Enhancing Multimodal Compositional Reasoning of Visual Language Models with Generative Negative Mining

Ugur Sahin*[1] Hang Li*[2,3] Qadeer Khan[1,4] Daniel Cremers[1,4] Volker Tresp[2,4]
[1]Technical University of Munich [2]LMU Munich [3]Siemens AG
[4]Munich Center for Machine Learning

Abstract

Contemporary large-scale visual language models (VLMs) exhibit strong representation capacities, making them ubiquitous for enhancing image and text understanding tasks. They are often trained in a contrastive manner on a large and diverse corpus of images and corresponding text captions scraped from the internet. Despite this, VLMs often struggle with compositional reasoning tasks which require a fine-grained understanding of the complex interactions of objects and their attributes. This failure can be attributed to two main factors: 1) Contrastive approaches have traditionally focused on mining negative examples from existing datasets. However, the mined negative examples might not be difficult for the model to discriminate from the positive. An alternative to mining would be negative sample generation 2) But existing generative approaches primarily focus on generating hard negative texts associated with a given image. Mining in the other direction, i.e., generating negative image samples associated with a given text has been ignored. To overcome both these limitations, we propose a framework that not only mines in both directions but also generates challenging negative samples in both modalities, i.e., images and texts. Leveraging these generative hard negative samples, we significantly enhance VLMs' performance in tasks involving multimodal compositional reasoning. Our code and dataset are released at https://ugorsahin.github.io/enhancing-multimodal-compositional-reasoning-of-vlm.html.

1. Introduction

Contrastive learning has been demonstrated to be an effective and popular technique for training large large-scale visual language models [26, 34, 50]. This is due to the availability of a large corpus of images and text reaching an order of millions of samples that can readily be scraped from the internet [39, 40]. Training on human-curated supervised data of a similar scale would be infeasible due to the sheer amount of effort required for annotation. Meanwhile, training on the limited supervised data would not yield results of performance comparable to that of contrastive methods trained on data of much higher magnitude. In fact, this contrastive pretraining on a large corpus of data has led to

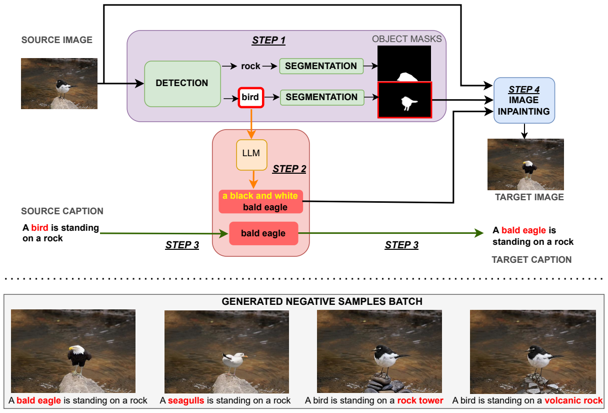

SOURCE IMAGE
STEP 1
OBJECT MASKS
DETECTION
rock
bird
SEGMENTATION
SEGMENTATION
STEP 4
IMAGE
INPAINTING
LLM
STEP 2
a black and white bald eagle
bald eagle
TARGET IMAGE
SOURCE CAPTION
A bird is standing on a rock
STEP 3
STEP 3
A bald eagle is standing on a rock
TARGET CAPTION
GENERATED NEGATIVE SAMPLES BATCH
A bald eagle is standing on a rock
A seagulls is standing on a rock
A bird is standing on a rock tower
A bird is standing on a volcanic rock

Figure 1. **Top:** Gives the overview of our proposed generative approach for image-text synthesis from a given source image and a corresponding caption. *Step 1:* The source image is first passed through a detection and segmentation algorithm to identify all the relevant objects in the scene (bird and rock) and also create independent masks of these objects (See Subsection 3.1.1). The remaining steps in this figure focus on the bird object. *Step 2:* A large language model (LLM) then takes the detected objects to create 1) an alternate representation of that object (bald eagle) 2) A more fine-grained and descriptive representation of the same object (a black and white bald eagle) (See Subsection 3.1.2). *Step 3:*. The source caption is replaced with an alternate representation to produce the target caption. *Step 4:* The original mask of the object and the descriptive alternate caption are fed to an inpainting algorithm to replace the *bird* with *a black and white bald eagle* in the source image to produce the target image (See Subsection 3.1.4). **Bottom:** Shows a batch of some other generated variations of the same source image.

[*] Equal contribution.

enhanced image and text representations that benefit downstream tasks including image and text retrieval [31, 49], text generation [24, 25, 30], and image generation [35, 37].

Despite the impressive results VLMs have achieved on the above tasks, one challenging problem that still remains is their limited compositional ability [29, 42, 49]. Compositionality refers to the challenge where the samples have significantly different semantic scene depictions despite similar textual representations. For e.g. the two sentences *1) a black dog with a white cat* and *2) a white dog with a black cat* may appear to be textually similar but have very different scene depictions. While humans can easily discern the context between the two sentences, VLMs tend to struggle, posing a significant challenge in regards to this compositional reasoning [38, 42]. This is further exacerbated when words in the sentences are exactly the same but differ only in order, as is the case in the example described above. [42] proposed a new dataset to specifically evaluate this compositional reasoning of various VLMs. They showed that these VLMs tend to struggle with compositionality. A plausible explanation is given in recent work which identified that VLMs are prone to exploiting shortcut strategies [12]. Given a caption, the VLM may choose to focus on only a certain region among the rich scene representation while ignoring other objects in the scene. For e.g. given the source caption in Fig. 1: *a bird is standing on a rock* may decide to only focus on the bird and the rock while completely ignoring the background and placing less emphasis on the bird species or type of rock. This tends to happen because in the usual contrastive learning setting, the negative samples are already significantly different from one another. Hence, VLMs only need to detect these major differences, instead of truly understanding the complex structure of the entire scene [5].

To train a model to truly compositionally reason about the scene and text, we would like to mine for hard negative samples. This hard negative mining is among the promising directions to tackle this problem [16, 26, 31, 36, 49]. It includes finding examples with minimal changes in the text or image but yielding different contexts. The bottom part of Fig. 1 shows four negative samples of the source data point. Note that the caption and image of the negative samples have a subtle difference from the source but it completely changes the context (bird species, rock formation). Such negative samples capture a more fine-grained representation of the image and text content. The model is now forced to learn the subtle differences between for e.g. a seagull and a bald eagle or volcanic rock and a tower of rocks. Just being aware that some bird is sitting on some rock would not be enough for the model. It has to additionally focus on the bird species and type of rock formation. However, such hard negative samples with subtle differences in text and images may not necessarily exist. Therefore, how do we mine for such non-existent hard negative samples in both modalities, i.e., text and images? For text, most existing works on negative mining augment the textual descriptions [9, 16, 49]. But how can we ensure that the generated sentences are even linguistically meaningful? Moreover, how do we mine hard negative samples in the image space?

Motivated by recent advances in image understanding and generation [14, 23, 25, 35, 37], we propose a framework to generate negative images to facilitate contrastive learning. Specifically, recent development in image understanding models such as SAM enables a reliable segmentation of objects from a complex scene. Moreover, large image generation models such as Stable Diffusion (SD) can convert text descriptions into images. The inpainting mode of SD allows it to modify part of the image while keeping the remaining part unchanged. As shown in the upper part of Fig. 1 with blue arrows, we are able to edit the original images with minimal changes in the pixel space thus constituting hard-to-discriminate image examples. This is done by replacing the word *bird* with the prompt *bald eagle*. Generating images similar in category (e.g., bird) but different in appearance for a large number of samples would be a tedious process for a human. Therefore, we automate the process using LLMs to generate such prompts. These LLMs automatically propose alternative concepts to replace the original words in the caption without losing the linguistic meaning. As shown in Fig. 1, the LLM changes the word *bird* into a specific category of *bald eagle*. In the end, we obtain a batch of hard negative examples at the bottom of Fig. 1. To match the left-most image from four texts with minimal word changes, the model needs to encode fine-grained bird and rock information from the image and texts.

We conducted extensive experiments using the dataset generated by our proposed method to demonstrate the power of our framework. In this regard, the contributions of our framework are summarized as follows:

- We show that our method which uses negative sample generation improves VLM performance on a wide range of benchmarks meant to assess compositional visual-language reasoning. These include Winoground, ARO, CREPE, and VL-Checklist.

- We release our dataset which is comprised of fine-grained object differences and attribute changes in the images and text. Such subtle differences in the dataset make it challenging for pre-trained state-of-the-art visual language models to correctly compositionally reason about the data points. The supplementary material contains a subset of our dataset. The complete dataset is accessible at our project page https://ugorsahin.github.io/enhancing-multimodal-compositional-reasoning-of-vlm.html.

2. Related Work

Contrastive pre-training of VLMs Contrastive Pre-training of large-scale models trained together on both vision and language modalities have shown superior representation [26, 34] and zero-shot transfer ability [16], leading to success on a wide spectrum of related tasks [25,30,37]. Due to the significant amount of image-text data crawled from the internet [39, 40], the unsupervised contrastive learning paradigm stands out as a primary approach to pre-train VLMs [18, 34, 47]. Contrastive learning relies on negative examples which train models to discriminate between them and the positive examples. If the negative examples are significantly different from the positive, the model can easily discriminate between the two. In contrast, if the negative and positive samples are similar, the model can learn to correctly discriminate only if it understands the subtle, fine-grained differences between the two. Such hard negative samples provide the model with greater predictive power. Inspired by metric learning [36], hard negative mining based on learned embeddings became a popular approach to improve contrastive learning [36], with different methods for mining negative samples being proposed [4,6,17,33,45,51]. To address the limitation that certain negative examples are hard to find in existing datasets, recent works have rather explored synthesizing negative text, and hard negative captions on standard image-text datasets [27] by word shuffling [49], negative verbs [31], negative text augmentation [8, 9, 16, 31, 38]. Compared to these approaches, our method focuses on mining hard negatives from both image and text domains, leveraging large-scale generative language and vision models.

Benchmarking Visual-linguistic Compositional reasoning Compositional reasoning is the ability to understand complex scenes and text with diverse structures [42]. This includes for e.g. the capacity to discern between sentences with the same words but in a different order, or a scene with the same objects but slightly different colors, etc. There are many datasets [7, 29, 42, 52] used for benchmarking different aspects of compositional reasoning. For e.g. Winoground [42] tests for rich structures in text order, CREPE [29] for constituting objects, their relations, and attributes, ARO [49] for shuffled word order. These benchmarks demonstrated that most SOTA VLMs showed poor performance when probed for compositional reasoning. Our method on the other hand is capable of understanding the subtle visual-lingustic cues thereby demonstrating superior performance.

Additional Image and Text Data Generation Synthetic image generation using text-to-image models has proven effective in various computer vision tasks such as image classification [13], object detection [32, 46], image captioning [44], and contrastive learning [2]. Generated images can complement existing datasets with a diverse set of images that may not be present in the existing datasets, enriching the overall range of available visual examples. [2] propose to improve contrastive learning with synthetic image generation, which is probably the closest to our work. However, it generates synthetic images from scratch, whereas we edit realistic images from a human-curated dataset. Generating additional text samples from LLMs is a very promising direction. LLMs such as ChatGPT and LLaMA exhibit well modeling of language structure [1, 11] and thus can be utilized to manipulate text to enrich the text samples [9]. For example, [10] proposes to rewrite texts in COCO to improve contrastive image-text pertaining [22]. Similarly, we utilize LLM to generate contrastive text samples for detected objects in the image. Then we utilize text-to-image models to edit the original image to obtain negative examples.

3. Method

This section first outlines our data generation pipeline that leverages the latest state-of-the-art LLMs and multimodal generative models for high-quality sample generation (See Fig. 1). We then present the finetuning framework that exploits our new dataset of hard negative examples to enhance the compositional reasoning abilities of VLMs.

3.1. Hard Negative Example Generation

Our framework can be used to enhance the richness of any image-caption pair dataset. For our experiments, we use data generated based on the human-curated image-text pair COCO dataset, but our approach can be extend to other datasets. In the following, we describe the core components of our generation pipeline. The main objective is to generating challenging negative examples which modifies local semantics of the scene and preserves the main context.

3.1.1 Image Analysis and Object Extraction

To accurately identify the regions of an image that need modification, we utilize a comprehensive annotation approach to decompose the scene into its constituent parts. Firstly we utilize off-the-shelf image-to-text models, specifically Tag2Text [15], for object detection and caption generation. As shown on the left side of Fig. 2, the Tag2Text model outputs a list of detected object labels in the image, along with a descriptive caption summarizing the entire scene. The descriptive caption is needed to ensure that all the identified objects have been covered in the caption.

However, Tag2Text lacks precise object localization capabilities and cannot demarcate precise object boundaries in the scene. To circumvent this issue, we integrate a segmentation model, such as Grounded-SAM [20, 28] into our framework. The segmentation model takes as input both the image and a label of one of the objects in the same image.

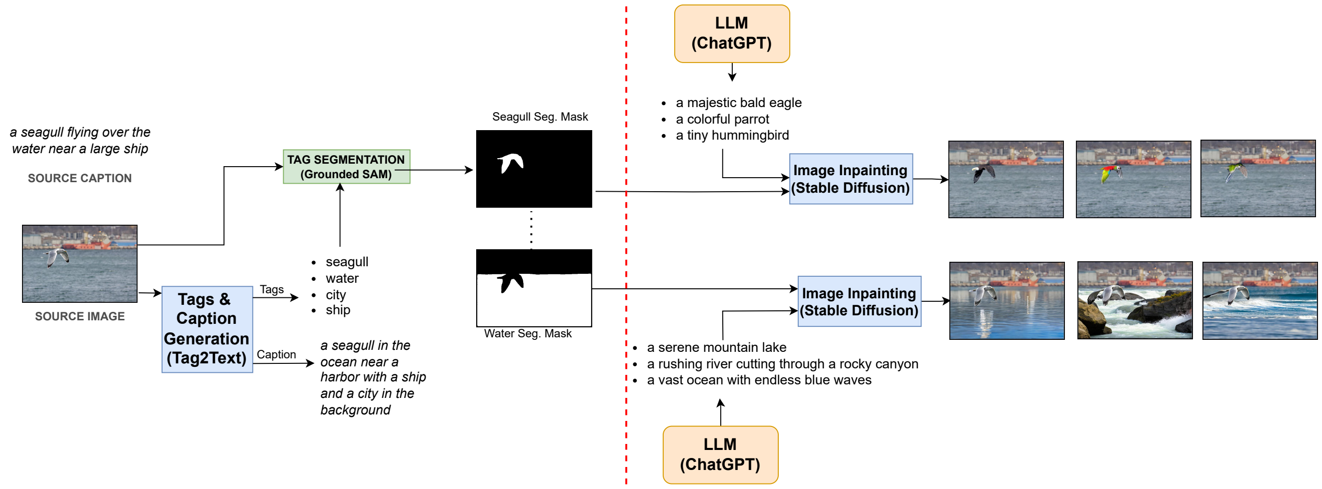

a seagull flying over the water near a large ship
SOURCE CAPTION
SOURCE IMAGE
TAG SEGMENTATION (Grounded SAM)
Seagull Seg. Mask
Water Seg. Mask
Tags & Caption Generation (Tag2Text)
Tags
seagull
water
city
ship
Caption
a seagull in the ocean near a harbor with a ship and a city in the background
LLM (ChatGPT)
a majestic bald eagle
a colorful parrot
a tiny hummingbird
Image Inpainting (Stable Diffusion)
Image Inpainting (Stable Diffusion)
a serene mountain lake
a rushing river cutting through a rocky canyon
a vast ocean with endless blue waves
LLM (ChatGPT)

Figure 2. Overview of data generation pipeline. **Left:** The portion to the left of the red dotted line demonstrates the process for determining segmentation masks of all objects in the scene, which is elaborated in Subsection 3.1.1. The Tag2Text model is first utilized to generate a list of tags for all objects in the scene. Segmentation masks from the source image are then created for all the individual tags (Masks for the seagull and water tags are shown). Note that the human-annotated source caption may not contain all the identified tags, e.g., city. Therefore, Tag2Text also generates a caption for the source image to encompass all the detected objects. The replacement of concepts for a new caption generation is explained in Subsection 3.1.3. **Right:** The portion to the right of the red dotted line figure corresponds to the process of generating images having subtle variations from the source image, as explained in Subsection 3.1.4. For this, we use the Stable Diffusion model which takes the segmentation masks along with the fine-grained description of objects with which the masks are replaced. The new descriptions are produced using ChatGPT.

Its output generates a binary mask that highlights the corresponding object region within the image.

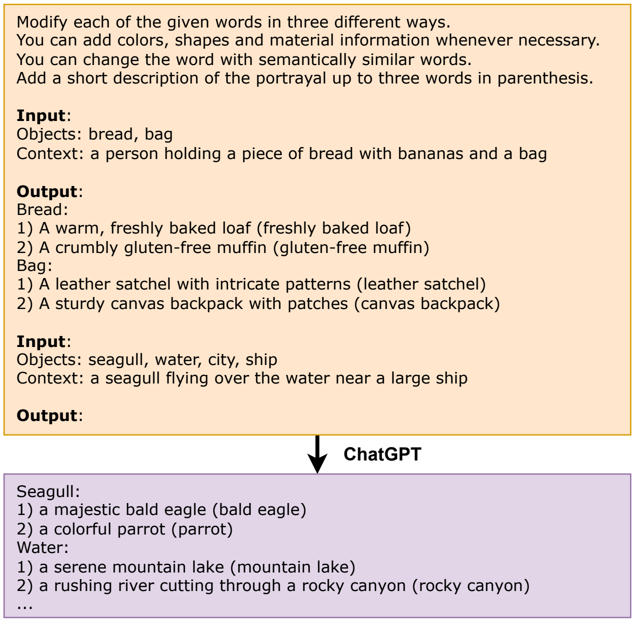
3.1.2 Concept Augmentation Using LLM

A detected object is transformed into a similar concept. Our aim focuses on modifying the object's appearance, attributes, and categories while keeping other things and the overall context the same. This includes transforming an object into a more fine-grained instance with richer attributes (e.g., transforming a house to a Victorian one with a wooden entrance), or modifying the background into different environments (transforming the sky into rocky mountains). For that, we resort to LLMs, such as open-sourced LLaMA [43] and ChatGPT which offer impressive possibilities due to their in-context learning capacity [43]. Given a few examples and a test sample, LLMs generate the output that adheres to the structures implied by the set of given examples. For instance, as illustrated in Fig. 3, the input prompt presents an example where the word *bread* is modified into *freshly baked loaf*. When the LLM is prompted with a test case *water*, it generates a *mountain lake* that follows a similar modification pattern. To enhance the generation, a source caption is fed to the LLM as context information, encouraging the generation of object variations that are more compatible with the background. Additionally, the prompt is manually designed to guide the LLM toward producing the desired output, i.e., instructions such as *using a maximum of three words* give more control over the style of the generated outputs. Further, the LLM is instructed to generate keywords summarizing the detailed descriptions. The detailed descriptions are used in image editing, whereas the keywords are used to replace the caption. This strategy ensures a relatively precise image caption, as well as adding more fine-grained details to the visual scene in the image generation stage. Our approach can automate the entire process by conveniently utilizing the ChatGPT API.

Note that this generation process is open-ended and can synthesize an arbitrary number of data samples. Being able to train on a large dataset is where the power of contrastive learning comes from. This is as opposed to supervised methods whose training is restricted to the number of labeled samples which are expensive and tedious to collect.

3.1.3 Caption Editing

We replace the object in the original source caption with the newly generated phrase. For example, in Fig. 2, we replace the seagull with a bald eagle to create a caption for the newly generated image. Specifically, the source human-annotated caption *a seagull flying over the water near a large ship* is changed into *a bald eagle flying over the water near a large ship*. However, Tag2Text may produce tags, e.g., *city*, that are not presented in the source caption. For



Modify each of the given words in three different ways.
You can add colors, shapes and material information whenever necessary.
You can change the word with semantically similar words.
Add a short description of the portrayal up to three words in parenthesis.

Input:
Objects: bread, bag
Context: a person holding a piece of bread with bananas and a bag

Output:
Bread:
1) A warm, freshly baked loaf (freshly baked loaf)
2) A crumbly gluten-free muffin (gluten-free muffin)
Bag:
1) A leather satchel with intricate patterns (leather satchel)
2) A sturdy canvas backpack with patches (canvas backpack)

Input:
Objects: seagull, water, city, ship
Context: a seagull flying over the water near a large ship

Output:

Seagull:
1) a majestic bald eagle (bald eagle)
2) a colorful parrot (parrot)
Water:
1) a serene mountain lake (mountain lake)
2) a rushing river cutting through a rocky canyon (rocky canyon)
...

Figure 3. Text variation generation by LLM, explained in Subsection 3.1.2. Following the pattern defined in the prompt which changes the objects bird and rock into different instances with rich attributes, LLM completes the text for the test sample, i.e., transforming water into different types of water.

that, the caption produced by the Tag2Text model is used to generate the ground truth caption for the modified image. For example, if the tag *city* is augmented to *historic town* by the LLM (explained in Subsection 3.1.2), we label the augmented image with caption: *a seagull in the ocean near a harbor with a ship and a historic town in the background.* We label the generated image with our edited caption as a ground-truth image-text pair.

3.1.4 Image Editing

To enable fine-grained modification of an image region, we adopt the concept of image inpainting for transforming the original object in the image into the target object. In this scenario, image inpainting involves removing a specific region and filling it with content that seamlessly integrates with the image's context while considering the input information. The inpainting model takes multiple inputs, as illustrated on the right side in Fig. 2. One of the inputs is the binary mask that identifies the object region that is desired to be replaced with a new object in the source image (See Subsection 3.1.1). The other important input to the inpainting model is the object description obtained from the LLM, which indicates the target object we want the region to be changed into (See Subsection 3.1.2). The model's output is a modified image that aims to replace the content with the desired object description while the remaining parts are unchanged. This process ensures that the modified images are realistic and similar to the original images.

3.1.5 Filtering

For each image, we randomly sample M objects to input to the LLM, which subsequently generates K text variations for each selected object. We employ filters to eliminate certain examples, e.g., wrong segmentation mask, missing parts in segmentation, confusion due to multiple objects, or the text is not descriptive enough. To address these issues, we utilize BLIP's [25] ITM head, which outputs a confidence value if a given image and text pair matches. Notice that generated image might be too noisy that it barely changes even in pixels, to filter the generated images, we first calculate the standard deviation across them followed by averaging. Then we calculate the average of standard deviance within channels. Images are removed if the difference is smaller than a threshold value. More implementation details are in Appendix A.

3.2. Finetuning Framework

3.2.1 Preliminary: Contrastive Loss

The CLIP model operates on a pair of image I and text T, encoding them separately into embedding space $\mathbb{R}^d$, denoted as $e_I = \mathcal{E}_I(I)$ and $e_T = \mathcal{E}_T(T)$. The image-text similarity score is computed as

$$S(T, I) = \exp(\frac{e_T^T e_I/\tau}{||e_T||^2||e_I||^2}),$$

where temperature τ is a learnable parameter. During the training process, we sample a batch of N pairs of images and texts from the training dataset. The training objective aims to maximize the similarity between matched pairs and minimize the similarity between unmatched pairs. This is achieved through the contrastive loss, formulated as

$$\mathcal{L} = \sum_i^N log(\frac{S(T_i, I_i)}{\sum_j^N S(T_i, I_j)}) + log(\frac{S(T_i, I_i)}{\sum_k^N S(T_k, I_i)}).$$

The first part of the loss ensures that for each text T_i, we increase the similarity to its paired image I_i while decreasing the similarity to the remaining images in the batch. Similarly, the second part iterates over texts for a sampled image, encouraging similarity to the paired text and discouraging similarity to other texts in the batch. By employing this loss function, we finetune CLIP on our dataset.

3.2.2 Mixing of Hard Negative Examples

As our data generation method is unsupervised, it may produce images that do not correspond well with the expected text. To mitigate the negative impact of noise in the dataset, we combine the generated samples with the original human-annotated COCO image-text pairs. Moreover, these image-text pairs, despite being of a smaller scale compared to the dataset used for pertaining CLIP, serve as a valuable resource to prevent the model from overfitting. We employ a simple strategy to sample a batch. For a batch, we sample rN pairs from our generated data and $(1 - r)N$ pairs from the original COCO dataset. These pairs are then concatenated to form a single batch of size N.

4. Experiments

This section describes the experimental setups, including the datasets used for evaluation, implementation details of the finetuning pipeline, evaluation metrics, and baseline models for a comprehensive comparison.

Datasets We evaluate our model on composition-oriented benchmarks of different scales and different compositional aspects. The following benchmarks are included in our experiments. 1) Winoground is a hand-crafted dataset of 800 image-text pairs, for each set of two texts, the texts have exactly the same words but with different word orders, the texts are mapped to two visually distinct images. 2) ARO has more than 50,000 test images paired with automatically built text examples with changed attributes, relationships, and word order, leveraging VG [21], COCO [27], and Flickr [48]. 3) CREPE introduces new negative texts for existing images in CC-12M [3], YFCC-5M [41], LAION-400M [40], where the number of changed words in the text is gradually increased, treated as different levels of complexity. For ARO and CREPE, their texts are generated with unique methods which are not covered in our training data. This makes them perfect candidates to verify the generalization ability of our approach. For the ablation study, we primarily perform experiments on Winoground, as it is manually verified and more challenging [7], since each text in this dataset has a corresponding hard negative image with complex semantics.

Our training dataset is generated based on the COCO dataset, which has a training split with 110k image-text pairs. In our experiments, we created variations for 12.656 unique images, where for each image, we selected approximately three objects on average and generated four text variations for each object. After filtering out low-quality generations, we ended up with 82.010 image and text pairs. We generate our test set from the COCO Karpathy test split [19] with 5k image-text pairs. Additionally, we manually verify the generation to better inform the model selection and more importantly, serve as sources for the community to facilitate visual language research. For our test set, we have 278 unique images with different image-text pairs for each of them due to our manual filtering. In the end, we obtained 122 images with 4 variations for each image, 139 images with 3 variations, and 17 images with 2 variations.

Implentation Details For dataset generation, we utilize the public implementation of Tag2Text[1], Grounded-SAM[2], and Stable Diffusion[3]. For finetuning, we follow the strategies in similar work [31, 49] and combine our generated sample with human-annotated labels. We set the ratio of real and synthetic data as $r = 0.5$. We utilize the OpenAI CLIP ViT/B-32 architecture. We use a batch size of 400, a learning rate of 1e-6, and a weight decay of 0.2, and fine-tune the model for 20 epochs. We employ the default image augmentation techniques that were used during pretraining CLIP. The experiments are conducted on an Nvidia A10G GPU with 24GB memory.

Evaluation We adopt the evaluation metric for different datasets, which are basically formulated as image-to-text and text-to-image retrieval tasks. For Winoground, we report image score, text score, and group score, meaning that the model should correctly choose the text among the two text candidates for each image. For CREPE, we report the hits@1 image-to-text retrieval score on the productivity set with complexity ranging from 4 to 12. For ARO, we employ their evaluation metric and report the mean of the performance for each subcategory.

5. Results

In this section, we present the comprehensive experimental results of our proposed method in comparison to the baseline across a diverse range of tasks. Moreover, we further verify the effectiveness of our generated dataset through an extensive ablation study.

Model	Text Score	Image Score	Group Score
CLIP	30.75	11.0	8.75
Ours	**34.25**	**12.5**	**10.0**
Relative Gains	+11.1%	+13.6%	+14.2%

Table 1. Comparison of our method with CLIP on Winoground benchmarks. We report the text score, image score, and group score which measure if the model can correctly match a text for an input image, or vice versa. The best performance is shown in **bold**. Our finetuned CLIP surpasses the baseline model by a substantial margin.

5.1. Evaluation on Visuo-Linguistic Benchmarks

Table 1, 2, 3 provide a detailed comparison of our finetuned CLIP model on our generated hard negatives with the released COCO checkpoints. We evaluate the model performance across a wide range of visual language reasoning

[1] https://tag2text.github.io/

[2] https://github.com/IDEA-Research/Grounded-Segment-Anything

[3] https://github.com/CompVis/stable-diffusion

	Compositional (171)			Complex (78)		
CLIP	31.58	11.70	9.36	23.08	6.41	3.85
Ours	**38.01**	**14.62**	**10.53**	**29.49**	**8.97**	**6.41**
Gains	+22.5%	+27.2%	+12.5%	+23.9%	+39.9%	+66.5%
	Unusual Image (56)			Unusual Text (50)		
CLIP	26.79	**8.93**	5.36	**34.0**	**14.0**	**10.0**
Ours	**28.57**	**8.93**	**8.93**	30.0	10.0	**10.0**
Gains	+6.7%	0.0%	+66.3%	-11.8%	-28.5%	0.0%

Table 2. Comprehensive analysis of method performance on Winoground subsets [7] which evalute distinct reasoning abilities. Numbers in the parenthesis indicate the number of samples in that subcategory. Our model excels in compositional reasoning tasks, while it may face challenges in tasks that require an understanding of unusual text which entails background knowledge.

tasks with various aspects of reasoning ability. Our findings demonstrate that our method outperforms CLIP by a substantial margin in the majority of these tasks. Specifically, in Tab. 1, we highlight the significant improvement achieved by our method in terms of complex image text matching tasks. Furthermore, we achieved a relative improvement of 22.5% in text score and 27.2% in image score on the compositional split filtered by [7]. This subset ensures image-text matching with only compositional ability, instead of other abilities such as visual difficulty. We report the performance of our method on detailed subcategories of Winoground with most test cases in Tab. 2.Our approach exhibits lower performance on the unusual text subset, which emphasizes understanding the nuanced meaning of the text. For example, matching *the brave in the face of fear* with an image that depicts a small cub confronting a fierce lion, is challenging for our approach. The presence of repetitive text samples in our augmented dataset may impact the finetuning of the text encoder (see Appendix B).

	ARO [49]			CREPE [29]		
Model	Attribute	Relation	Order	Atom	Swap	Negate
CLIP	0.59	0.62	**0.48**	0.20	**0.19**	**0.35**
Ours	**0.65**	**0.65**	0.45	**0.23**	**0.19**	0.13
Gains	+10.1%	+4.8%	-6.3%	+ 15.0%	0.0%	-31.5%

Table 3. Comparison of our method with the baseline on ARO and CREPE datasets for text retrieval. Our method can discriminate texts which can be mapped to real scenes with different semantics (ARO-Attribute, ARO-Relation, CREPE-Atom) but struggles with linguistic phenomenons such as negation *not* in CREPE, or grammatically incorrect sentences in ARO-Order.

Table 3 presents a comprehensive comparison of the performance between our CLIP and the baseline on the ARO and CREPE datasets. This analysis of the ARO dataset reveals an interesting phenomenon, wherein our method performs significantly better in the attribute category, e.g., matching the adjective *white* to the object *dog*, while demonstrating comparatively low performance in the order category, wherein the word order is randomly changed, e.g., *a white cat* into *cat a white*. This outcome is expected as our training data does not encompass sentences that are grammatically incorrect. Tab. 3 confirms similar findings for CREPE. Our model demonstrates a significant improvement, especially in the atom category, where the objects and their attributes are changed. However, our model struggles with the negate category, such as transforming a dog into ***not*** *a dog*. This outcome is expected as our training dataset lacks such examples.

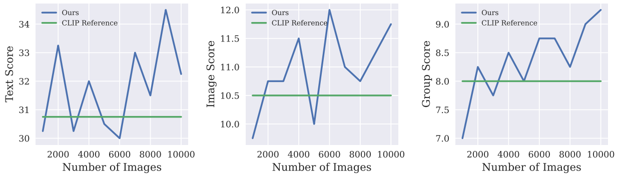


Figure 4. Model performance with increasing numbers of samples. Finetuning our model on incrementally increased generated data shows a consistent trend: as the data size grows, the model's performance is improved. This suggests the potential for generating more training examples to further enhance the model. Note the x-axis shows the number of unique images.

	VG			SWIG	VAW
Data	Object	Attribute	Relation	Object	Attribute
CLIP	79.0	69.8	58.2	71.8	65.7
Ours	**85.1**	70.7	53.8	75.8	66.4
TSVLC [9]	82.8	**75.5**	**62.6**	**78.2**	**68.4**

Table 4. Comparison of our method with CLIP and TSVLC on VL-Checklist. The scores are obtained by averaging each subcategory within object, attribute, and relation.

Similar to previous findings, Table 4 presents the improvements of our approach over CLIP on the VL-Checklist dataset in the object and attribute categories. The performance in the relation category is decreased as expected. Furthermore, we compare our method to a state-of-the-art approach proposed in TSVLC, which solely utilizes text augmentations. It is crucial to note that the comparison is not fair as the SOTA approach has been trained on a much larger dataset with a larger batch size and curated losses. Nevertheless, our model demonstrates comparable performance to that approach. A comprehensive analysis of detailed subsets of the VL-Checklist is in Appendix D.

5.2. Ablation Study

The influence of human-curated dataset COCO. Table 5 provides a comparison between our generated dataset and the use of only the COCO dataset, which contains ground truth image text pairs from human annotators. The COCO image text pairs are part of our constructed dataset,

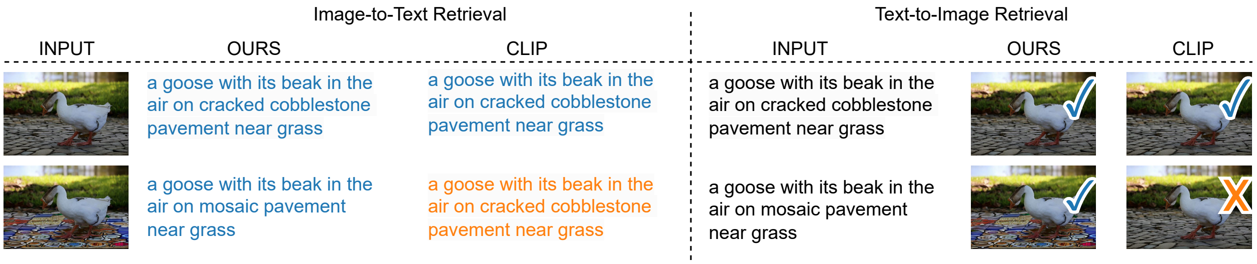


Image-to-Text Retrieval
Text-to-Image Retrieval
INPUT
OURS
CLIP
INPUT
OURS
CLIP
a goose with its beak in the air on cracked cobblestone pavement near grass
a goose with its beak in the air on cracked cobblestone pavement near grass
a goose with its beak in the air on cracked cobblestone pavement near grass
a goose with its beak in the air on mosaic pavement near grass
a goose with its beak in the air on cracked cobblestone pavement near grass
a goose with its beak in the air on mosaic pavement near grass

Figure 5. Examples of retrieved text for a given image (left) and retrieved images for a given text (right) by our method and the baseline. Correct matches are shown in blue, while the incorrect predictions are marked in orange.

and it was not clear if CLIP has incorporated COCO in its pretraining. To ensure the rigor of our analysis and examine the effect of existing labeled image-text pairs, we conduct an experiment comparing the performance of our method with CLIP that is finetuned only on the COCO dataset. Even though finetuning on COCO bring marginal improvement, our improvement is much more significant. This supports the assumption that the existing dataset may not contain sufficient hard negative examples.

Model	Text	Image	Group
CLIP	30.75	11.0	8.75
CLIP-COCO	30.75	**12.5**	9.5
Ours	**34.25**	**12.5**	**10.0**

Table 5. The influence of training data on the model performance on the Winoground dataset. CLIP-COCO is a finetuned CLIP model using our finetuning protocol on the COCO dataset. Ours denotes our final model finetuned on the mixture of our generated dataset and COCO.

Impact of the number of generated samples. In Fig. 4, we analyze the impact of the data size of our generated samples on the Winoground performance. As seen from the figure, when the number of generated data samples increases, our model's performance on the image-text reasoning task improves. This demonstrates the value of incorporating more data in training to enhance the model's capabilities. This highlights the advantage of our method which utilizes large generative models, such as LLMs and text-to-image models, to generate high-quality examples, essential for tackling the vast image space. Due to hardware issues, our experiments are conducted until the data scale shown in the figure. Nonetheless, the results reveal a clear upward trend, indicating the potential for further improvements with a larger dataset.

5.3. Qualitative Results

Fig. 5 shows a qualitative comparison of our model and the naive CLIP model on our test benchmark. For each input image or text, the most similar text or images among the two candidate texts or images are found by our method and CLIP. While our method can distinguish the details of the image, the CLIP model fails on this task. We report the performance of our models on our test set in Tab. 6.

Model	Text All	Image All	Group All	Text 1	Image 1	Group 1
CLIP	21.51	20.79	10.75	60.21	57.87	40.64
Ours	**27.96**	**24.01**	**13.62**	**62.23**	**60.21**	**43.19**

Table 6. Comparison of our finetuned model to the CLIP baseline on our generated test set, evaluated in a similar metric as Winoground [42].

6. Limitations

Our work is limited by the performance of generative models such as ChatGPT and Stable Diffusion. We rely on the capacity of such models to produce high-quality examples. The diversity is mostly constrained by the power of LLMs. Additionally, we primarily manipulate local features such as objects and background, while may restrict the scope of negative examples generated, e.g., manipulating the relationship between two objects. Addressing these limitations is crucial for future improvements.

7. Conclusion

Our work tackles the limitations of existing visual language models in terms of compositional reasoning between text and images. We proposed a data generation pipeline that leveraged generative models to introduce challenging negative examples required for contrastive learning. Our proposed method effectively improves the compositionality and discriminative capabilities of VLMs. Experimental results demonstrate that training with our method consistently outperforms existing VLMs on various compositional reasoning benchmark datasets. This was done by addressing the scarcity of hard negative examples for both the image and text modalities. Our work highlights the importance of generative approaches in advancing the field of visual language understanding and bridging the gap between humans and VLMs on compositional reasoning tasks.

Acknowledgement This work was supported by the ERC Advanced Grant SIMULACRON.

References

[1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 3

[2] Paola Cascante-Bonilla, Khaled Shehada, James Seale Smith, Sivan Doveh, Donghyun Kim, Rameswar Panda, Gül Varol, Aude Oliva, Vicente Ordonez, Rogerio Feris, et al. Going beyond nouns with vision & language models using synthetic data. *arXiv preprint arXiv:2303.17590*, 2023. 3

[3] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3558–3568, 2021. 6

[4] Long Chen, Xin Yan, Jun Xiao, Hanwang Zhang, Shiliang Pu, and Yueting Zhuang. Counterfactual samples synthesizing for robust visual question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10800–10809, 2020. 3

[5] Zhenfang Chen, Peng Wang, Lin Ma, Kwan-Yee K Wong, and Qi Wu. Cops-ref: A new dataset and task on compositional referring expression comprehension. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10086–10095, 2020. 2

[6] Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. *Advances in neural information processing systems*, 33:8765–8775, 2020. 3

[7] Anuj Diwan, Layne Berry, Eunsol Choi, David Harwath, and Kyle Mahowald. Why is winoground hard? investigating failures in visuolinguistic compositionality. *arXiv preprint arXiv:2211.00768*, 2022. 3, 6, 7

[8] Sivan Doveh, Assaf Arbelle, Sivan Harary, Amit Alfassy, Roei Herzig, Donghyun Kim, Raja Giryes, Rogerio Feris, Rameswar Panda, Shimon Ullman, et al. Dense and aligned captions (dac) promote compositional reasoning in vl models. *arXiv preprint arXiv:2305.19595*, 2023. 3

[9] Sivan Doveh, Assaf Arbelle, Sivan Harary, Eli Schwartz, Roei Herzig, Raja Giryes, Rogerio Feris, Rameswar Panda, Shimon Ullman, and Leonid Karlinsky. Teaching structured vision & language concepts to vision & language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2657–2668, 2023. 2, 3, 7

[10] Lijie Fan, Dilip Krishnan, Phillip Isola, Dina Katabi, and Yonglong Tian. Improving clip training with language rewrites. *arXiv preprint arXiv:2305.20088*, 2023. 3

[11] Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. *Advances in Neural Information Processing Systems*, 35:30583–30598, 2022. 3

[12] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020. 2

[13] Ruifei He, Shuyang Sun, Xin Yu, Chuhui Xue, Wenqing Zhang, Philip Torr, Song Bai, and Xiaojuan Qi. Is synthetic data from generative models ready for image recognition? *arXiv preprint arXiv:2210.07574*, 2022. 3

[14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2

[15] Xinyu Huang, Youcai Zhang, Jinyu Ma, Weiwei Tian, Rui Feng, Yuejie Zhang, Yaqian Li, Yandong Guo, and Lei Zhang. Tag2text: Guiding vision-language model via image tagging. *arXiv preprint arXiv:2303.05657*, 2023. 3

[16] Yufeng Huang, Jiji Tang, Zhuo Chen, Rongsheng Zhang, Xinfeng Zhang, Weijie Chen, Zeng Zhao, Tangjie Lv, Zhipeng Hu, and Wen Zhang. Structure-clip: Enhance multi-modal language representations with structure knowledge. *arXiv preprint arXiv:2305.06152*, 2023. 2, 3

[17] Tri Huynh, Simon Kornblith, Matthew R Walter, Michael Maire, and Maryam Khademi. Boosting contrastive self-supervised learning with false negative cancellation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2785–2795, 2022. 3

[18] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR, 2021. 3

[19] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015. 6

[20] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023. 3

[21] 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*, 123:32–73, 2017. 6

[22] Varun Kumar, Ashutosh Choudhary, and Eunah Cho. Data augmentation using pre-trained transformer models. *arXiv preprint arXiv:2003.02245*, 2020. 3

[23] Hang Li, Jindong Gu, Rajat Koner, Sahand Sharifzadeh, and Volker Tresp. Do dall-e and flamingo understand each other? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1999–2010, 2023. 2

[24] 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*, 2023. 2

[25] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR, 2022. 2, 3, 5

[26] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34:9694–9705, 2021. 1, 2, 3

[27] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 3, 6

[28] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 3

[29] Zixian Ma, Jerry Hong, Mustafa Omer Gul, Mona Gandhi, Irena Gao, and Ranjay Krishna. Crepe: Can vision-language foundation models reason compositionally? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10910–10921, 2023. 2, 3, 7

[30] Ron Mokady, Amir Hertz, and Amit H Bermano. Clipcap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734*, 2021. 2, 3

[31] Liliane Momeni, Mathilde Caron, Arsha Nagrani, Andrew Zisserman, and Cordelia Schmid. Verbs in action: Improving verb understanding in video-language models. *arXiv preprint arXiv:2304.06708*, 2023. 2, 3, 6

[32] Minheng Ni, Zitong Huang, Kailai Feng, and Wangmeng Zuo. Imaginarynet: Learning object detectors without real images and annotations. *arXiv preprint arXiv:2210.06886*, 2022. 3

[33] Filip Radenovic, Abhimanyu Dubey, Abhishek Kadian, Todor Mihaylov, Simon Vandenhende, Yash Patel, Yi Wen, Vignesh Ramanathan, and Dhruv Mahajan. Filtering, distillation, and hard negatives for vision-language pre-training. *arXiv preprint arXiv:2301.02280*, 2023. 3

[34] 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. 1, 3

[35] 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. 2

[36] Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. *arXiv preprint arXiv:2010.04592*, 2020. 2, 3

[37] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 2, 3

[38] Madeline Chantry Schiappa, Michael Cogswell, Ajay Divakaran, and Yogesh Singh Rawat. Probing conceptual understanding of large visual-language models. *arXiv preprint arXiv:2304.03659*, 2023. 2, 3

[39] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *arXiv preprint arXiv:2210.08402*, 2022. 1, 3

[40] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021. 1, 3, 6

[41] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016. 6

[42] 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. 2, 3, 8

[43] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 4

[44] Brandon Trabucco, Kyle Doherty, Max Gurinas, and Ruslan Salakhutdinov. Effective data augmentation with diffusion models. *arXiv preprint arXiv:2302.07944*, 2023. 3

[45] Haonan Wang, Minbin Huang, Runhui Huang, Lanqing Hong, Hang Xu, Tianyang Hu, Xiaodan Liang, and Zhenguo Li. Boosting visual-language models by exploiting hard samples. *arXiv preprint arXiv:2305.05208*, 2023. 3

[46] Weijia Wu, Yuzhong Zhao, Mike Zheng Shou, Hong Zhou, and Chunhua Shen. Diffumask: Synthesizing images with pixel-level annotations for semantic segmentation using diffusion models. *arXiv preprint arXiv:2303.11681*, 2023. 3

[47] Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. Filip: Fine-grained interactive language-image pre-training. *arXiv preprint arXiv:2111.07783*, 2021. 3

[48] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014. 6

[49] 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*, 2023. 2, 3, 6, 7

[50] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Revisiting visual representations in vision-language models. In *Proceedings of the IEEE/CVF conference on*

computer vision and pattern recognition, pages 5579–5588, 2021. 1

[51] Wenzheng Zhang and Karl Stratos. Understanding hard negatives in noise contrastive estimation. *arXiv preprint arXiv:2104.06245*, 2021. 3

[52] 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. 3