitskiy et al., 2020] – N is the number of patches and d the dimensionality of the patch embeddings. We denote as t the text description, or caption, associated with $\mathbf{x}$. We extract a scene graph. For example,

[2]Code for the binding module is given in the Appendix Fig 16.

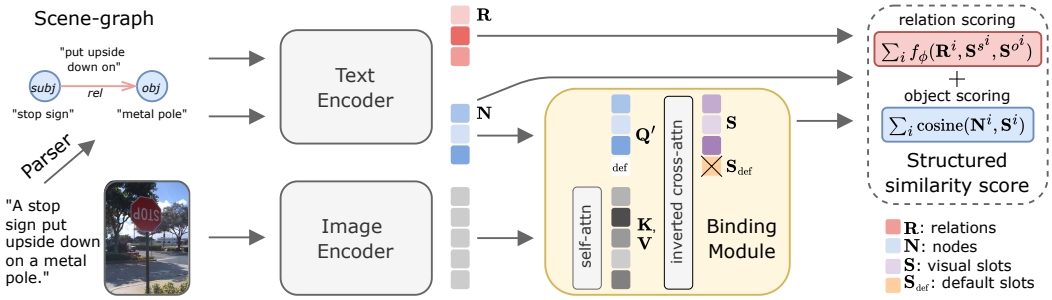


Figure 1: **Object-Centric CLIP (OC-CLIP) overview.** OC-CLIP begins with scene parsing, where we utilize a text parser (*e.g.*, Llama3-based) to extract objects and relations from the input caption. The extracted text objects and relations are then fed into a text encoder, which generates distinct text embeddings for both nodes and relations. In parallel, the corresponding image is processed by an image encoder to produce patch-level image embeddings. These image embeddings are then combined with the text entity embeddings and passed through a *binding module*, which outputs visual token slots embeddings. Both modality are aligned in a *new space* using a structured similarity score that matches nodes embeddings to visual slots and models relational constraints between them.

the scene graph of "A red apple to the left of a blue car" will be represented with the set of nodes {"red apple", "blue car"} and the set of edges {("to the left of", "red apple", "blue car")}. In practice, we represent $\mathcal{N}$ as a matrix of node features $\mathbf{N}$, where each row contains the embedding of a node in the graph. Moreover, we represent each s^i and o^i in the relationship tuples as indices referencing the nodes (rows) in $\mathbf{N}$.

3.1 Binding Module

Our first contribution resides in the binding module. The idea is that when comparing the content of a caption and an image we do not want the features of different objects to interfere with each other but rather keep them separate at a representational level. The role of the binding module is thus to extract a slot-centric representation of an image where the content of the slots are pushed to represent the nodes of the associated scene graph.

To do so, we implement the binding module using a *inverted* cross-attention layer [Wu et al.], where the queries are the nodes from our scene graph and the keys and values are the image patches. We normalize the attention coefficients over the queries' dimension in order to introduce a competition between queries to explain different parts of the visual input. We follow common practice and set the attention's softmax temperature to $\sqrt{D}$, with D being the dimensionality of the dot-product operation. Applying the softmax along the queries' dimension pushes all the candidate keys to be softly matched to at least one query. However, captions mostly describe specific parts of the image, and rarely capture all the visual information. Since we want only the relevant visual information to be captured by the queries, we add a set of default query tokens, stored in a matrix $\mathbf{Q}_{\text{default}}$, which participate in the competitive attention mechanism – with the goal of absorbing the visual information not captured in the caption. These default query tokens are dropped in the subsequent computation steps of our model (akin to registers in ViT backbones [Darcet et al., 2024]). We find the default query tokens crucial to stabilize the training our model.

The binding module computations are formalized as follows:

$$
\begin{aligned}
\mathbf{Q} &= \mathbf{W}_q \mathbf{N}, \\
\mathbf{K}, \mathbf{V} &= \mathbf{W}_k \bar{\mathbf{x}}, \mathbf{W}_v \bar{\mathbf{x}}, \\
\mathbf{Q}' &= [\mathbf{Q}; \mathbf{Q}_{\text{default}}], \\
\text{Attn}(\mathbf{Q}', \mathbf{K}, \mathbf{V}) &= \text{softmax}\left(\frac{\mathbf{Q}' \cdot \mathbf{K}^T}{\sqrt{D}}, \text{dim='Q'}\right) \cdot \mathbf{V}, \\
\mathbf{S}, \mathbf{S}_{\text{default}} &= \text{Attn}(\mathbf{Q}', \mathbf{K}, \mathbf{V}).
\end{aligned} \tag{1}
$$

Here, $\mathbf{W}_q$, $\mathbf{W}_k$, and $\mathbf{W}_v$ are the linear projection weight matrices for the queries, keys, and values, respectively, $\mathbf{S}$ are the visual slots, $\mathbf{S}_{\text{default}}$ are the visual slots from default query tokens, which are discarded for subsequent steps, and [.] denotes the concatenation operation.

Thus, the output of this binding module are the visual slots $\mathbf{S}$. Intuitively, these slots are pushed to represent the visual objects, or entities, that correspond to the nodes of the scene graph. Their object-centric learning is driven by the structured similarity that we detail in the next section.

3.2 Structured similarity score

Our second contribution resides in the introduction of a structured similarity score, whose goal is to promote the constraints imposed by the scene graph on the learnable visual slots. Our proposed structured similarity score is composed of an *object scoring* function and a *relationship scoring* function. The object scoring function assesses the presence of each node in the scene graph (objects present in the caption). We model this function as the sum of the cosine similarity between each textual node representation $\mathbf{N}^i$ and its assigned visual slot $\mathbf{S}^i$. The relationship scoring function encourages the relational constraints imposed by each edge in the scene graph and is defined as a learnable function f_ϕ of the relationship embedding $\mathbf{r}^i$, and the visual slot representations $\mathbf{S}^{s^i}$ and $\mathbf{S}^{o^i}$ corresponding to the subject and object of the relationship, respectively. We derive the overall structured similarity score over the visual slots $\mathbf{S}$ from an image $\mathbf{x}$ and a graph $\mathcal{G} = (\{N^i\}_{i=1..M}, \{(\mathbf{r}^i, s^i, o^i)\}_{i=1..P})$ such that: $S(\mathbf{x}, \mathcal{G}) = \frac{\alpha \sum_{i=1..M} \text{cosine}(\mathbf{N}^i, \mathbf{S}^i) + \beta \sum_{i=1..P} f_\phi(\mathbf{r}^i, \mathbf{S}^{s^i}, \mathbf{S}^{o^i})}{\alpha M + \beta P}$ where α and β are learned parameters controlling the strength of each score. M and P are the number of nodes and relationships in the scene graph $\mathcal{G}$, respectively.

We define f_ϕ as follows:

$$f_\phi(\mathbf{r}, \mathbf{S}^s, \mathbf{S}^o) = \text{cosine}\left(\mathbf{r}, f_s([\mathbf{r}, \mathbf{S}^s]) + f_o([\mathbf{r}, \mathbf{S}^o])\right), \tag{2}$$

where [.] denotes the concatenation of two vectors and f_s and f_o are MLPs that reduce the dimensionality of their inputs. Note that we model the relationship scoring function so that it keeps the same scale as the object scoring function and can take the order of the relationship into account.

3.3 Training

The model is trained using the following loss $\mathcal{L} = \mathcal{L}_{itc} + \mathcal{L}_{rel}$ where $\mathcal{L}_{itc}$ is the image-text contrastive loss defined to minimize the distance between image and scene graph representations from paired text-image data while maximizing the distance between image and scene graph representations from unpaired text-image data as:

$$\mathcal{L}_{itc} = -\sum_{i=1}^{B} \left(\log \frac{\exp^{S(\mathbf{x}_i, \mathcal{G}_i)}}{\sum_{j=1}^{B} \exp^{S(\mathbf{x}_j, \mathcal{G}_i)}} + \log \frac{\exp^{S(\mathbf{x}_i, \mathcal{G}_i)}}{\sum_{j=1}^{B} \exp^{S(\mathbf{x}_i, \mathcal{G}_j)}} \right), \tag{3}$$

where B is the number of elements in the batch. Note that the S is the structured similarity score defined in Eq. 3.2. $\mathcal{L}_{rel}$ is the loss that pushes the model to learn a non-symmetric relationship scores:

$$\mathcal{L}_{rel} = -\sum_{i=1}^{B} \log \frac{\exp^{S(\mathbf{x}_i, \mathcal{G}_i)}}{\exp^{S(\mathbf{x}_i, \mathcal{G}_i)} + \exp^{S(\mathbf{x}_i, \bar{\mathcal{G}}_i)} + \exp^{S(\mathbf{x}_i, \tilde{\mathcal{G}}_i)}}, \tag{4}$$

where $\bar{\mathcal{G}}$ and $\tilde{\mathcal{G}}$ are altered scene graphs. In $\bar{\mathcal{G}}$, we swap the order of the subject and the object of a relationship, whereas in $\tilde{\mathcal{G}}$, we randomly chose the relationship's subject and object from the nodes in the scene graph. We ablate the main components of OC-CLIP in Appendix A.2

4 Results

We evaluate OC-CLIP's inductive biases in 3 different settings:

- **Addressing CLIP's binding problem.** We show the efficiency of OC-CLIP in addressing the binding problem compared to finegrained hard-negative based augmentation on a synthetic dataset.(Section 4.1).
- **Compositional understanding.** We showcase OC-CLIP's compositional understanding, in domain, on real-world object-centric attribute binding and spatial relationship understanding benchmarks (Section 4.2).

- **Scaling on noisy data.** We show that OC-CLIP consistently outperforms a CLIP-based model in both zero-shot single object classification and zero-shot compositional understanding multi-object text retrieval, when training both models *fully* from scratch on larger-scale and noisy dataset (Section 4.3).

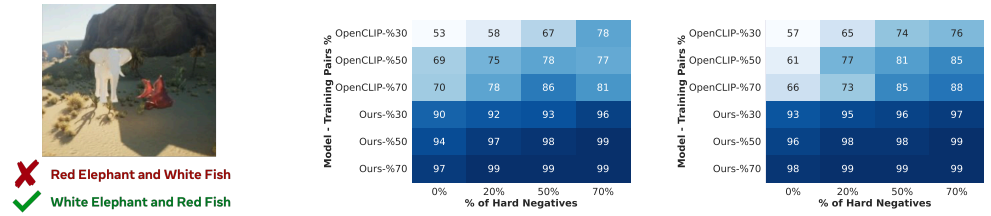
4.1 Addressing CLIP's bag-of-words behavior

In this section, we aim to assess the efficiency and effectiveness of leveraging *finegrained* hard-negatives (swap attributes negatives) OC-CLIP and CLIP-like models in addressing the binding problem. To do so, we use a synthetic dataset with a closed-set vocabulary, from which we can enumerate all possible *object-attribute* conjunctions and systematically evaluate the potential of CLIP-like models and OC-CLIP in addressing *simple* swap-attribute retrieval tasks under varying hard-negative sample sizes.

Dataset. We consider a controlled 3D environment based on PUG [Bordes et al., 2023] and build a dataset composed of a single textured animal, or pairs of animals, in different backgrounds. We use a combination of 4 textures, 20 animal classes, and 5 different backgrounds – *e.g.*, see example in Figure 2a. We follow prior benchmarks [Hsieh et al., 2023a] and perform a text-retrieval task between the correct caption and the associated negative caption. We give additional details about the subsets compositions in Appendix A.6.

Baseline and OC-CLIP training. We train models on data splits from our synthetic data, while considering an increasing proportion of hard-negative samples. We consider a CLIP model initialized with OpenCLIP weights Ilharco et al. [2021]. We also initialize OC-CLIP's text and vision backbones with OpenCLIP weights, but train OC-CLIP's binding module from scratch.

Results. Our results, presented in Figures 2b and 2c, show that simply adding more hard-negatives to OpenCLIP's training plateaus and is not sample-efficient, as the swap-attribute binding performance always underperforms OC-CLIP trained on less data without any targeted hard-negatives in a simple object-attribute binding task. On seen object pairs, with 70% of the possible pairs and 70% of their corresponding swap-attribute hard-negatives CLIP plateaus at 81% compared to OC-CLIP which solves the task at 97% on the same training data size and *no* swap attributes hard-negatives. We hypothesize that the root cause of this issue lies in the representation format used in CLIP's original formulation, which relies on a single vector to capture complex semantic relationships. Our proposed method introduces inductive biases that allow the model to learn more structured representations, avoiding superposition of features [Greff et al., 2020b] and effectively mitigating the bag-of-words behavior.



(a) Synthetic data example (b) Seen object pairs (c) Unseen object pairs

Figure 2: **Efficiency and effectiveness of OC-CLIP: Analysis on synthetic data.** Performance of the finetuned OpenCLIP and OC-CLIP models on a binary classification task between a caption and its corresponding hard-negative given a synthetic image, as shown in (a). Performance is shown as a function of the percentage of animal pairs (y-axis) seen during training and the proportion of hard-negatives used in the training data (x-axis). Results shown for (a) seen and (b) unseen object pairs.

4.2 Compositional Understanding

In this section, we verify that the observations made in the controlled environment presented in Section 4.1 also transfer to real-word datasets, thereby assessing the real-world compositional understanding of OC-CLIP.

Datasets. We train OC-CLIP and finetune OpenCLIP *in-domain* on a set of datasets relevant for real-world compositional understanding. The training text descriptions representing positive samples are taken from COCO [Lin et al., 2014], Visual-Genome (VG) [Krishna et al., 2017]

and GQA [Hudson and Manning, 2019]. The latter annotates images coming from Visual Genome [Krishna et al., 2017] with objects and both spatial and non-spatial relationships, and thus contains a high representation of spatial prepositions. We evaluate the different models on the most challenging benchmarks representative of compositional understanding, ensuring that we validate both their *attribute binding* and *spatial relationship* understanding capabilities. In particular, we use SugarCrepe [Hsieh et al., 2023b] and ARO-A [Yuksekgonul et al., 2023a] for attribute binding and ARO-Relation (ARO-R) [Yuksekgonul et al., 2023a], COCO-spatial and GQA-spatial [Kamath et al., 2023] for spatial relationship understanding. Although Hsieh et al. [2023b] showed that other benchmarks such as VL-Checklist [Zhao et al., 2023], COCO-Order and Flickr-Order splits of ARO [Yuksekgonul et al., 2023a] were easily hackable because the negatives are not semantically correct, we include the results on those benchmarks for reference in Appendix A.5.

Training. As in section 4.1, we initialize the text and vision backbones of OC-CLIP with pre-trained model weights, and train the binding module of OC-CLIP from scratch. In particular, we initialize the text backbone with OpenCLIP weights [Ilharco et al., 2021] and consider two different vision backbones, OpenCLIP (ViT-B-16) [Ilharco et al., 2021] and DinoV2 (ViT-B-14) [Oquab et al., 2024], to show the flexibility of our binding module and learned structured similarity score. We noticed that taking the patches from earlier layers in OpenCLIP helps the training and ablate it in Appendix A.2. We use a batch size of 128 and a learning rate of $2 \cdot 10^{-4}$ to train OC-CLIP for 100 epochs. We use a batch size of 256 – following previous finetuning approaches [Kamath et al., 2023, Yuksekgonul et al., 2023b] – and a learning rate of $4 \cdot 10^{-6}$ for 20 epochs to finetune the OpenCLIP baseline. We run all the models for 3 seeds and report the mean performance along with their standard deviation. Note that since OC-CLIP's binding module is trained from scratch, OC-CLIP's learned vision-language-aligned space does not rely on the vision-language alignment captured by the CLS token of OpenCLIP's backbone (in fact, we drop the CLS token). Therefore, **the new representation space learned by OC-CLIP can only be expected to generalize within the vocabulary it has been trained on.** [3]

Baselines. We report the performance of a representative set of strong baselines which we separate in two groups: the first group of baselines are models trained contrastively and finetuned in-domain (on COCO/VG) and the second group are hard-negative-based and recaptioning-based methods, further divided into small scale and large scale. For the first group, we include OpenCLIP – referred to as OpenCLIP-FT –, BLIP [Li et al., 2023a], and XVLM [Zeng et al., 2022]. BLIP is augmented with an image-text matching loss and XVLM uses bounding boxes to assist the object-centric binding. Note that these two baselines are also equipped with a language modeling objective which may help identify unplausible captions. For the second group, we select methods that augment the dataset with rule-based text hard-negatives (NegCLIP [Yuksekgonul et al., 2023b]), language-model-based hard-negatives (CE-CLIP Zhang et al. [2020] and CLIP-SVLC [Doveh et al., 2023b]), and image-&-language-model-based hard-negatives (CLIP-CC [Zhang et al., 2024a]). We also include dense recaptioning baselines such as DAC [Doveh et al., 2023a] for reference.

Attribute Binding Results. We evaluate the attribute binding capabilities of OC-CLIP and baselines on SugarCrepe [Hsieh et al., 2023b] and ARO-A [Yuksekgonul et al., 2023b] benchmarks. We report the results in Table 1. When comparing $\text{OpenCLIP}_{\text{FT}}$ to OC-CLIP (ours – both models), we observe notable performance boosts on the hard SugarCrepe's swap-attribute, and swap-object. In particular, $\text{OC-CLIP}_{\text{B-14}}$ achieves improvements of +16.5% on the hard swap attribute split, +20.4% on the swap object split, and a smaller +4.1% on the replacement relationship split. Moreover, both OC-CLIP models perform similarly to $\text{OpenCLIP}_{\text{FT}}$ on the remaining SugarCrepe splits. This is to be expected since the remaining splits do not require precise binding to distinguish between positive and negative captions and may therefore be solved with a bag-of-words-like representation. We additionally compare OC-CLIP to finetuned versions of CLIP that rely on in domain hard-negatives (NegCLIP, CE-CLIP, CC-CLIP, MosaiCLIP) and with dense recaptioning (DAC-LLM and DAC-SAM). In particular DAC finetunes OpenCLIP with $\sim$ 3M VLM-generated dense captions (along with their corresponding hard negatives) that significantly increase the vocabulary coverage compared to methods that only finetune in domain (*e.g.*, on COCO). Interestingly, OC-CLIP still outperforms them on both swap-attribute and swap-object, showing improvements of +13.6% and +8.4% over the second best performing method, respectively. Those results confirm the behavior that we observed

[3] **Important note** : Although OC-CLIP is compared in domain against post-training baselines of CLIP, our approach is a pretraining strategy and can only be compared in domain. We discuss this point further in the Appendix A.1

in Section 4.1 and the inefficiency of hard-negative methods in solving the binding problem of CLIP-like models, even at the scale of DAC finetuning.

Model	Swap		Add		Replace			ARO
	Object	Attribute	Object	Attribute	Object	Attribute	Relation	Attribution
Zero-shot								
OpenCLIP	68.2	66.2	82.7	80.3	93.8	82.8	67.3	63.2
In-domain ft baselines								
BLIP	66.2	76.2	-	-	**96.5**	81.9	68.35	**88.0**
XVLM	64.9	73.9	-	-	95.2	87.7	77.4	86.8
$\text{OpenCLIP}_{\text{FT-B-16}}$	63.1 ±0.6	72.4±1.1	**93.4** ±0.2	83.1 ±0.5	95.4	87.0 ±0.6	75.5 ±0.6	60.0
Hard-Negative - small scale								
NegCLIP	75.2	75.4	88.8	82.8	92.7	85.9	76.5	71
CE-CLIP	72.8	77	92.4	93.4	93.1	88.8	79	76.4
CC-CLIP	68.6	73.6	86.7	90.3	95.9	87.9	76.2	-
CLIP-SVLC	-	-	-	-	-	-	-	73.0
MosaiCLIP	-	-	-	-	-	-	-	78.0
Hard-Negative/Dense Captioning - large scale								
DAC-LLM	75.1	74.1	89.7	**97.7**	94.4	**89.3**	**84.4**	73.9
DAC-SAM	71.8	75.3	87.5	95.5	91.2	85.9	83.9	70.5
Ours								
OC-CLIP B-32	76.5 ±0.5	85.6 ±0.5	86.5±0.7	85.7±0.7	91.3 ±0.6	88.6 ±0.2	75.4 ±0.3	82.5±0.1
OC-CLIP B-16	76.6 ±0.6	87.5 ±0.5	91.1±0.4	83.8 ±1.0	94.6 ±0.4	87.9 ±0.1	76.0 ±0.4	83.2±0.3
OC-CLIP B-14	**83.5** ±0.2	**88.9** ±0.6	92.8±0.1	84.8 ±0.1	95.9 ±0.4	89.2±0.1	79.6 ±0.3	84.0±0.

Table 1: **Attribute binding: Performance on SugarCrepe and ARO-A.** Both OpenCLIP-FT and OC-CLIP are initialized with the same OpenCLIP checkpoints. OC-CLIP is trained with two ViT base backbones with different resolutions: OpenCLIP's backbone (B-16) and Dinov2 (B-14).OC-CLIP's bidning module is always trained from scratch.

Relationship Understanding Results. We evaluate the spatial relationship understanding capabilities of OC-CLIP and baselines on COCO-spatial, GQA-spatial, and ARO-Relation (ARO-R). Note that ARO-Relation contains both spatial and non-spatial relations but about half of the test examples consists of left/right relationships understanding. We report the results in Table 2 and show consistent improvements of both OC-CLIP models over the baseline models and across the 3 datasets. In particular, the best OC-CLIP model outperforms OpenCLIP-FT by +44.1% on COCO-spatial, +43.6% on GQA-spatial, and +34.8% on ARO-R. When compared to contrastive VLMs finetuned with in-domain data (XVLM, BLIP), OC-CLIP models exhibit superior performance, with improvements between +10% and +27% over the strongest contrastive finetuned VLM. Finally, when compared to baselines leveraging hard-negatives (NegCLIP), OC-CLIP remains the highest performer. Additional results on the ARO benchmark are reported in Table 9 of the Appendix.

4.3 Training OC-CLIP from scratch

In this section, we aim to assess the potential of OC-CLIP when trained *fully* from scratch from scene-graphs obtained from large scale non-human-curated captioning dataset.

Model	COCO-spatial	GQA-spatial	ARO-R
XVLM	73.6	67	73.4
BLIP	56.4	52.6	59
NegCLIP	46.4	46.7	80.2
$\text{OpenCLIP}_{\text{FT}}$	45.6 ±0.2	49.1±1.1	50.1±0.4
OC-CLIP (B-16)	86.3	90.0	84.3
OC-CLIP (B-14)	**89.7**	**92.7**	**84.9**

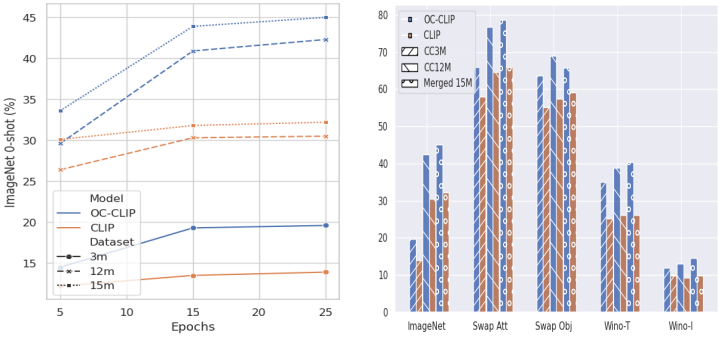
Table 2: **Spatial relationship understanding: Performance on COCO-spatial, GQA-spatial from the Whats'up Benchmark and ARO-R.** We finetune both OpenCLIP ($\text{OpenCLIP}_{\text{FT}}$ here) and OC-CLIP in-domain on COCO, Visual Genome, and GQA data. Both models are initialized with the same OpenCLIP checkpoints.

Datasets. We train both ViT-B-16 OpenCLIP model and OC-CLIP *fully* from scratch on increasingly large dataset sizes using CC3M [Sharma et al., 2018], CC12M [Changpinyo et al., 2021] and the combination of both datasets. We evaluate all models on ImageNet [Deng et al., 2009] zero-shot classification in this section, and report results on the ELEVATER suite [Li et al., 2022] in Appendix (Table 6). We also evaluate zero-shot compositional understanding of the models on the challenging swap-object and swap-attribute splits of SugarCrepe, and on Winoground [Thrush et al., 2022].

Baseline and OC-CLIP training. Both CLIP and OC-CLIP architectures are trained *fully* from scratch for 5, 15, or 25 epochs, using a batch size of 4096, a learning rate of $1 \cdot 10^{-3}$, 2k steps of learning rate warm-up, and a cosine decay. As recommended by Mu et al. [2021], we use AdamW

optimizer with 0.5 of weight decay and β_2 set to 0.98. We use a ViT-B-16 backbone for both models. Since OC-CLIP's text bakcbone only needs to encode single objects and relationships, we use a smaller text bakcbone with a context length of 20 and only 6 layers instead of 12 . Note that we do not tune the hyper-parameters in this experiment. We further discuss those design choices in Appendix A.3.

Results. We start by verifying the sample efficiency of OC-CLIP using ImageNet [Deng et al., 2009] zero-shot classification performance in Figure 3a. We show that OC-CLIP shows better sample-efficiency than the baseline trained on the same data, while using a smaller text backbone. We then evaluate OC-CLIP on zero-shot classification and compositional understanding in Figure 3b. Interestingly, OC-CLIP shows performance gains in general zero-shot classification ($+12.8\%$ on ImageNet, when trained from scratch on CC3M+CC12M) while also showcasing substantial improvements in zero-shot compositional understanding. For example, OC-CLIP exhibits a notable $+12.7\%$ and $+6.6\%$ in SugarCrepe's swap-attribute and swap-object splits, respectively. This experiment shows that the structured training of OC-CLIP is also effective when scaling to noisy image-caption dataset and, therefore, does not solely rely on high-quality human captions. We additionally report extensive zero-shot downstream classification performance on the ELEVATER [Li et al., 2022] suite and discuss the computation trade-off of our approach in Appendix A.3. We leave further scaling for future work.



(a) Sample efficiency (b) Zero-shot accuracy

Figure 3: **Scaling the training on noisy data.** CLIP and OC-CLIP are trained *from scratch* on varying sizes of data (3M, 12M and 15M) for a varying number of epochs. OC-CLIP shows (b) better zero-shot compositional understanding performance on SugarCrepe's swap-attribute and swap-object, and on Winoground (I = Image score and T = Text score), as well as (a) better sample efficiency shown on zero-shot ImageNet classification.

5 Conclusion and limitations

Conclusion. In this paper, we proposed Object-Centric CLIP (OC-CLIP), a pretraining method to enhance the compositional scene understanding of CLIP-like models by leveraging advances from object-centric representation learning. Our approach adapts the slot-centric representation paradigm to CLIP and dynamically aligns each representational slot with the objects mentioned in the text description. This is achieved by the introduction of a binding module and a structured similarity score that allows to train OC-CLIP in a contrastive way. We evaluated our approach against finegrained hard-negative augmentation strategies and demonstrated that OC-CLIP significantly enhances the binding of object-centric attributes and spatial relationships across a representative set of challenging real-world compositional image-text matching benchmarks.Finally, we show the scaling potential of OC-CLIP to be trained from scratch on a noisy dataset [Changpinyo et al., 2021, Sharma et al., 2018]. Notably, we report performance gains in zero-shot classification ($+12.8\%$ in ImageNet 6) while maintaining substantial improvements in zero-shot SugarCrepe swap-attribute ($+12.7\%$) and swap-object ($+6.6\%$) splits.

Limitations and Future Work. Our OC-CLIP model has several limitations and avenues for future work. Notably, our approach relies on a parser to extract object-centric attributes and spatial relationships from text descriptions. While we have chosen an LLM-based parser, which is discussed in Appendix A.4, studying the different biases of LLM-based parser families could be interesting. Additionally, while we show the scaling potential of OC-CLIP at 15M scale (A.1, A.3), the model needs further scaling to be fully comparable to all the CLIP variants, trained at least at 400M scale.

References

Rameen Abdal, Peihao Zhu, John Femiani, Niloy J. Mitra, and Peter Wonka. Clip2stylegan: Unsupervised extraction of stylegan edit directions, 2021. URL https://arxiv.org/abs/2112.05219.

Rim Assouel, Lluis Castrejon, Aaron Courville, Nicolas Ballas, and Yoshua Bengio. VIM: Variational independent modules for video prediction. In Bernhard Schölkopf, Caroline Uhler, and Kun Zhang, editors, *Proceedings of the First Conference on Causal Learning and Reasoning*, volume 177 of *Proceedings of Machine Learning Research*, pages 70–89. PMLR, 11–13 Apr 2022. URL https://proceedings.mlr.press/v177/assouel22a.html.

Rabiul Awal, Saba Ahmadi, Le Zhang, and Aishwarya Agrawal. Vismin: Visual minimal-change understanding, 2024. URL https://arxiv.org/abs/2407.16772.

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, Thomas Unterthiner, Daniel Keysers, Skanda Koppula, Fangyu Liu, Adam Grycner, Alexey Gritsenko, Neil Houlsby, Manoj Kumar, Keran Rong, Julian Eisenschlos, Rishabh Kabra, Matthias Bauer, Matko Bošnjak, Xi Chen, Matthias Minderer, Paul Voigtlaender, Ioana Bica, Ivana Balazevic, Joan Puigcerver, Pinelopi Papalampidi, Olivier Henaff, Xi Xiong, Radu Soricut, Jeremiah Harmsen, and Xiaohua Zhai. Paligemma: A versatile 3b vlm for transfer, 2024. URL https://arxiv.org/abs/2407.07726.

Florian Bordes, Shashank Shekhar, Mark Ibrahim, Diane Bouchacourt, Pascal Vincent, and Ari Morcos. Pug: Photorealistic and semantically controllable synthetic data for representation learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 45020–45054. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/8d352fd0f07fde4a74f9476603b3773b-Paper-Datasets_and_Benchmarks.pdf.

Florian Bordes, Richard Yuanzhe Pang, Anurag Ajay, Alexander C Li, Adrien Bardes, Suzanne Petryk, Oscar Mañas, Zhiqiu Lin, Anas Mahmoud, Bargav Jayaraman, et al. An introduction to vision-language modeling. *arXiv preprint arXiv:2405.17247*, 2024.

Mu Cai, Haotian Liu, Siva Karthik Mustikovela, Gregory P. Meyer, Yuning Chai, Dennis Park, and Yong Jae Lee. Making large multimodal models understand arbitrary visual prompts. In *CVPR 2024*, 2024.

Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12M: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *CVPR*, 2021.

Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, et al. Pali: A jointly-scaled multilingual language-image model. *arXiv preprint arXiv:2209.06794*, 2022.

Jaemin Cho, Seunghyun Yoon, Ajinkya Kale, Franck Dernoncourt, Trung Bui, and Mohit Bansal. Fine-grained image captioning with clip reward, 2023. URL https://arxiv.org/abs/2205.13115.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023. URL https://arxiv.org/abs/2305.06500.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers, 2024. URL https://arxiv.org/abs/2309.16588.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Sivan Doveh, Assaf Arbelle, Sivan Harary, Roei Herzig, Donghyun Kim, Paola Cascante-Bonilla, Amit Alfassy, Rameswar Panda, Raja Giryes, Rogerio Feris, et al. Dense and aligned captions (dac) promote compositional reasoning in vl models. *Advances in Neural Information Processing Systems*, 36:76137–76150, 2023a.

Sivan Doveh, Assaf Arbelle, Sivan Harary, Rameswar Panda, Roei Herzig, Eli Schwartz, Donghyun Kim, Raja Giryes, Rogerio Feris, Shimon Ullman, and Leonid Karlinsky. Teaching structured vision&language concepts to vision&language models, 2023b. URL https://arxiv.org/abs/2211.11733.

Gamaleldin Elsayed, Aravindh Mahendran, Sjoerd van Steenkiste, Klaus Greff, Michael C Mozer, and Thomas Kipf. Savi++: Towards end-to-end object-centric learning from real-world videos. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 28940–28954. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/ba1a6ba05319e410f0673f8477a871e3-Paper-Conference.pdf.

S. M. Ali Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, Koray Kavukcuoglu, and Geoffrey E. Hinton. Attend, infer, repeat: Fast scene understanding with generative models, 2016. URL https://arxiv.org/abs/1603.08575.

Enrico Fini, Pietro Astolfi, Adriana Romero-Soriano, Jakob Verbeek, and Michal Drozdzal. Improved baselines for vision-language pre-training. *Transactions on Machine Learning Research*, 2023.

Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters, 2021. URL https://arxiv.org/abs/2110.04544.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6904–6913, 2017.

Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Chris Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference, 2020a. URL https://arxiv.org/abs/1903.00450.

Klaus Greff, Sjoerd van Steenkiste, and Jürgen Schmidhuber. On the binding problem in artificial neural networks, 2020b. URL https://arxiv.org/abs/2012.05208.

Matthew Honnibal and Ines Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 2017.

Cheng-Yu Hsieh, Jieyu Zhang, Zixian Ma, Aniruddha Kembhavi, and Ranjay Krishna. Sugarcrepe: Fixing hackable benchmarks for vision-language compositionality. In *NeurIPS 2023*, 2023a.

Cheng-Yu Hsieh, Jieyu Zhang, Zixian Ma, Aniruddha Kembhavi, and Ranjay Krishna. Sugarcrepe: Fixing hackable benchmarks for vision-language compositionality, 2023b. URL https://arxiv.org/abs/2306.14610.

Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709, 2019.

Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, 2021. URL https://doi.org/10.5281/zenodo.5143773.

Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. Hard negative mixing for contrastive learning, 2020. URL https://arxiv.org/abs/2010.01028.

Amita Kamath, Jack Hessel, and Kai-Wei Chang. What's "up" with vision-language models? investigating their struggle with spatial reasoning, 2023. URL https://arxiv.org/abs/2310.19785.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017.

Tiep Le, Vasudev Lal, and Phillip Howard. Coco-counterfactuals: Automatically constructed counterfactual examples for image-text pairs. In *NeurIPS 2023*, 2023.

Chunyuan Li, Haotian Liu, Liunian Harold Li, Pengchuan Zhang, Jyoti Aneja, Jianwei Yang, Ping Jin, Houdong Hu, Zicheng Liu, Yong Jae Lee, and Jianfeng Gao. Elevater: A benchmark and toolkit for evaluating language-augmented visual models, 2022. URL https://arxiv.org/abs/2204.08790.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023a.

Zhuang Li, Yuyang Chai, Terry Yue Zhuo, Lizhen Qu, Gholamreza Haffari, Fei Li, Donghong Ji, and Quan Hung Tran. Factual: A benchmark for faithful and consistent textual scene graph parsing, 2023b. URL https://arxiv.org/abs/2305.17497.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In David J. Fleet, Tomás Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, volume 8693 of *Lecture Notes in Computer Science*, pages 740–755. Springer, 2014. doi: 10.1007/978-3-319-10602-1_48. URL https://doi.org/10.1007/978-3-319-10602-1_48.

Zhiqiu Lin, Xinyue Chen, Deepak Pathak, Pengchuan Zhang, and Deva Ramanan. Revisiting the role of language priors in vision-language models. *arXiv preprint arXiv:2306.01879*, 2024.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *NeurIPS 2023*, 2023.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024a. URL https://llava-vl.github.io/blog/2024-01-30-llava-next/.

Qinying Liu, Wei Wu, Kecheng Zheng, Zhan Tong, Jiawei Liu, Yu Liu, Wei Chen, Zilei Wang, and Yujun Shen. Tagalign: Improving vision-language alignment with multi-tag classification, 2024b. URL https://arxiv.org/abs/2312.14149.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention, 2020. URL https://arxiv.org/abs/2006.15055.

Jan Hendrik Metzen, Piyapat Saranrittichai, and Chaithanya Kumar Mummadi. Autoclip: Auto-tuning zero-shot classifiers for vision-language models, 2024. URL https://arxiv.org/abs/2309.16414.

Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors. *CLIPScore: A Reference-free Evaluation Metric for Image Captioning*, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.595. URL https://aclanthology.org/2021.emnlp-main.595.

Shanka Subhra Mondal, Jonathan D. Cohen, and Taylor W. Webb. Slot abstractors: Toward scalable abstract visual reasoning, 2024. URL https://arxiv.org/abs/2403.03458.

Norman Mu, Alexander Kirillov, David Wagner, and Saining Xie. Slip: Self-supervision meets language-image pre-training, 2021. URL https://arxiv.org/abs/2112.12750.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2024. URL https://arxiv.org/abs/2304.07193.

Roni Paiss, Ariel Ephrat, Omer Tov, Shiran Zada, Inbar Mosseri, Michal Irani, and Tali Dekel. Teaching clip to count to ten. In *ICCV 2023*, 2023.

Letitia Parcalabescu, Michele Cafagna, Lilitta Muradjan, Anette Frank, Iacer Calixto, and Albert Gatt. Valse: A task-independent benchmark for vision and language models centered on linguistic phenomena. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, page 8253–8280. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.acl-long.567. URL http://dx.doi.org/10.18653/v1/2022.acl-long.567.

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, 2023. URL https://arxiv.org/abs/2307.01952.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. Laion-5b: An open large-scale dataset for training next generation image-text models, 2022. URL https://arxiv.org/abs/2210.08402.

Maximilian Seitzer, Max Horn, Andrii Zadaianchuk, Dominik Zietlow, Tianjun Xiao, Carl-Johann Simon-Gabriel, Tong He, Zheng Zhang, Bernhard Schölkopf, Thomas Brox, and Francesco Locatello. Bridging the gap to real-world object-centric learning, 2023. URL https://arxiv.org/abs/2209.14860.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1238. URL https://aclanthology.org/P18-1238/.

Yingtian Tang, Yutaro Yamada, Yoyo Minzhi Zhang, and Ilker Yildirim. When are lemons purple? the concept association bias of vision-language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://openreview.net/forum?id=5sGLPiG1vE.

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*, pages 5238–5248, 2022.

Taylor Webb, Shanka Subhra Mondal, and Jonathan D Cohen. Systematic visual reasoning through object-centric relational abstraction. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 72030–72043. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/e3cdc587873dd1d00ac78f0c1f9aa60c-Paper-Conference.pdf.

Yi-Fu Wu, Klaus Greff, Google Deepmind, Gamaleldin F. Elsayed, Michael C. Mozer, Thomas Kipf, and Sjoerd van Steenkiste. Inverted-attention transformers can learn object representations: Insights from slot attention. URL https://api.semanticscholar.org/CorpusID:266090680.

Ziyi Wu, Jingyu Hu, Wuyue Lu, Igor Gilitschenski, and Animesh Garg. Slotdiffusion: Object-centric generative modeling with diffusion models, 2023. URL https://arxiv.org/abs/2305.11281.

Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang. Groupvit: Semantic segmentation emerges from text supervision, 2022. URL https://arxiv.org/abs/2202.11094.

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?, 2023a. URL https://arxiv.org/abs/2210.01936.

Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? In *International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=KRLUvxh8uaX.

Yan Zeng, Xinsong Zhang, and Hang Li. Multi-grained vision language pre-training: Aligning texts with visual concepts, 2022. URL https://arxiv.org/abs/2111.08276.

Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18123–18133, 2022a.

Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning, 2022b. URL https://arxiv.org/abs/2111.07991.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training, 2023. URL https://arxiv.org/abs/2303.15343.

Jianrui Zhang, Mu Cai, Tengyang Xie, and Yong Jae Lee. Countercurate: Enhancing physical and semantic visio-linguistic compositional reasoning via counterfactual examples, 2024a. URL https://arxiv.org/abs/2402.13254.

Le Zhang, Rabiul Awal, and Aishwarya Agrawal. Contrasting intra-modal and ranking cross-modal hard negatives to enhance visio-linguistic compositional understanding, 2024b. URL https://arxiv.org/abs/2306.08832.

Yuhao Zhang, Hang Jiang, Yasuhide Miura, Christopher D Manning, and Curtis P Langlotz. Contrastive learning of medical visual representations from paired images and text. *arXiv preprint arXiv:2010.00747*, 2020.

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.

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, 2023. URL https://arxiv.org/abs/2207.00221.

Chenhao Zheng, Jieyu Zhang, Aniruddha Kembhavi, and Ranjay Krishna. Iterated learning improves compositionality in large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13785–13795, 2024.

A Appendix

Impact Statement. This paper presents work whose goal is to advance the field of machine learning, specifically the multi-modal architecture of CLIP. CLIP itself has broad applications and impact thus our proposed technique shall be considered when such models are used.

A.1 OC-CLIP as a Pretraining alternative to CLIP

Our method is not intended to be a post-training method –but rather a pre-training strategy showcasing the benefits of object-centric inductive biases in the context of contrastive pretraining. When using our binding module in the in-domain setting (Section 4.2), we do not expect to maintain CLIP's zero-shot performance. As our binding and relationship modules are trained from scratch, they do not rely on the previously aligned CLS tokens (in fact, we drop the vision CLS token completely). As a result, our model can only be expected to work well within the vocabulary domain it is exposed to when tested for its zero-shot performance.

Given that we only inherit the vision tokens from the CLIP vision backbone, we believe that it's still worthwhile to check whether our training is detrimental to the inherited vision tokens. We probe this by computing linear evaluations of the pre-trained CLIP-based vision backbone vs the vision backbone after our OC-CLIP training strategy on COCO/VG. To do so, we average-pool the patch-tokens that serve as input to the binding module and train a linear probe on commonly used image classification datasets. We report those results in Table 4 and observe that our training strategy did not significantly impact the generalizability of the backbone despite only training on COCO/VG. In order to further motivate our approach as a good pretraining alternative, we also compare to additional baselines trained from scratch on CC3M/CC12M specifically designed to address the lack of compositional generalization in CLIP models. Specifically, we compare to NegCLIP [Yuksekgonul et al., 2023b], which augments CLIP training with rule-based hard negatives and IL-CLIP [Zheng et al., 2024], a novel iterated learning method for contrastive learning, which improves compositional understanding. Our results in Table A.1 show the effectiveness of our proposed inductive biases as an effective alternative to CLIP training.

Training Data	Model	SC-add	SC-replace	SC-swap	Wino (T2I)	Imagenet
CC3M	OpenCLIP-3M	61.9	64.3	52.9	8.1	13.7
	NegCLIP	59.3	59.2	60.1	11.8	11.8
	IL-CLIP	66.1	67.0	54.5	**13.3**	14.2
	OC-CLIP	**74.6**	**73.0**	**64.7**	12	**19.6**
CC12M	OpenCLIP-12M	67.5	70.0	60.2	7.2	31.4
	NegCLIP	65.0	70.2	67.2	7.3	28.9
	IL-CLIP	73.8	73.0	62.9	10.1	32.8
	OC-CLIP	**81.7**	**79.8**	**72.9**	**13.0**	**42.4**

Table 3: Training from scratch on CC3M/CC12M comparison of OC-CLIP with OpenCLIP, IL-CLIP, and NegCLIP. Comparison is done on SugarCrepe [Hsieh et al., 2023b] swap splits, Winoground [Thrush et al., 2022] T2I score and Imagenet [Deng et al., 2009]

	Food-101	CIFAR-10	CIFAR-100	Cars	Aircraft	DTD	Pets	FER-2013	STL-10	EuroSAT	RESISC45	GTSRB	Country211	UCF101 Frames	Caltech	MNIST	ImageNet
OpenCLIP	**87.1**	95.8	83.3	63.9	**30.1**	**80.3**	82.8	65.6	97.8	95.9	93.9	**88.3**	**20.1**	79.5	**91.2**	99.2	73.1
OC-CLIP	87.0	**96.0**	**83.7**	**64.0**	30.0	79.8	82.8	65.6	**97.9**	**96.0**	93.9	88.0	20.0	**79.6**	91.1	**99.3**	**73.2**

Table 4: Linear average pooling of CLIP vs OC-CLIP. CLIP is pretrained on Laion400M and OC-CLIP is trained on COCO/VG.

A.2 Ablations

In Table 5 we ablate the key design choices of our model. Specifically, we investigate two key components of the model: the use of competitive (inverted) cross-attention and the local graph contrastive loss. On the one hand, results show that removing the competitive cross-attention mechanism greatly affects fine-grained attribute binding (decreasing from 89.0 to 85.9). On the other hand, removing the local graph contrastive loss significantly impacts downstream relational understanding, with accuracy decreasing from 80.5 to 72.8. Adding attention layers helps relational understanding (boosting performance from 77.6 to 80.5), while adding more default tokens does not necessarily help with attribute binding. These findings highlight the importance of the main design choices behind OC-CLIP.

LOC LOSS	COMP. X-ATT	ATTN LAY	DEFAULT	REL	ATT
✓	✓	✓	1	80.5	89.0
✓	✓	✓	4	79.2	87.6
-	✓	✓	1	72.8	87.7
✓	-	✓	-	78.3	85.9
✓	✓	-	1	77.6	87.8

Table 5: **Ablation of OC-CLIP's main components.** Fine-grained accuracy on attribute binding and relational splits of SugarCrepe

We then further discuss some important design choice of OC-CLIP. Notably :

- The **similarity score** coefficients α and β that control the weight of the objects and relations in the global graph-image similarity score.
- **Binding module inductive biases** and their impact on compositional understanding performance.
- **Local Loss** impact on downstream compositional understanding of relationships.
- **Layer selection** with OpenCLIP backbone.

Similarity Score OC-CLIP's structured global similarity score is a combination of the object and relationship components respectively weighted by two learnt parameters α and β balancing the different contributions. We let the model learn those parameters throughout the training. However, during preliminary experiments we tested a different combinations of initial coefficient within the [1.5, 1, 0.5, 0.1] grid and noticed that the model was always converging to a $\frac{\alpha}{\beta} \sim 3$ without any difference in the downstream compositional performance. We thus fix the initial coefficients to $\alpha = 1.5$ and $\beta = 0.5$ and treat them as parameters.

Default Token and Competitive Cross Attention In the binding module we propose to use an inductive biases to encourage the query tokens to attend to different groups of patches. In order to do so we use a competitive attention mechanism, the so called inverted cross attention common to many object-centric image encoder architecture [Locatello et al., 2020, Wu et al.]. We found that the use of inverted cross attention impacts slightly the fine-grained attribute binning performance (see ATT performance in Table 5), -Comp Att model does not use any inverted cross attention and is rather implemented with a regular cross attention mechanism, the softmax being done along the keys dimensions.). The finegrained attribute understanding (ATT) is affected by the absence of competitive attention between query slots going from 89.0% to 85.9% accuracy.

Local Graph Contrastive Loss In designing the structured similarity score of OC-CLIP the relational component is formulated as the following cosine similarity $f_\phi(\mathbf{r}, \mathbf{S}^s, \mathbf{S}^o) = \text{cosine}(\mathbf{r}, f_s([\mathbf{r}, \mathbf{S}^s]) + f_o([\mathbf{r}, \mathbf{S}^o])$. In theory both $f_s([\mathbf{r}, \mathbf{S}^s])$ and $f_o([\mathbf{r}, \mathbf{S}^o])$ can collapse to ignore the subject object visual representation. In order to prevent such collapse we propose to add a local graph contrastive loss that shares similarity with hard-negative based learning. We enforce the model to model with a higher similarity the graph composed of the same nodes but with either swapped object and subject indices or shuffle objects and subjects indices within the local graph. In both of those cases the relation component of the structured similarity score becomes (for a single relation graph) :

$$\text{swap } \tilde{G}; \text{cosine}(\mathbf{r}, f_s([\mathbf{r}, \mathbf{S}^s]) + f_o([\mathbf{r}, \mathbf{S}^o]) \tag{5}$$

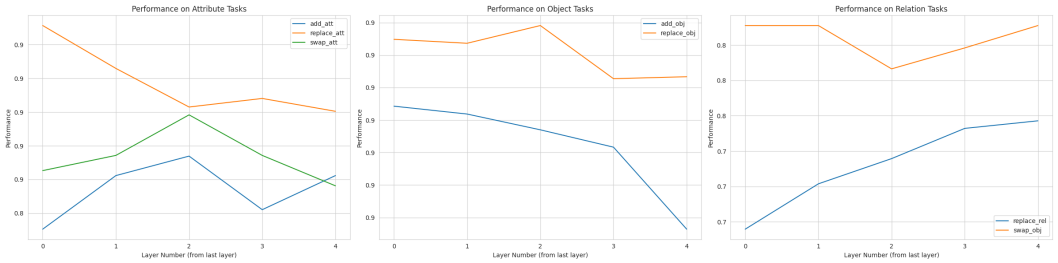


Figure 4: ViT features layer ablation.

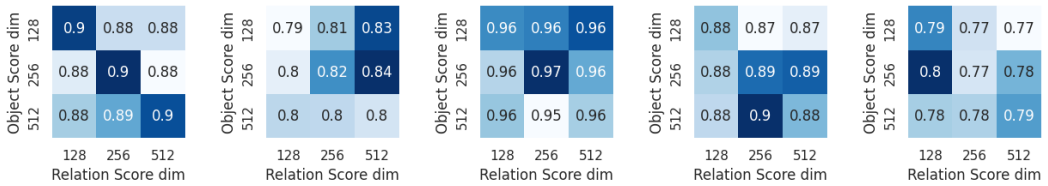
$$\text{swap } \tilde{G}; \text{cosine}(\mathbf{r}, f_s([\mathbf{r}, \mathbf{S}^o]) + f_o([\mathbf{r}, \mathbf{S}^s]) \tag{6}$$

$$\text{shuffle } \bar{G}; \text{cosine}(\mathbf{r}, f_s([\mathbf{r}, \mathbf{S}^{j!=s}]) + f_o([\mathbf{r}, \mathbf{S}^{i!=o}]) \tag{7}$$

This prevents the model from collapsing because ground-truth G is distinguishable from $\tilde{G}$ and $\bar{G}$ only if the visual representations are not ignored in the relationships components. As shown in Table 5, removing the local loss effectively impacts downstream relational understanding on SugarCrepe with a REL accuracy decreasing from 80.5 to 72.8 hence showing the effectiveness of the local graph contrastive loss.

Scoring dimensionality Our structured similarity score allows the text encoder to focus on encoding information about individual objects and their relationships, rather than the entire scene configuration. To achieve this, we experimented with different dimensionality for both the object scoring bottleneck and the relationship scoring bottleneck. Specifically, each of these scores is designed as a cosine distance between a text representation and a visual component (as described in Section 3.2), with each operating at a bottleneck dimension of d_{obj} and d_{rel}. In contrast, OpenCLIP represents both the scene caption and the visual representation at a shared dimension of $d = 512$. We expect that our model can operate effectively at a much lower dimensionality, as it requires less capacity to encode single objects and relationships. We present an ablation study of these two dimensions in Figure 5 and notice that our model is quite robust when we operate on lower dimensional space (eg. 128). We use this results to scale our experiments and train the model from scratch with a smaller text encoder as explained in the next section A.3 corresponding to experiments shown in Section 4.3.

Layer Selection Previous work focusing on dense segmentation tasks[Xu et al., 2022] show that taking features from earlier layer in CLIP's ViT help with fine-grained tasks. Here we ablate OC-CLIP's version using the OpenCLIP pretrained backbone when inserting the binding module after layer $L - k$ for $k \in [0..4]$, 0 being the last layer. Results on merged Sugarcrepe splits are given in Figure 4. The object related queries seem to decrease as a function of k where as the replace rel split is increasing with k. To that end, we actually insert our binding module after layer $L - 2$, where L is the index of the last transformer layer of the ViT.



(a) Swap Att (b) Swap Obj (c) Replace Obj (d) Replace Att (e) Replace Rel

Figure 5: **Score dimensionality ablations** In this ablations we keep the initialization seed fixed and vary the dimensionality of the relation score d_{rel} (x-axis) and object score d_{obj}(y-axis) and report the performance on the swap and replace splits of sugarcrepe.

A.3 Experiments on CC3M/12M.

In the compositional understanding experiments we compare our approach with data-centric finetuning methods that do not add any additional parameters. These methods are expected to retain some of

	Food-101	CIFAR-10	CIFAR-100	SUN397	Cars	Aircraft	DTD	Pets	FER-2013	STL-10	EuroSAT	RESISC45	GTSRB	KITTI	Country211	PCAM	UCF101 Frames	CLEVR	HatefulMemes	MNIST	SST2	ImageNet
CLIP (3m)	12.8	44.9	19.9	27.9	1.2	1.3	9.7	12.6	1.3	76.1	15.8	13.5	6.9	24.9	0.6	44.3	17.8	11.1	51.2	7.8	47.4	13.9
OC-CLIP (3m)	15.3	57.1	24.8	33.1	1.2	1.6	13.3	16.3	9.9	82.0	16.2	22.0	4.5	30.8	0.7	55.6	22.5	12.5	52.0	11.6	50.1	**19.2**
CLIP (12m)	42.7	63.2	30.2	42.7	15.5	3.1	14.5	52.6	13.1	85.7	12.3	28.5	8.3	34.9	3.9	54.4	33.9	11.8	51.4	11.2	51.9	30.5
OC-CLIP (12m)	54.1	74.5	44.6	51.4	21.1	3.7	19.4	66.4	7.4	91.9	31.8	40.6	8.1	41.9	5.9	56.1	45.7	12.7	49.2	9.7	50.2	**42.3**
CLIP (15m)	43.4	72.3	33.8	44.2	15.8	2.3	14.0	53.8	9.2	89.1	24.0	30.0	11.0	28.0	3.3	50.1	37.4	12.5	50.6	10.2	50.1	32.2
OC-CLIP (15m)	54.5	82.3	46.6	54.1	20.2	3.7	22.1	69.2	5.2	94.2	26.8	44.4	9.4	29.9	5.9	52.9	47.3	14.5	51.3	9.0	49.9	**45.0**

Table 6: Zero-shot evaluation of CLIP vs OC-CLIP. Trained on varying size of data (cc3m, cc12m, merged 15m) for 25 epochs.

the general capabilities of the initial backbone. In contrast, our binding and relationship modules is trained from scratch, which means it may not generalize as well to unseen data and can only be expected to work well within the vocabulary domain it has been exposed to (eg. COCO/VG/GQA in our experiments setting). However an interesting question would be to asses whether such inductive biases and structured similarity object might have some sclaing potential on noisy and non human curated datasets such as CC12M [Changpinyo et al., 2021]. To answer that question we propose to train both CLIP and OC-CLIP architectures from scratch on combinations of CC3M, CC12M and CC3M+12M and compare both of their general understanding and compositional downstream performance. In addition to the zero-shot evaluation, we also provide a computational analysis of the binding module to gain insights into its behavior and limitations.

Training Details In order to show the potential of OC-CLIP to learn from scene-graph obtained from a non human-curated captioning dataset we train both ViT-B-16 OpenCLIP model and OC-CLIP from scratch on CC3M [Sharma et al., 2018], CC12M [Changpinyo et al., 2021] and the merge of both (15M) . We did not tune the hyperparameters and used the same hyperparameters as suggested in [Mu et al., 2021]. Both models are trained for 5, 15, and 25 epochs, using a batch size of 4096, a learning rate of $1e-3$, 2k steps learning rate warmup and a cosine decay after. As recommended by Mu et al. [2021] we used AdamW optimizer with 0.5 of weight decay and β_2 set to 0.98.We report extensive zero-shot downstreeam classification performance on the ELEVATER [Li et al., 2022] suite in Table 6. We did not use any templates and use raw labels instead. Both models were trained using 4x8 V100 GPUS with a local batch size of 128. OC-CLIP shows performance gains in both zero-shot classification (a notable +12.7% in ImageNet) when trained on the same setting. These experiments show that the structured training of OC-CLIP can scale to automatic alt-text captioning dataset. We leave further scaling for future work as the main focus of our work is to emphasize the binding problem that arises when using a vector-based representation and a set of inductive biases as a way of operating on a more structured representation (eg. scene graph).

Computational analysis of OC-CLIP In OC-CLIP the visual and text modalities representations are no longer independent (as opposed to CLIP). A image representation is the results of some text-conditioned mechanism operated by the binding module. It essentially extracts relevant visual slots that constitutes the nodes of the scene graph coming from the caption. As a result, there is some notable computational overhead introduced by the additional cross-attention operations of the binding module. In particular :

- 1. The text encoder needs to encode the N nodes and R relations of the scene graph as opposed to a single sentence encoding in CLIP.
- 2. For each Image-Graph pair, The N text nodes cross-attends to N_{im} patches of the ViT in order to extract the structured visual slots.

When training OC-CLIP from scratch we propose to mitigate those two overheads respectively by :

- 1. Using a smaller embedding width (256 vs 512) and number of layers (6 vs 12) in the text encoder. Indeed OC-CLIP only need to encode information about objects and relationships and we expect such encoding to require much less capacity than an encoder that needs to encode a whole caption composed of multiple objects and relations between them.

- 2. We operate on a reduced embedding space 256 for the binding module and thus first project the ViT-B-16 patches from a 768 to a 256 embedding space before computing the nodes to patch cross attention logits.
- 3. We use the SigLip [Zhai et al., 2023] loss to make efficient use of batch chunking and gradient checkpointing.

We only perform experiments with a B-16 architecture for the ViT but perform the computational analysis fro both B and L backbones. We report the results in Table 7 We note that there is a significant overhead with a base architecture 2.2x but since the binding module perform the same number of operations no matter what the ViT is we show that when scaling the ViT backbone, the binding module is not the bottleneck anymore and the computational overhead is reduced (1.3x).

Model	ViT Backbone	Text (w,l,ctx)	Binding Module GFLOPs	Text GFLOPS	Vision GFLOPs	Total GFLOPs
OC-CLIP	B	(256, 6, 20)	12(*num workers)	180	1k	2.2x
CLIP	B	(512, 12, 77)	-	186	1k	1x
OC-CLIP	L	(256, 6, 20)	12(*num workers)	180	4.9k	1x
CLIP	L	(512, 12, 77)	-	186	4.9k	1.3x

Table 7: Computational Comparison of CLIP and OC-CLIP. Calculations are made for a local batch size (per GPU) of 64. We give the Total GFLOPs based on a global batch size of 8192 (=128 num workers). When scaling the ViT backbone the computational overhead of the binding module remains fixed and is not the main bottleneck anymore.

A.4 Scene Graph Parsing Discussion

Comparison of different parsing methods Although the parsing method is not the core of our contribution we provide here a couple of qualitative and quantitative comparisons to motivate the choice of using an LLM to perform the parsing of the captions despite the pre-processing computational overhead it entails. We identify 3 families of parsing method that operate on text-only input and provide insights on their respective :

- **Automatic parsing methods** : method based on hand-crafted rules about the semantics in order to extract tags and more complex dependency graphs. TagAlign also compares to nltk and justifies the choice of going to an llm-based method. We consider a representative of those automatic parsing methods based on spacy [Honnibal and Montani, 2017].
- **Finetuned factual scene graph parser** trained in a supervised way to extract scene graph. We consider a representative of them, a state-of-the-art factual scene graph parser based on T5 model [Li et al., 2023b] trained to extract fine-grained scene graph information about the objects and relations in an input caption.
- **LLM-based**, here we choose llama3-8b as a representative and leave the extensive analysisof the bias/cues of different llm families of model for future work.

We identified failures modes of automatic parsing and finetuned that are relevant to compositional understanding of clip-like models and justify the use of an llm-based parsing method and summarize them in Table 8. We show on one hand that automatic parsing methods are prone to oversimplification, missing relations and mistaking an attribute modifiers with an object. On the other hand supervised scene graph parser seems to be prone to relation classification error and important atteibute binding error when the different objects mentioned in a caption share the same label tag.

Caption	Spacy	T5	LLM
A brown cat is lying on a computer	Objects: a brown cat, a computer Relations: {on, 0, 1} (Oversimplification error)	Objects: brown cat, computer Relations: {lay on, 0, 1} (Relation classification error)	Objects: brown cat, computer Relations: {lying on, 0, 1}
A man is on the left of the dog	Objects: a man, the left, a dog (Wrong POS) Relations: {of, 1, 2} (Missing relation)	Objects: man, dog Relations: {at the left of, 0, 1}	Objects: man, dog Relations: {on the left of, 0, 1}
A woman in blue and a woman in red	Objects: a woman, red, a woman, (Wrong POS) Relations: {, 0, 1}, {in, 0, 2}, in, 2, 3}	Objects: blue red clothes, woman (Wrong attribute binding) Relations: {wear, 0, 1}	Objects: woman in red, woman in blue Relations: {}

Table 8: Comparison of parsing errors made by different parsers.

We additionally train OC-CLIP on COCO captions parsed by those 3 different parsing models and compare the downstream compositional understanding performance in Figure 6. Coherent with the qualitative analysis the choice of the parsing family mostly impact relational understanding. We observe for the SugarCrepe swap object (replace rel resp.) a decrease of 9.3% (resp. 14.1%) for spacy

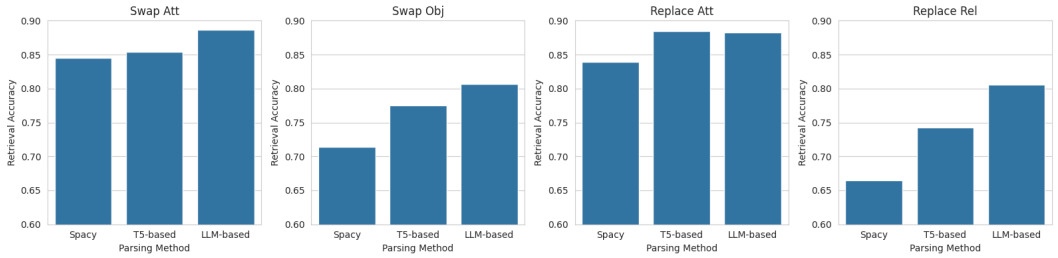


Figure 6: Downstream Compositional Understanding of OC-CLIP when trained on different parsing of COCO-Captions.

and 3.4% (resp. 6.3%) for a supervised T5 model as compared to OC-CLIP on scene graphs extracted by llama3-8b. Close to our work, TagAlign[Liu et al., 2024b] also quantitatively and qualitatively analyze the objects tags than can be extracted with an nltk-based and llm-based parser and show that training CLIP with an additional object and attribute tag classification loss with tags coming from an llm results in better downstream zero-shot semantic segemntation.

Limitations of LLM-based parsing for OC-CLIP We also acknowledge that using and LLM as a parser may also have some limitations and evaluating the impact of the downstream performance of different LLMs or VLMs is an interesting question. In particular, llm-based parsing might not extract accurate scene graphs, especially when the dependency between the objects in a captions is rather complex or ambiguous. And informing the parser in prompt with visual information might be an interesting direction. However the exact instanciation of the LLM-based parser used is orthogonal to our contribution and we leave this analysis for future work.

Scene Graph Parsing cost We performed the parsing by serving instances of Llama3.1-8b on v100 GPUs. Each datasets is then chunked in N process that do not require any GPUs and send requests to the served LLM parsers through vllm[4] to maximize the throughput of the parallelized requests. For reference we parsed the COCO datasets ($\sim$ 500k captions) parallelizing 10 instances of the parser, and with 128 chunks in 3.5 hours and Visual-Genome ($\sim$ 200k captions) with 8 instances, 64 chunks in 1.7hours. The parsing time can further be optimized by serving more instances, using more performant GPUs (A100, H100 etc..), serving each instance in parallel in more GPUs to maximized the number of requests that can be processed per second. For the cc3m and cc12m, in order to accelerate the parsing, we kept the LLM parser local using ollama[5] on v100 GPUs. CC3M was chunked into 500 and CC12M into 1000 smaller chunks, we launched the jobs sequentially. CC12M took about 3days to parse but could likewise be accelerated using faster GPUS.

A.5 Additional Compositional Understanding results

Our main goal is to evaluate CLIP-like models compositional understanding in plausible and grammatically correct cases. Hsieh et al. [2023b] have identified exploitable textual biases in previous mainstream procedurally-generated hard negatives benchmarks like the COCO and Flickr set of ARO and VL-checklist. Specifically they show that procedurally generated hard negatives are either highly grammatically incorrect and can be identified by a blind model or by a good language model that can measure the plausibility of the caption. The SugarCrepe is thus designed to pfollow the same fine-grained taxonomy on attributes, objects, relationships as VL-checklist but ensures that the hard-negative are not distinguishable by a blind model. The main results of our paper thus focus on this benchmark. We however also give the performance of our model on the full ARO suite and VL-Checklist in Table 9 for reference.

A.6 PUG Dataset

In this section we describe in more details the content of the synthetic experiments, give more context on the motivation along with additional results.

[4]https://github.com/vllm-project/vllm
[5]https://ollama.com/

Model	VL-Checklist			ARO			
	Object	Relation	Attribute	Attribution	Relation	COCO-order	Flickr-order
CLIP	80.0	63.0	67.4	63.2	60.0	47.9	60.2
BLIP	82.2	70.5	75.2	63.2	60.0	47.9	60.2
XVLM	85.8	70.4	75.1	86.8	73.4	-	-
Hard-negative Methods							
CLIP-SVLC	85.0	68.9.7	72.0	73.0	80.6	84.7	91.7
NegCLIP	84.1	63.5	70.9	71	81	86	91
CE-CLIP	84.6	71.8	72.6	76.4	83.0	-	-
MosaiCLIP	88.5	77.0	75.5	77	82.6	-	-
Dense captioning+Hard-Negative							
DAC-LLM$_{500K}$	66.5	56.8	57.4	63.8	60.1	50.2	61.6
DAC-LLM$_{3M}$	87.3	86.4	77.3	73.9	81.3	94.5	95.7
DAC-SAM$_{3M}$	88.5	89.7	75.8	70.5	77.2	91.2	93.9
DCI	80.7	70.1	68.7	67.6	76.2	88.6	91.3
DCI$_{NEG}$	88.4	61.3	70.4	62.0	57.3	39.4	44.6
OC-CLIP	90.7	80.0	75.6	84.0	84.9	94.2	84.8

Table 9: Results (%) on VL-Checklist and ARO Benchmark.

Motivation The rise of data-centric hard negative methods were motivated by the bag-of-words behaviour [Yuksekgonul et al., 2023b] of CLIP noticed in "simple swap-attribute" retrieval tasks. Hard-negative methods propose to mitigate this behaviour by finetuning CLIP-like models on data points with minimal changes but semantically different meanings. However we experimentally observed that all the methods fail to increase performance specifically in swap attribute kind of splits. In order to further isolate the root cause, we propose a series of synthetic experiments that compare covering more hard-negative data points with OC-CLIP on varying proportion of training samples and hard-negative samples. By restricting the environment to a closed-set vocabulary of backgrounds, attributes, and object classes, we can enumerate all possible hard-negatives, allowing us to systematically evaluate the effectiveness of different approaches. Our results show that simply *adding more hard-negatives plateaus and is not sample-efficient, as the swap attribute binding performance always underperforms OC-CLIP trained on less data without any hard-negatives* in a simple object-attribute binding task 2. However, when combined with OC-CLIP inductive bias, hard-negatives complementarily improve downstream performance. This suggests that our model, OC-CLIP, is a more sample-efficient approach to addressing the bag-of-words behavior of CLIP models. We hypothesize that the root cause of this issue thus lies in the representation format used in CLIP's original formulation, which relies on a single vector to capture complex semantic relationships. Our proposed method introduces inductive biases that allow the model to learn more structured representations, avoiding superposition of features [Greff et al., 2020b] and effectively mitigating the bag-of-words behavior. Through these synthetic experiments, we demonstrate the effectiveness of our approach and provide insights into the sample-efficiency limitations of existing data-centric methods.

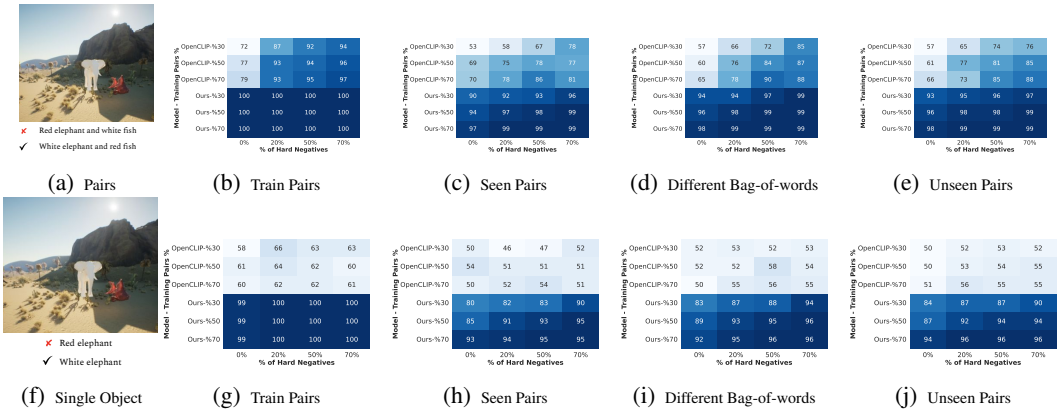
Dataset splits The synthetic experiments we propose are based on the controlled 3D environment PUG [Bordes et al., 2023]. We operate in a 3D envrionment with pairs or single textured animals in different backgrounds. The factors of variation are :

- **5 Backgrounds** : desert, arena, ocean floor, city, circus
- **20 Animals** : goldfish, caribou, elephant, camel, penguin, zebra, bear, crocodile, armadillo, cat, gecko, crow, gianttortoise, rhinoceros, dolphin, lion, orca, pig, rabbit, squirrel
- **4 textures** : red, white, asphalt, grass
- **2 spatial constraints** for pairs : left/right, above/under

The different splits We then construct splits that aim at evaluating separately attribute binding and spatial relationships understanding. In all the different splits, we include images with single animals in all the possible *background-texture-animal* conjunctions.

Attribute Binding Splits The attribute binding training and testing splits are constructed as follows : (1) - We list all the possible pairs of animals,(2) - We randomly and i.i.d. select a percentage % N_{train} of pairs to include in the train split, (3) - For each training pair we select a pair of assigned attribute (for example if cat and caribou are in the train split we will assign red to cat and white to caribou and will remove all the other attribute-animal conjunction from the training. This is done

such that we can control for the *replace attribute* hard negative presence. (4) - For each pair in the training set we separate the corresponding hard negative examples with the same bag of words but swapped attributes (referred to as *Seen Pairs* in Figure 7) and the same pair but a different bag of words (referred to as *Different Bag-of-words* in 7) , (5) - finally we also isolate unseen pairs of animals. We also include the accuracy on the training pairs that do not have their corresponding hard negatives in the test set).



(a) Pairs (b) Train Pairs (c) Seen Pairs (d) Different Bag-of-words (e) Unseen Pairs

(f) Single Object (g) Train Pairs (h) Seen Pairs (i) Different Bag-of-words (j) Unseen Pairs

Figure 7: **Attribute Binding on PUG - Additional Results** Performance of the finetuned OpenCLIP and OC-CLIP models on a binary classification task between a caption and its corresponding hard-negative. We do that for captions that mention Pairs of animals (**top row**) like the example in Figure (a) and for captions that mention a single animal (**bottom row**) like the example in Figure (b).To assess the models' performance, we compute the accuracy across two dimensions. The first one is the percentage of animal pairs (y-axis) seen during training (animals like elephants and fish could be seen either alone or with other animals but never together). The second dimension (x-axis) is the number of hard-negatives used in the training data. For instance, whether we have the combination "red elephant" and "white fish" in the training data while we only have "white elephant" and "red fish" in the test data.

Spatial Relation understanding Splits For these splits we do not assign specific pairs of attributes to train/test split but rather consider pairs of animals and their order with respect to the spatial relationship tested and systematically include all the possible attributes assignment to those pairs. We then construct the different splits by restricting the number of pairs and their spatial configuration.

Hard Negative Samples For both tasks the hard negative samples we consider are align with the test tasks taxonomy. For attribute binding we always test the model's ability to distinguish between eg. *a red cat and a white caribou* and *a white cat and a red caribou*. Hence we consider as a hard negative sample any image that corresponds to the swapped attribute version of a training pairs. To augment the dataset with hard negative, we sample i.i.d. a percentage % N_{hard} of the training pairs and include in their corresponding hard negatives in the train set. Similarly for the spatial relationship understanding task, we test the model's ability to distinguish between eg. *a red cat to the left of a white caribou* and *a white caribou to the left of a red cat*. Hence we consider as a hard negative sample any image that corresponds to the swapped order with respect to the relationship tested of the animal pairs seen during training.

A.6.1 Spatial Relation Understanding

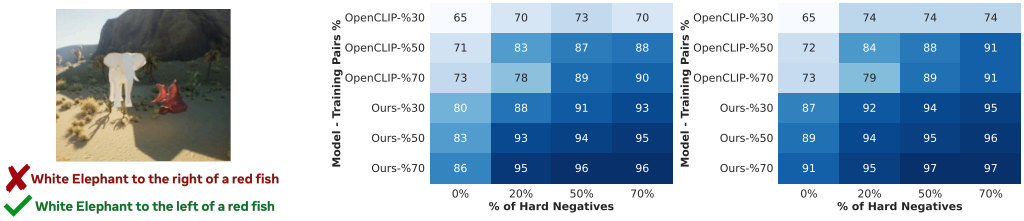
In this section, we aim to evaluate the spatial relationship understanding capabilities of the models. To do so, we conduct controlled experiments using data splits where not all pairs of animals are seen during training. The relations considered in these experiments are "left/right" and "above/below". Hence, the task is to choose between the original caption of the form "X left of Y" and the caption with the swapped order "Y left of X". We consider the following generalization axes:

- **Unseen object order:** This axis tests the generalization when swapping the order of objects in a relationship. For example, "elephant to the left of fish" may be used for training, while "elephant to the right of fish" is used for evaluation

- **Unseen object pairs:** This axis test for unseen pairs of animals in seen relationships.

We follow the experimental setup of section **??**, and finetune OpenCLIP and OC-CLIP while considering the effect of adding different % of hard negative images and/or different % of object pairs to the training data.

We test both models on image-text retrieval tasks and report the results in Figure 8. Figure 8(b) shows the results for the unseen object order generalization, whereas Figure 8(c) presents the results for the unseen object pairs. As shown in Figure 8(b), OC-CLIP outperforms OpenCLIP in all data regimes considered, with improvements between 6% and 18%. Similarly, as shown in Figure 8(c), OC-CLIP improves upon OpenCLIP in all data regimes, yielding absolute improvements between 5% and 20%.



(a) Example image (b) Unseen object order (c) Unseen object pairs

Figure 8: **Spatial Relationship Understanding.** We finetune OpenCLIP and train OC-CLIP's binding module on splits containing different % of animals pairs (y-axis) and different % of hard-negative image in the training split (x- axis). We test the models on images with either unseen order (b) or unseen pairs (c) during training. The testing is done against the swapped order of the ground truth caption as shown in the visual example (a).

A.7 Parsing

For the parsing of the training and testing data we used a llama-3-70b Instruct model with the following prompt :

Parsing Prompt

Given a caption, your task is to parse it into its constituent noun phrases and relationships. The noun phrases should represent independent visual objects mentioned in the caption without semantic oversimplification. For each caption, output the parsed noun phrases (e.g., entities) and relationships in JSON format, placing the dictionary between [ANS] and [/ANS] brackets. In the relationships, use indices to specify the subject and object of the relationship mentioned in the caption. The indices of the subject and object should be integers. Here are a few examples:

```
Caption: A large brown box with a green toy in it
Output:
[ANS]
{
  "entities": [
    "large brown box",
    "green toy"
  ],
  "relationships": [
    {
      "relationship": "in",
      "subject": 1,
      "object": 0
    }
  ]
}
[/ANS]

[...] More examples
```

PAY ATTENTION to the following:
- Relationships MUST relate two different entities in the caption and NOT be unary. For example, in the caption 'red suitcases stacked upon each other', 'stacked upon each other' is not considered a relationship.
- Do not forget any relationships.
- Relationships MUST be directed. 'and' is not a relationship.
- Pay attention to spatial relationships like 'behind', 'left of', 'with', 'below', 'next to', etc. 'and' is not a relationship.
- Check the right dependencies when the relationships are not direct. In the caption template a X with a Y in it, it refers to X.
- Pay attention to co-references.

Now, parse the following caption into its constituting entities and relationships. You MUST place the answer between [ANS] and [/ANS] delimiters.
Caption:

To showcase the effectiveness of this parser, we are showcasing below the most and least common entities and relations that are find by this parser across MS-COCO and CC12M. In Figure 9, we can see that the most common entities are people for the COCO dataset as well as for CC12M in Figure 11. We also plot the least common entities in Figure 10. Finally, we also compute the number of tokens that is require for modelling the entities and relations. In Figure 14, we display the frequency of the number of tokens use to encode the relations and entities on COCO. Using just 10 tokens, we can encode most of the entities and relations, thus we do not need to have a text encoder that take into input 77 tokens but can use a much smaller one instead. In Figure 15, we show a similar plot but normalized. Even on a dataset with noisy captions like CC12M, most entities and relations can be encoded with less than 20 tokens.

A.8 Datasets

Training Data For the compositional experiments we train both OpenCLIP and OC-CLIP on a aggregated data form COCO-Captions (COCO) [Lin et al., 2014], Visual Genome (VG) [Krishna et al., 2017] and GQA [Hudson and Manning, 2019]. All these datasets cover the same 110k images from COCO but focus on different kind of annotations. COCO provide global scene annotation, Visual Genome emphasizes specific region descriptions and general relationships and GQA annotates

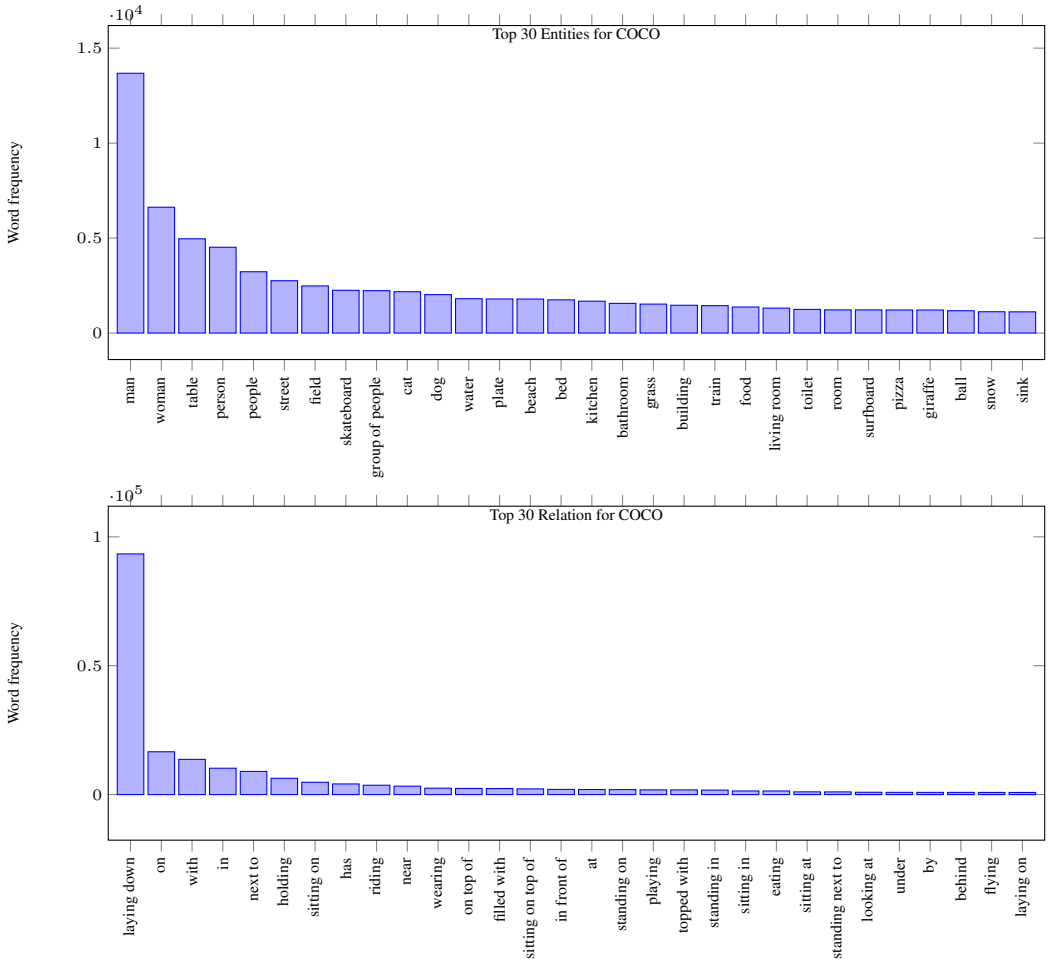


Figure 9: Plot of the most common entities and relations that were extracted by our LLM-based parser for the COCO datasets.

both objects and spatial relationships. Both Visual Genome and GQA have annotated scene graph that we do not need to parse to train OC-CLIP. For OpenCLIP, we sample 2 region annotations from VG to from a caption following this template *A photo of a* {*Region 1*} *and a* {*Region 2*}. Similarly to get the captions from GQA, if there is a relationship we follow Kamath et al. [2023] and give the model a caption following this template *A photo of* {*Subject*} {*Rel*} {*Object*}. If only objects are mentionned we sample up to 3 objects and give the model a caption following this template *A photo of* {*Obj1*},{*Obj2*},{ *Obj3*} .

A.9 Training Details and Hyperparameters

In table 10 we detail the hyperparameters of the OC-CLIP architecture for results in real-world compositional understanding (section 4.2).

Optimization Details In order to train OC-CLIP we followed prior work and use Adam Optimizer with β_1 and β_2 set to 0.9 and 0.95 and a weight decay of 0.2. We used different learning rate for the pretrained backbones and for our modules that we train from scratch : learning rate of $2e^{-4}$ for the binding and the scoring modules, learning rate of $2e^{-5}$ for the text Transformer backbone, and a smaller rate of $1e^{-6}$ for the ViT backbone. We also used a warmup schedule for both of the text (1k steps) and the vision (5k steps) backbones followed by a cosine decay. We train the model for a total of 100 epochs.

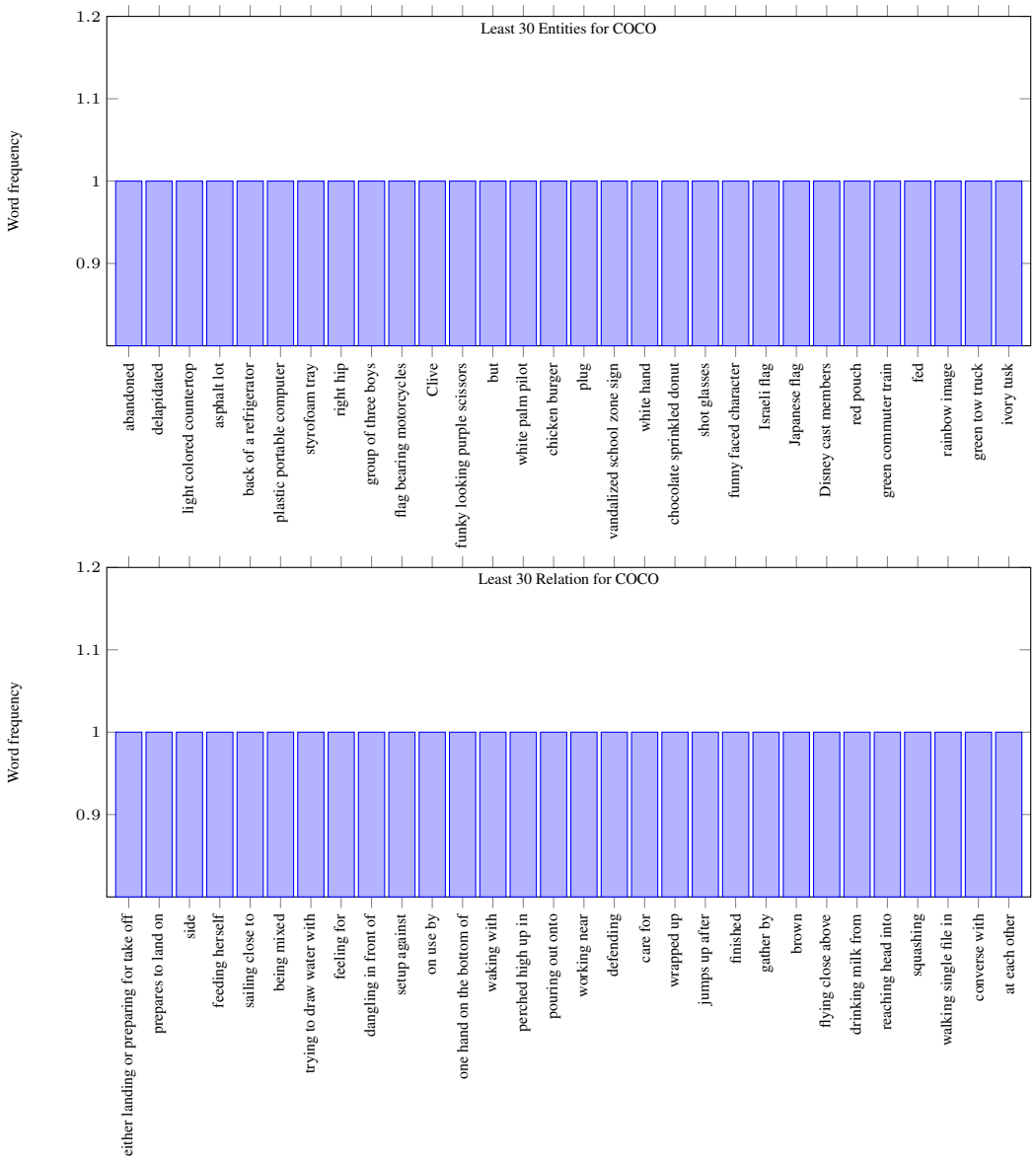


Figure 10: Plot of the least common entities and relations that were extracted by our LLM-based parser for the COCO datasets.

A.10 Binding Module Code

See Figure 16

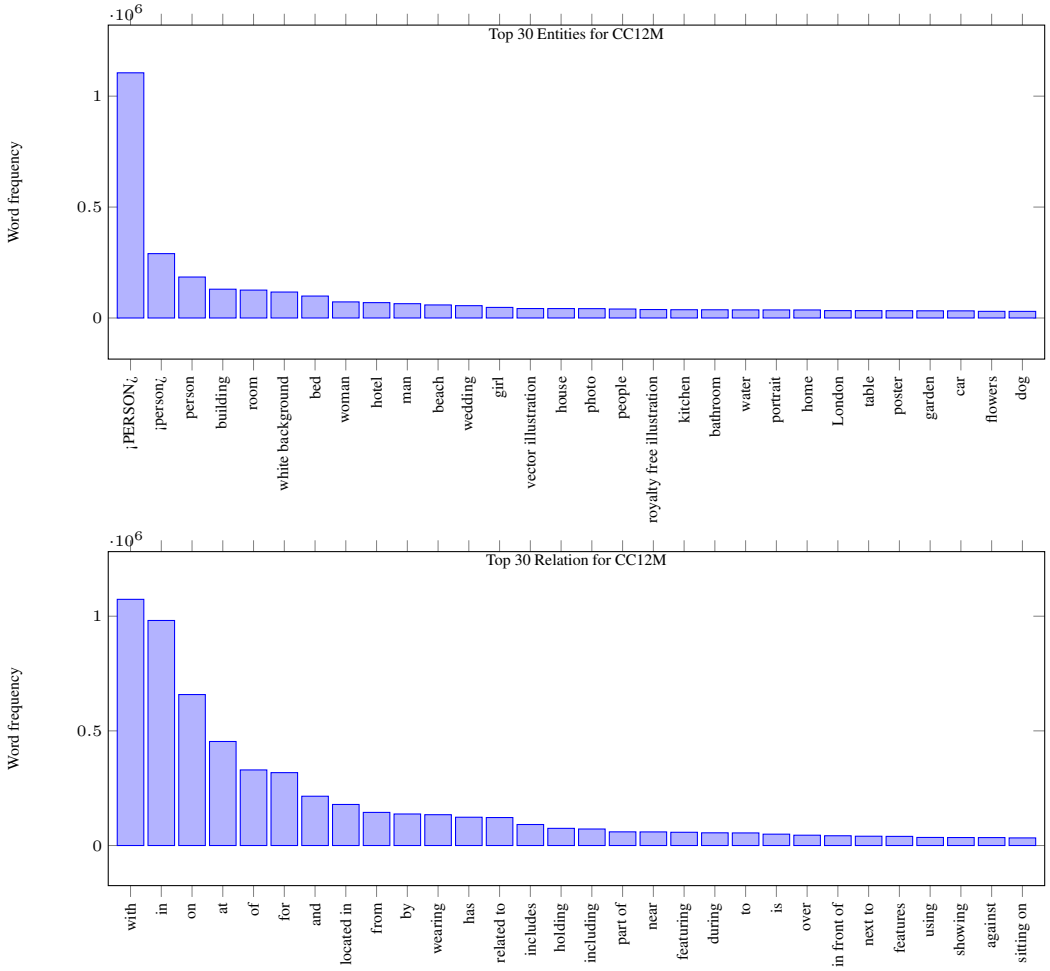


Figure 11: Plot of the most common entities and relations that were extracted by our LLM-based parser for the CC12M dataset.

Hyperparameter/Parameter Init	Architecture	Value
Binding Module		
– Image Patches Processing	Linear	768×256
– Self-Attention #Layers/#Heads		2/4
– Self-Attention MLP ratio/act		2/nn.GELU
– Keys K, Values V	Linear	256, 256
– Normalization Keys/Values	LayerNorm	256
Grouping Module		
– Cross-Attention #Heads		1
– Queries	Linear	256
– Normalization Queries	LayerNorm	256
– Num Default Tokens Q_{default}	nn.Param(N_d,256)	1
Scoring Functions		
– Object Scoring Function	cosine sim	
– Relation Scoring subject f_s	MLP(128 + 256, 128)	2 layers
– Relation Scoring object f_o	MLP(128 + 256, 128)	2 layers
– Coef ent init (learned parameter)		1.5
– Coef rel init (learned parameter)		0.5

Table 10: Table of hyperparameters for OC-CLIP architecture

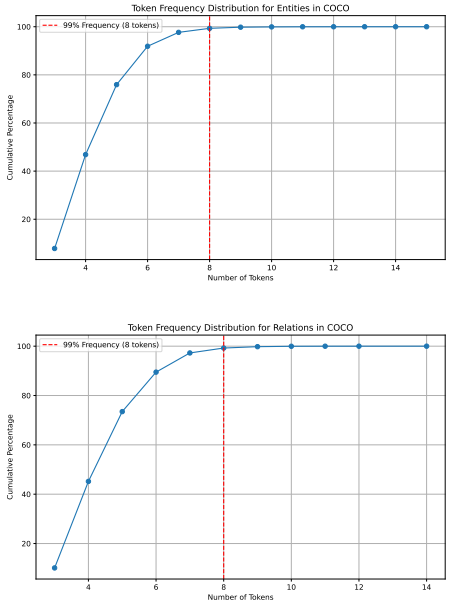


Figure 12: Distribution of the number of tokens require for modeling the entities and relations on COCO (we do not need more than 8 tokens to capture 99% of the entities in COCO). Since we need less token, we can leverage a smaller text encoder to extract the entities and relations.

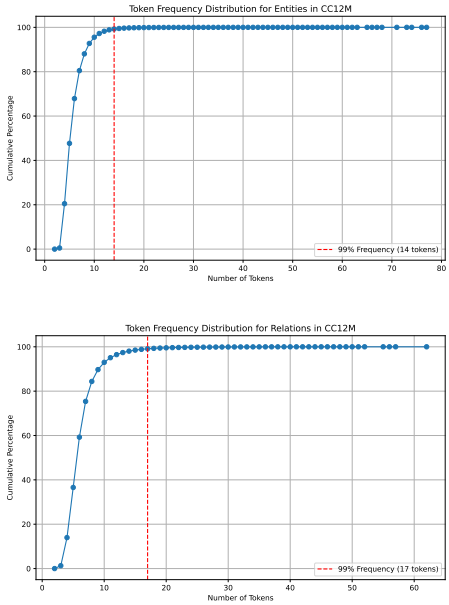


Figure 13: Distribution of the number of tokens require for modeling the entities and relations on CC12M (we do not need more than 14 tokens to capture 99% of the entities in CC12M). Since we need less token, we can leverage a smaller text encoder to extract the entities and relations.

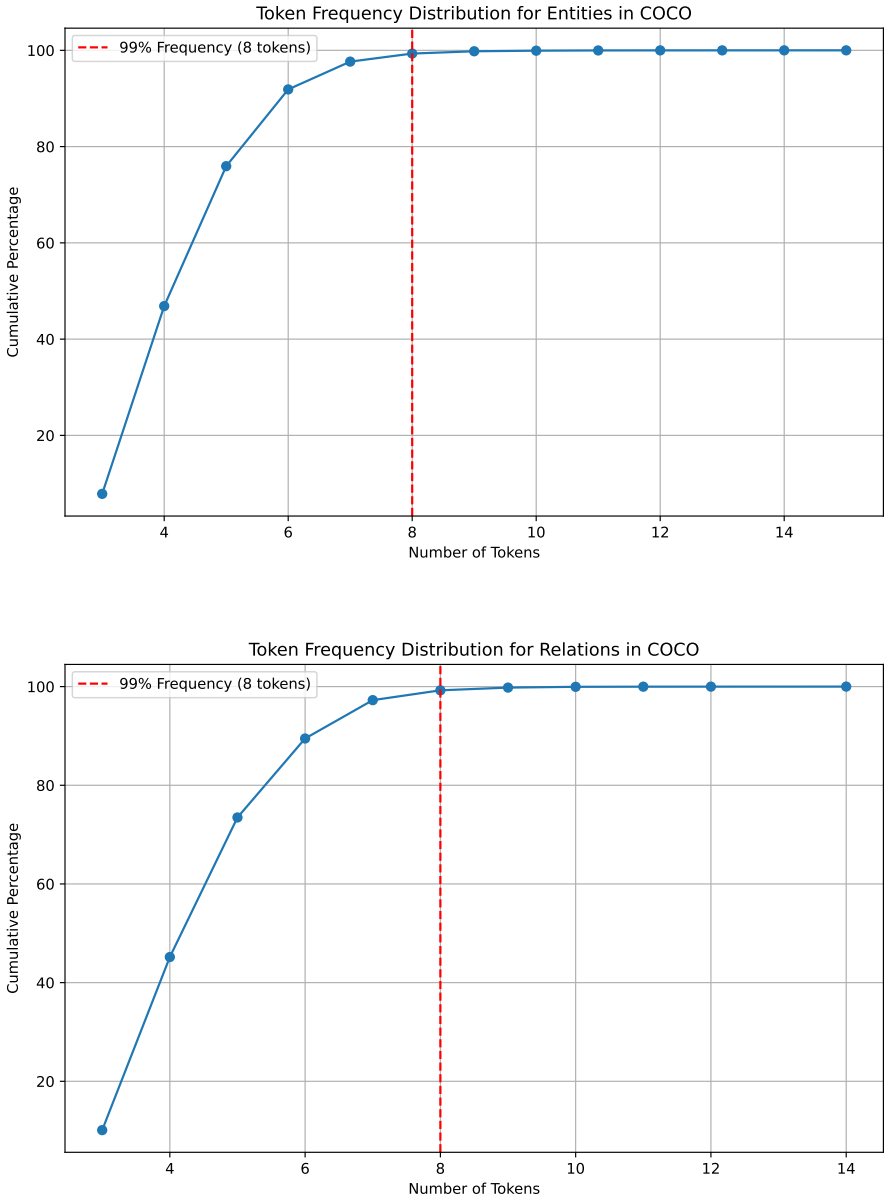


Figure 14: Distribution of the number of tokens require for modeling the entities and relations on COCO (we do not need more than 8 tokens to capture 99% of the entities in COCO). Since we need less token, we can leverage a smaller text encoder to extract the entities and relations.

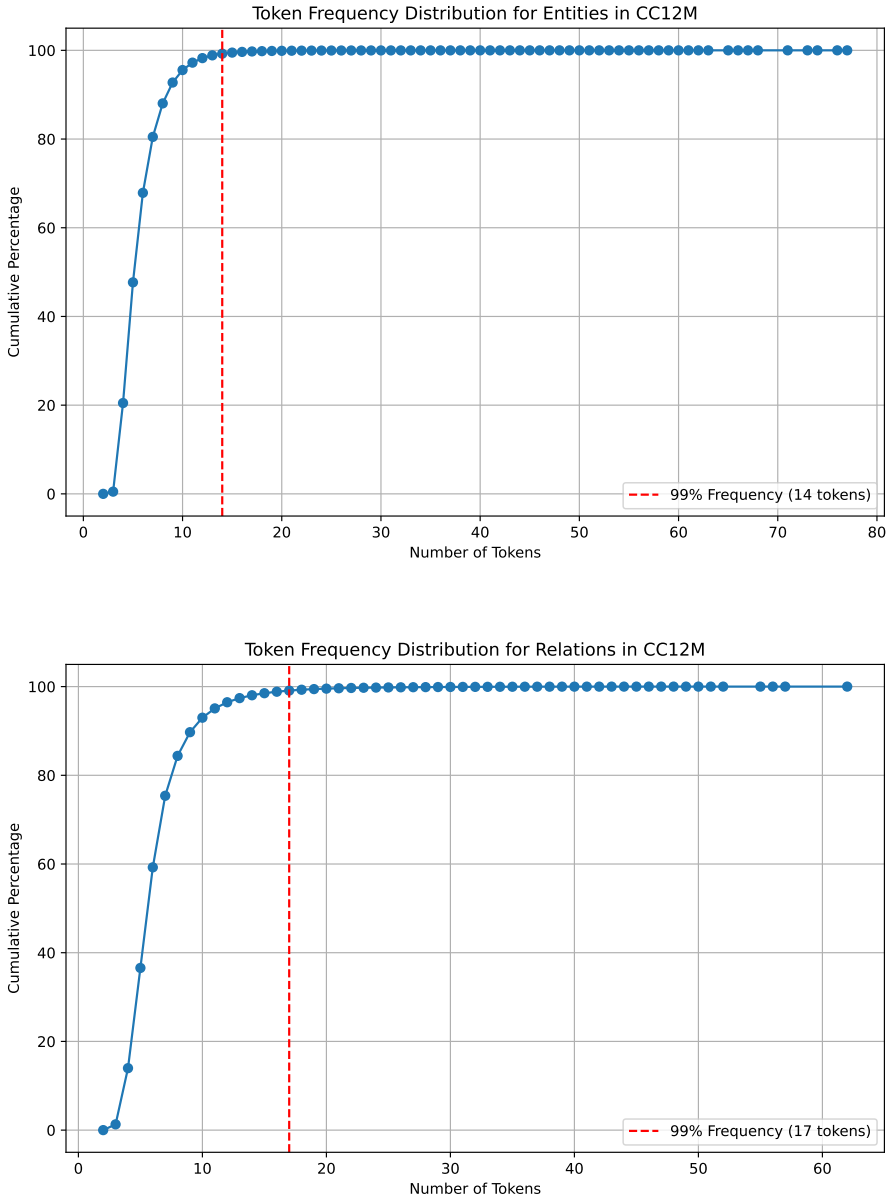


Figure 15: Distribution of the number of tokens require for modeling the entities and relations on CC12M (we do not need more than 14 tokens to capture 99% of the entities in CC12M). Since we need less token, we can leverage a smaller text encoder to extract the entities and relations.

```
class AssignAttention(nn.Module) :
    def __init__(
        self, dim, qkv_bias=False, qk_scale=None):
        super().__init__()
        self.scale = qk_scale or dim**-0.5
        self.q_proj = nn.Sequential(nn.Linear(dim, dim, bias=qkv_bias))
        self.k_proj = nn.Sequential(nn.Linear(dim, dim, bias=qkv_bias))
        self.v_proj = nn.Sequential(nn.Linear(dim, dim, bias=qkv_bias))

    def forward(self, query, key=None, value=None):
        #before cross attention projections
        q = self.q_proj(query)
        k = self.k_proj(key)
        v = self.v_proj(value)
        #scaled dot product
        attn = (q @ k.transpose(-2, -1)) * self.scale
        #softmax across query dim
        attn_dim = -2
        attn = F.softmax(attn, dim=attn_dim) + 1e-8
        #attn normoalization
        attn = attn / (attn.sum(dim=-1, keepdim=True))
        output = torch.einsum("bqk,bkd->bqd", attn, v)
        return output
class BindingModule(nn.Module) :
    def __init__(self,in_vis_dim, dim, num_patches, num_default_tokens) :
        super().__init__()
        self.im_proj = nn.Sequential(nn.Linear(in_vis_dim, dim),
        nn.GELU(), nn.Linear(dim, dim))
        self.pos_embeddings = nn.Parameter(torch.randn(num_patches, dim))
        self.img_processor = nn.Sequential( ResidualAttnBlock(dim, 4),
        ResidualAttnBlock(dim, 4))
        self.default_tokens = nn.Parameter(
        torch.randn(1, num_default_tokens, dim))
        self.to_kq_groups = nn.Sequential(nn.Linear(dim, 2 * dim))
        self.dim = dim
        self.num_default_tokens = num_default_tokens
        self.k_norm = nn.LayerNorm(dim)
        self.v_norm = nn.LayerNorm(dim)
        self.assign_slots = AssignAttention(dim)

    def encode_patches(self, patches) :
        patches = self.im_proj(patches)
        patches = patches + self.pos_embeddings
        patches = self.img_processor(patches)
        K_img, V_img = torch.split(
        self.to_kq_groups(patches),self.dim,dim=-1)
        K_img, V_img = self.k_norm(K_img), self.v_norm(V_img)
        return K_img, V_img

    def group(self, query_tokens, K_img, V_img) :
        #adding default tokens
        query_tokens= torch.cat([query_tokens,default_tokens ], 1)
        out = self.assign_slots(query_tokens, K_img, V_img)
        #remove default tokens
        out = out[:, :-self.num_default_tokens]
        return out
```

Figure 16: Code for the Binding Module

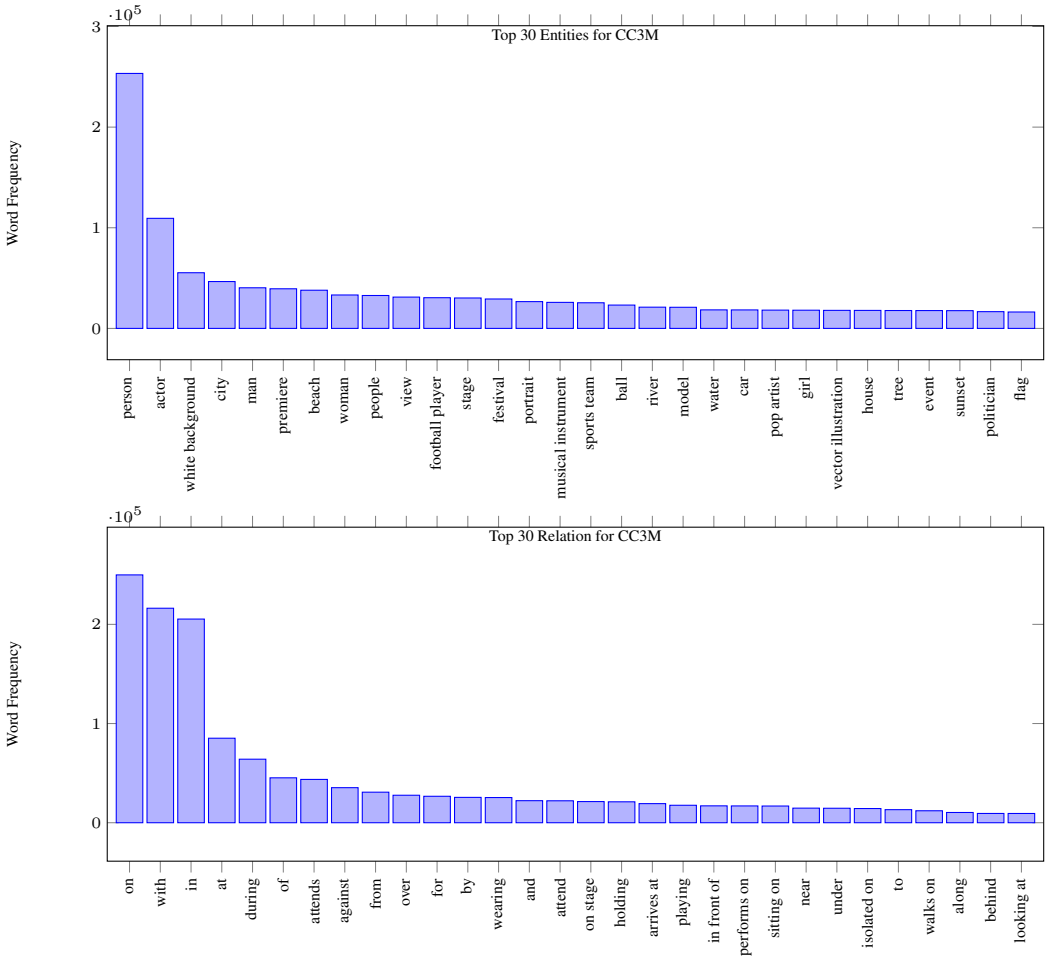


Figure 17: Top entities and relation for CC3M

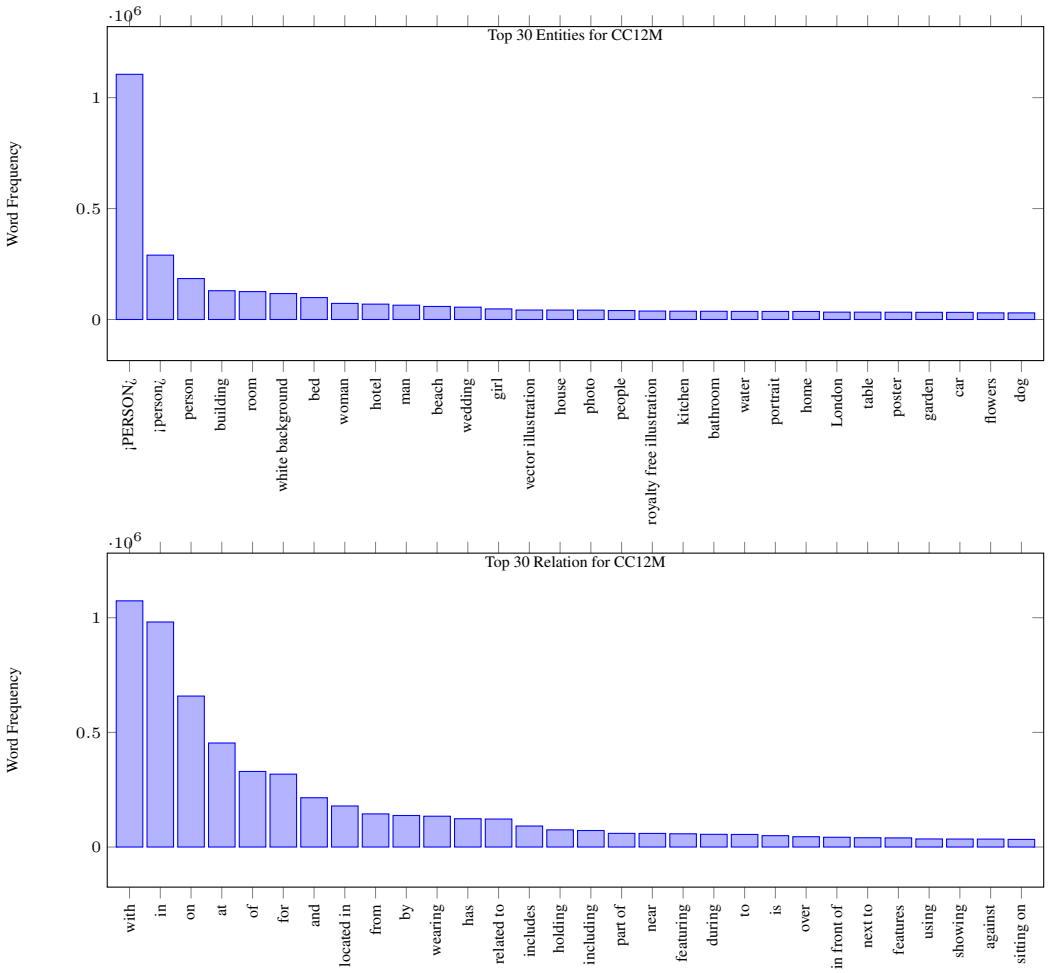


Figure 18: Top entities and relation for CC12M

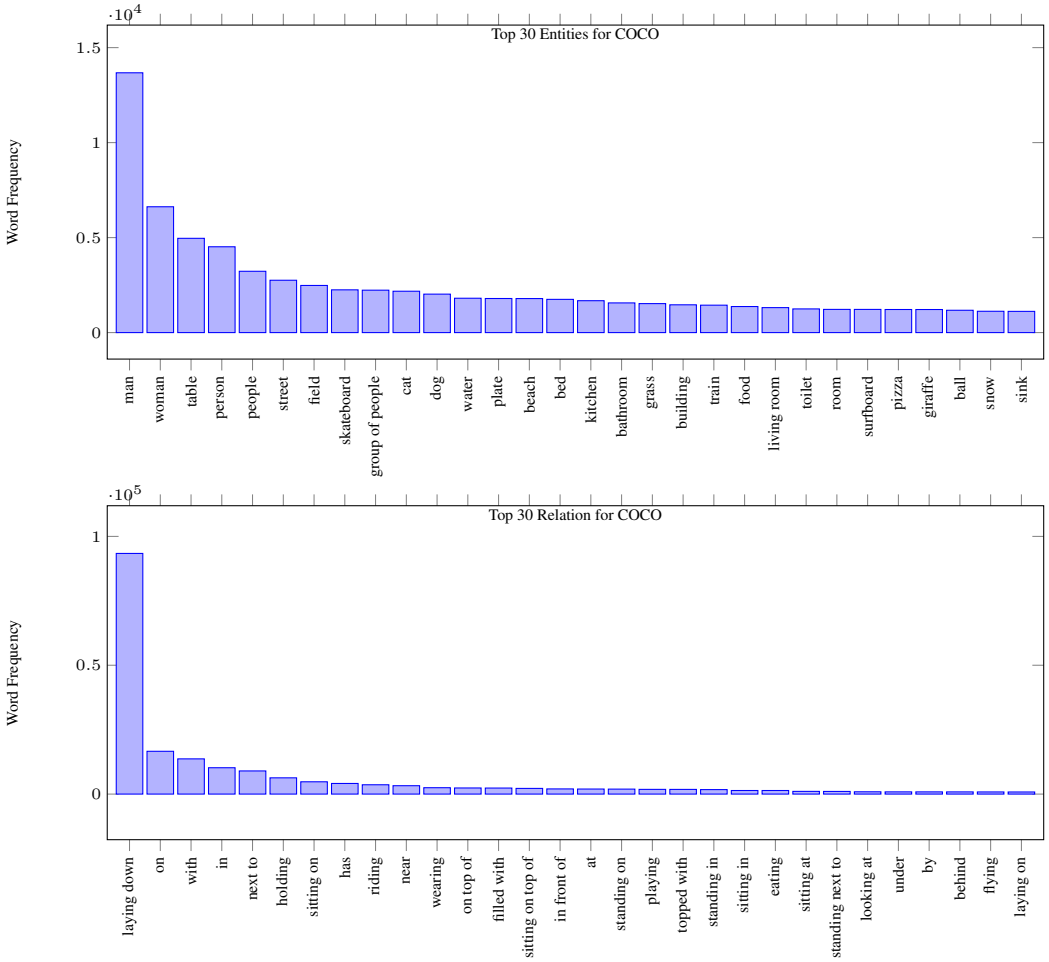


Figure 19: Top entities and relation for COCO

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The 3 parts of our Results section each verify the 3 claims listed in the introduction

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: We discuss limitations of our method (scale, computational overhead, parsing) extensively in our appendices.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: No theoritical results

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We disclose all the training parameters, dataset preprocessing, and pseudo code of our binding module to reproduce the results.

   Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Pseudo code is given, we plan on releasing the code later.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All the details are given in Appendix A.9

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We added an impact statement in the Appendix

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: We plan to release the code later.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification:

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification:

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.