CLoVe: Encoding Compositional Language in Contrastive Vision-Language Models

Anonymous ACL submission

Abstract

Recent years have witnessed a significant increase in the performance of Vision and Language tasks. Foundational Vision-Language Models (VLMs), such as CLIP, have been leveraged in multiple settings and demonstrated remarkable performance across several tasks. Such models excel at object-centric recognition yet learn text representations that seem invariant to word order, failing to compose known concepts in novel ways. However, no evidence exists that any VLM, including large-scale single-stream models such as GPT-4V, identifies compositions successfully. In this paper, we introduce a method to significantly improve the ability of existing models to encode compositional language, with over 10% absolute improvement on standard benchmarks, while maintaining the performance on more standard benchmarks. In this paper, we present a method to considerably improve the compositionality of CLIP-like pre-trained models while preserving its performance on other tasks. We will provide model weights that can be used to swap existing CLIP-like weights and have a noticeable boost in compositional performance.

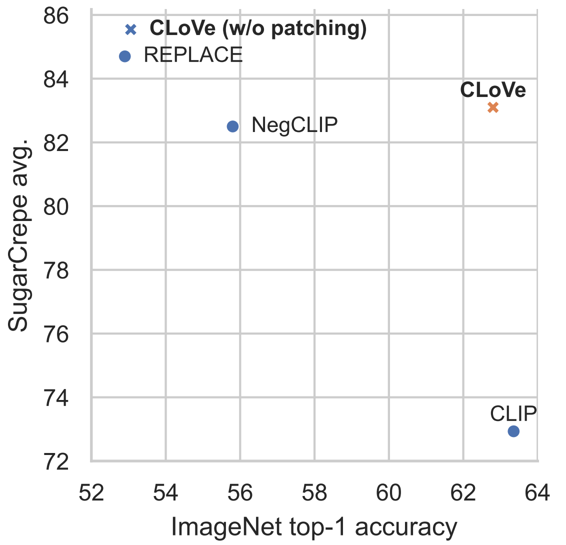

Figure 1: Our proposed method CLoVe significantly improves the compositionality performance (as measured by an average of SugarCrepe's seven fine-grained tasks) of pre-trained CLIP-like models while preserving their performance on other downstream tasks (as measured by ImageNet). Comparisons with more benchmarks are presented in Tables 3 and 4. Baselines: REPLACE (Hsieh et al., 2023) and NegCLIP (Yuksekgonul et al., 2023).

1 Introduction

There has been a significant increase in the performance of Vision and Language tasks over the last few years (Radford et al., 2021; Jia et al., 2021; Rombach et al., 2022; Alayrac et al., 2022; Laurençon et al., 2023). Vision-Language Models (VLMs), such as CLIP (Radford et al., 2021), have been leveraged in multiple settings, either directly or indirectly as foundational models, and demonstrated remarkable performance across several tasks (Bommasani et al., 2021; Ramesh et al., 2021, 2022; Rombach et al., 2022; Castro and Caba, 2022; Li et al., 2023).

Such models excel at object-centric recognition yet learn text representations that seem invariant to word order (Thrush et al., 2022; Yuksekgonul et al., 2023; Castro et al., 2023), failing to compose known concepts in novel ways (Ma et al., 2023; Hsieh et al., 2023). For example, as shown in Figure 1, CLIP has top performance on ImageNet tasks but falls behind on compositionality benchmarks.

Language compositionality is essential to recognizing more complex concepts in images or making text-to-image models successfully generate a novel scene with specific constraints (Hafri et al., 2023). For instance, in an image depicting *"the woman shouts at the man,"* it is important to recognize who is shouting at whom to understand the scene correctly.

Yet, no evidence exists that any VLM, including large-scale single-stream models such as GPT-4V (OpenAI, 2023), identifies compositions successfully. This assertion is supported by the fact that existing benchmarks that test compositionality continue to be an open challenge (Thrush et al., 2022; Yuksekgonul et al., 2023; Ma et al., 2023; Hsieh et al., 2023).[1]

To address these limitations, previous work has introduced methods to increase the compositional capabilities of pre-trained VLMs, such as NegCLIP (Yuksekgonul et al., 2023) and REPLACE (Hsieh et al., 2023). However, such methods come at a significant cost: they sacrifice the performance on more common object-centric recognition, as measured by ImageNet (Deng et al., 2009), EuroSAT (Helber et al., 2019, 2018), and CIFAR100 (Krizhevsky, 2009). For instance, as shown in Figure 1, NegCLIP showed an increase (compared to the pre-trained model) in its ability to address SugarCrepe (Hsieh et al., 2023) compositionality benchmark from 72.9% to 82.5% while, at the same time, its performance on ImageNet (Deng et al., 2009) top-1 accuracy dropped from 63.4% to 55.8%. Similarly, Hsieh et al. (2023) applied REPLACE to reach a high score of 84.7% on SugarCrepe, but at the cost of a significant drop to 52.9% on its ImageNet accuracy.

In this paper, we introduce a method to significantly improve the ability of existing two-tower models to encode compositional language while keeping the performance on more standard benchmarks, as shown in Figure 1. Specifically, our contributions are as follows. First, we show that **data curation** can significantly impact how a model can handle compositional knowledge. Second, we confirm that training along with **hard negatives** can bring additional improvements. Third, we show experimentally that **model patching** can be employed to preserve model performance on previous tasks. Finally, we combine these ideas into a new model called CLoVE and show that it can **significantly improve compositionality over a contrastively pre-trained VLM** such as CLIP while maintaining the performance on other tasks. Upon publication, we will provide checkpoints that others can use to substitute their CLIP-like model weights for a version with significantly better language composition abilities.

[1] See Section 2 for details.

2 Related Work

Benchmarking Compositionality. Several frameworks have been proposed to measure model performance on language compositionality. Shekhar et al. (2017) crafted a benchmark of foil image captions generated by changing a single word from the correct captions. Models must identify if the image-caption pair correspond to each other, among other tasks. Winoground (Thrush et al., 2022) carefully built a high-quality dataset of 400 examples, each consisting of two images and two captions. These two captions contain the exact word but in a different order following one of several strategies (e.g. swapping the subject and the object). Each image must match the correct caption for the models to pass this test. Models cannot simply rely on their ability to recognize concepts in images, as the elements repeat but are composed differently.

Diwan et al. (2022) found that passing the Winoground benchmark successfully requires composition skills along with many others, such as commonsense reasoning and locating tiny objects. Yuksekgonul et al. (2023) argued that Winoground is too small to draw statistically significant conclusions and built a benchmark called ARO consisting of examples with a single image, a correct caption, and multiple automatically-generated incorrect captions. CREPE (Ma et al., 2023) crafted a benchmark to measure compositionality in terms of systematicity and productivity. It considers both seen and unseen compounds, among other phenomena. SugarCrepe (Hsieh et al., 2023) is a recent benchmark that avoids ungrammatical and nonsensical negative captions while being large. They showed it cannot be easily solved by computing the probability of the text captions without looking at the image. Other benchmarks have also been created that consider compositionality as well as other phenomena, such as VALSE (Parcalabescu et al., 2022), RareAct (Miech et al., 2020), VL-Checklist (Zhao et al., 2022), Cola (Ray et al., 2023), SVO-Probes (Hendricks and Nematzadeh, 2021), and CLEVR (Johnson et al., 2017).

Methods to Improve Compositionality. Several works have shown that VLMs cannot recognize compositions successfully (Shekhar et al., 2017; Miech et al., 2020; Parcalabescu et al., 2022; Thrush et al., 2022; Hendricks and Nematzadeh, 2021; Yuksekgonul et al., 2023; Castro et al., 2023; Ma et al., 2023). For this reason, NegCLIP (Yuksekgonul et al., 2023) was proposed to improve how CLIP (Radford et al., 2021) composes concepts. It consists of adding hard negative texts by taking the captions from the training batch and automatically generating sentences with the exact words but in a different order. This approach makes the model distinguish between an image and the caption in the correct order compared to the exact words in an arbitrary order (as well as the other negative captions within the batch). Hsieh et al. (2023) build upon NegCLIP and CREPE (Ma et al., 2023) and propose three ways to generate random negatives: REPLACE, SWAP, and NEGATE. All these methods start from a Scene Graph representation of the sentence and operate over it. REPLACE, which had the best overall results, performs single-atom replacements. SWAP exchanges two atoms within the scene graph. Finally, NEGATE introduces negation words (i.e., *no* or *not*). We build upon NegCLIP (Yuksekgonul et al., 2023) and REPLACE (Hsieh et al., 2023) while we propose to use synthetically-generated captions to scale them up, as well as applying model patching (Ilharco et al., 2022) to avoid catastrophic forgetting. As far as we know, we introduce the first method that significantly improves the composition skills of contrastively-trained models while preserving their zero-shot performance on other downstream tasks.

Cap and CapPa (Tschannen et al., 2023) are two recently introduced methods that employ captioning instead of contrastive learning (as in CLIP) to train VLMs. Tschannen et al. (2023) showed that these methods present an excellent performance on compositionality as measured by ARO (Yuksekgonul et al., 2023) and SugarCrepe (Hsieh et al., 2023). As these methods rely on captioning and thus on computing the probability of the text given an image, they are inefficient for retrieval and classification. For ARO, they showed that they can achieve high performance without looking at the image (they call it a "blind decoder"). For SugarCrepe, the authors did not compute this specific baseline. Hence, we cannot infer the extent to which these models handle compositions successfully. Our method is different from them as it builds on top of CLIP-like two-tower models, which are efficient for retrieval and classification, and it does not rely on computing the probability of text, which is generally unimportant for such settings as all texts are equally likely (unlike in image captioning).

3 Increasing Compositionality in Contrastive VLMs

To address the compositionality limitations observed in previous models, we propose strategies to address the three main aspects of developing a contrastive VLM: data curation, contrastive learning, and model tuning. We introduce CLoVe, a model that leverages the strengths of an existing pre-trained contrastive VLM and enhances it with language composition skills. Figure 2 shows an overview.

CLoVe includes the following steps, presented in more detail below:

3.0 Pre-trained Model. Our goal is to improve the compositionality of an existing pre-trained VLM. We select a pre-trained CLIP model or pre-train one as an initial step.

3.1 Synthetic Captions. Synthetic data generation can be effectively used to enlarge the training data. We use a large dataset with synthetic captions.

3.2 Hard Negatives. Contrastive VLMs rely on the availability of negative training data. We add randomly generated hard text negatives to the dataset and train a fine-tuned model with increased compositionality capabilities.

3.3 Model Patching. The pre-trained model and the fine-tuned model are combined through model patching. Patching allows us to keep the compositionality obtained with the fine-tuned model while recovering the pre-trained model performance on previously supported tasks.

3.0 Pre-trained Model

Rather than starting from scratch, we aim to enhance the composition capabilities of an existing contrastive VLM. This work uses CLIP (Contrastive Language-Image Pre-training; Radford et al., 2021), a pre-training method demonstrating impressive zero-shot performance on classification and retrieval tasks involving vision or language. It involves learning image and text representations in a joint space by leveraging large-scale weakly-supervised datasets. These datasets contain image-text pairs with varying degrees of correspondence. For each image, the model must learn the corresponding positive text from a set that includes this text and a random sample of $N-1$ other texts (negative samples) by employing the InfoNCE objective (Oord et al., 2018). Similarly, the model

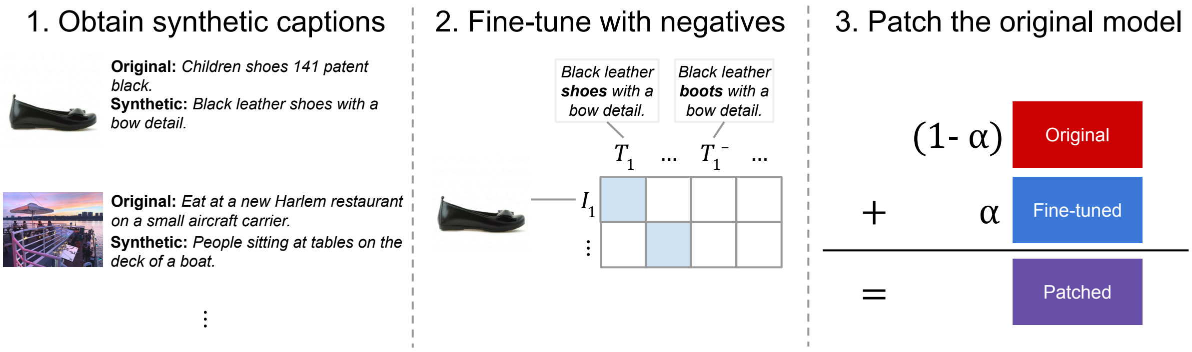

Figure 2: Our CLOVE method consists of three steps. First, obtain synthetic captions for a large image dataset. Second, fine-tune a pre-trained CLIP-like model on it along with hard negative texts. Third, patch the original model with the fine-tuned one.

must identify which image corresponds to a given text. CLIP is trained with mini-batch gradient descent, where this objective is applied to each pair in the N-sized batch, and the negatives are typically sourced from the rest of the batch.

3.1 Synthetic Captions

Synthetic captions provide a great hybrid between the training dataset size and the quality of the captions. We leverage LAION-COCO (Schuhmann et al., 2022b), a 600-million dataset with images from the 2-billion-sized English subset of LAION-5B (Schuhmann et al., 2022a) that were captioned with BLIP ViT-L/14 (Li et al., 2022), which was fine-tuned on COCO and filtered with two versions of OpenAI-pre-trained CLIP (Radford et al., 2021; ViT-L/14 and RN50x64). Even though the captions are limited in style (typically following the style of COCO captions), the LAION-COCO authors found that the synthetically generated captions have a similar quality to those written by humans. We believe these captions focus more on describing visual information than the captions from its original dataset (LAION), based on multiple examples from this dataset. See Section 4.1 for an ablation of the training dataset.

3.2 Hard Negatives

Yuksekgonul et al. (2023) proposed NegCLIP, an extension of CLIP's training procedure that generates a hard negative text for each example in the batch by rearranging the image caption words. These generated negatives are included within the negative test sets of the learning objective. Hsieh et al. (2023) proposed an alternative called REPLACE and showed that the model can achieve better compositionality skills if such negatives are generated from carefully selected single-word replacements. These replacements are performed on one of the entities, relations, or attributes obtained from first parsing the sentence as a scene graph, then selecting an alternative word from its antonyms or co-hyponyms by leveraging WordNet (Fellbaum, 2010)[2]. These methods rely on high-quality captions. Otherwise, the generated negatives will have changes that cannot be visually appreciated or will mostly be ungrammatical or nonsensical, and the model's downstream performance will be severely affected. Take the following example from LAION that accompanies an image of a cardholder: *"5x Orange Ball Wedding Party PLACE CARD HOLDER Table Name Memo Paper Note Clip."* If we apply REPLACE, supposing we can parse the sentence correctly, the word "table" could be replaced with "bed". However, this would not make it a negative since the table is additional contextual information the caption included that cannot be visually appreciated. Such a change will introduce more noise to the model's training process.

For this reason, these works have employed the COCO captions (Lin et al., 2014; Chen et al., 2015) dataset. COCO consists of images along with high-quality human-annotated captions that describe them. Nevertheless, with 600,000 image-text pairs, COCO is at least three orders of magnitude smaller than the typically used image-text training datasets. This issue limits learning and makes models overfit. Additionally, COCO presents a limited number of objects and actions. 700 out

[2]More precisely, the method proposes to look for words that share a grand-co-hypernym.

of the 1000 object classes in ImageNet-1k are not present in COCO (Venugopalan et al., 2017). We propose combining these hard-negative techniques with a synthetic-caption dataset, such as LAION-COCO (Schuhmann et al., 2022b) (introduced in the previous subsection).

3.3 Model Patching

NegCLIP (Yuksekgonul et al., 2023) and REPLACE (Hsieh et al., 2023) make models improve significantly on language compositional skills. However, in exchange, they sacrifice the performance on general object recognition, as measured by their ImageNet performance. For this reason, we propose applying one of such methods and subsequently employing a method called "model patching" (Ilharco et al., 2022). Model patching makes a fine-tuned model recover the performance on previously supported tasks. This procedure consists of performing a weight-space average between the pre-trained and the fine-tuned models. Concretely, for each pre-trained model weight w_i^{PT} and fine-tuned model weight w_i^{FT}, we compute their weighted average to obtain a new model weight w_i:

$$w_i = (1-\alpha)w_i^{PT} + \alpha w_i^{FT} \tag{1}$$

In Section 4.3, we show that this method helps the model gain compositionality properties while maintaining its object-recognition performance.

3.4 Implementation Details

Unless otherwise noted, the implementation details are the following.

We write our code on Python 3.10 using PyTorch (Paszke et al., 2019) v2.1, starting from `open_clip`'s (Ilharco et al., 2021; Cherti et al., 2023) codebase. We run the experiments using the AdamW optimizer (Loshchilov and Hutter, 2019), with a linear learning rate warmup for 2000 steps to 1e-6, later decayed with a cosine schedule (Loshchilov and Hutter, 2017). We use a weight decay of 0.1. Our initial pre-trained model is ViT-B-32 from OpenAI (Radford et al., 2021). We train the models through one billion examples by randomly sampling with replacement from shards of up to 10 000 samples, where the final size of each depends on the image availability at download time. We successfully downloaded about 80% of LAION-400M (Schuhmann et al., 2021), 80% of LAION-COCO (Schuhmann et al., 2022b), and 60% of COYO-700M (Byeon et al., 2022) images. The text captions are in English. We employ one node with 8x A100 Nvidia GPUs and 96 CPU cores (`p4d.24xlarge` from AWS) for four days and a half. The batch size is 256 per GPU.

The choice of learning rate was based on multiple preliminary experiments to make sure it was not learning too slowly or that it was making the training loss go up. The training steps and samples were selected to ensure we gave enough time for the method to learn and converge. The choice of total batch size and compute budget was determined based on our availability compute and considering that CLIP-like methods need a large batch size. All reported experiments are based on a single run since they are computationally expensive.

We re-implemented REPLACE (Hsieh et al., 2023) with the following changes and decisions, primarily because the code for this part is unavailable. We skip employing BERT (Devlin et al., 2019) to filter the generated negatives and instead proceeded to replace words based on the frequency of the new words, which is a first-order approximation of computing probabilities with a contextualized model. For the replacements, given that the authors do not mention prepositions but we find them replaced in the provided data, we proceeded to replace prepositions. For the replacement words, we try to respect the rest of the sentence by conjugating them (e.g., the person for the verbs, and the number for the nouns) and using a similar casing to the replaced word. We used spaCy (Honnibal et al., 2020) v3.7.2 (the model `en_core_web_sm`) and `pyinflect` v0.5.1. We employed a different Scene Graph Parsing implementation, `SceneGraphParser` v0.1.0. We avoid replacing a word with a potential synonym by looking at the synsets in common of their lemmas from WordNet (Fellbaum, 2010), leveraging NLTK (Bird et al., 2009) v3.8.1. We managed to reproduce the same numbers the original authors reported. We will make our code publicly available to make it easy for anybody to reproduce and build on top of our results.

We set $\alpha = 0.6$ for the model patching based on the ablation from Section 4.3.

4 Experiments

We conduct three ablations studies and a comparison with related work on multiple benchmarks. In Section 4.2, we evaluate if employing hard negative texts during training improves the recognition performance of compositions. We compare different training datasets in Section 4.1. In Section 4.3, we test the importance of patching the original model after training with hard negative texts. Finally, in Section 4.4, we compare our method to previous ones. Unless otherwise noted, all evaluations are zero-shot, meaning we performed no in-domain fine-tuning on a benchmark-specific training split.

4.1 The Importance of Synthetic Captions

We hypothesize that training dataset quality is essential to model compositionality performance. For example, in LAION (Schuhmann et al., 2021), a dataset commonly used to train CLIP-like models, you can find examples that present excessive information that cannot be easily mapped to visual concepts depicted in any image, such as: *"Platinum Dance Academy T-shirt. Orders must be placed by Friday, September 26th. Delivery approximately 2 weeks or less."*

Datasets with high-quality annotations such as COCO (Lin et al., 2014; Chen et al., 2015) can be used. However, such datasets are typically small (less than a million samples). A hybrid approach, with high-quality data and a large dataset, can be obtained using synthetic captions, as described in Section 3.1. We are interested in comparing this dataset with LAION-400M or COCO directly, as well as two ways to combine the datasets: a) concatenation and b) sampling with equal probability.[3] Note that these ways of combining LAION and COCO differ from LAION-COCO, a different dataset (see Section 3.1). In addition, we consider COYO-700M (Byeon et al., 2022), a large-scale dataset that was constructed similarly to LAION-400M.

Table 1 compares the performance of fine-tuning a pre-trained CLIP model on different datasets without employing negatives. LAION-COCO (Schuhmann et al., 2022b) presents the best results overall, with a large margin on ARO. For this benchmark, it is the only presented dataset that significantly outperforms the pre-trained model. In the case of the SugarCrepe benchmark, we observe that all datasets provide improvements over the pre-trained model. Interestingly, Betker et al. (2023) also found synthetic captions helpful for text-to-image generation models. They show synthetic captions help such models generate images that align better with the input text.

[3]Note LAION-400M is about 700 times larger than COCO.

Fine-tuning dataset	Attr.	Rel.	C-Ord.	F-Ord.
pre-trained	63.5	59.8	47.7	59.9
Without hard negative texts				
COYO	63.6	55.4	34.8	43.4
LAION (L)	64.9	64.0	40.2	47.0
COCO (C)	62.5	61.6	**73.8**	39.8
concat. L & C	**65.9**	59.0	43.7	50.3
sample unif. L & C	64.6	55.7	59.8	29.7
LAION-COCO	65.4	**66.0**	70.5	**76.9**
With hard negative texts				
COYO	69.5	75.6	71.7	79.7
LAION (L)	67.9	72.6	78.3	85.4
COCO (C)	**70.2**	67.6	90.9	74.5
concat. L & C	70.1	76.2	83.4	88.6
sample unif. L & C	69.9	71.6	82.7	60.8
LAION-COCO	69.0	**77.4**	**91.7**	**93.6**

Table 1: The zero-shot performance of fine-tuning CLIP with different datasets, with and without hard negative texts. The best results are in **bold**. An underline indicates results within 1% of best.

	Attr.	Rel.	C-Ord.	F-Ord.
pre-trained	63.5	59.8	47.7	59.9
fine-tuned	65.4	66.0	70.5	76.9
+ negatives	69.0	**77.4**	**91.7**	**93.6**
+ negatives*	**69.4**	75.4	77.5	86.1

Table 2: The importance of employing negatives to improve the zero-shot performance on recognizing compositions. The best results are in **bold**. An underline indicates results within 1% of best. *The last row shows the results of using half the batch size – there are gains even when the total device memory is the same, given that employing negatives effectively doubles the batch size.

4.2 The Importance of Hard Negatives

Yuksekgonul et al. (2023); Hsieh et al. (2023) showed that employing randomly generated text negatives as part of the training process can significantly improve the language compositionality skills of pre-trained models. We apply REPLACE (Hsieh et al., 2023) to obtain randomly generated hard negative text along with the LAION-COCO dataset (Schuhmann et al., 2022b) and compare it to fine-tuning without negatives. We present the results in Table 2. In this setting, we can observe that employing negatives improves performance over not using them, as measured by the ARO benchmark (Yuksekgonul et al., 2023) (its tasks are, in the order that we show them: VG-Attribution, VG-Relation, COCO-Order, and Flickr30k-Order).

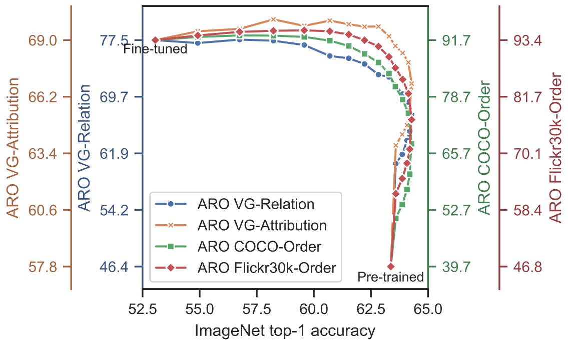

Figure 3: The effect of applying model patching to both an object-centric benchmark (ImageNet, Deng et al., 2009; x-axis) and a compositionality benchmark (ARO, Yuksekgonul et al., 2023; the four y-axes represent its four tasks), when varying the value of the weight in the average, α. The value of α varies from 0 (the pre-trained model) to 1 (the fine-tuned model) in 0.05 increments, and the lines connect such points. We can obtain models with good zero-shot performance in ImageNet and compositionality when α is around 0.4–0.7. Note the four y-axes were adjusted to make the pre-trained and fine-tuned model points match to focus on how the lines vary between them.

4.3 The importance of Model Patching

Existing methods to improve CLIP's compositionality by employing negatives used by Yuksekgonul et al. (2023); Hsieh et al. (2023) do so by considerably hurting the model's performance on more standard object-centric benchmarks such as ImageNet (Deng et al., 2009).

Figure 3 presents the effect of varying this value for both a compositionality benchmark and an object-centric one. When α is around 0.4–0.7, the model performs well on both.

4.4 CLoVe: Bringing Compositionality into CLIP

We compare our method to other baselines in Figure 1. Our method presents an average 10% absolute improvement on SugarCrepe (Hsieh et al., 2023) (over its seven fine-grained tasks), a challenging benchmark on compositionality, over a pre-trained CLIP model while having an ImageNet performance within 1%. Our method presents results comparable to other existing methods without losing ImageNet performance. Additionally, we show that our method performs better than others on compositionality when we do not apply the model patching step.

In Table 3, we show a comparison of our method with others in three compositionality benchmarks: ARO (Yuksekgonul et al., 2023), SugarCrepe (Hsieh et al., 2023) (over its three coarse-grained tasks), and SVO-Probes (Hendricks and Nematzadeh, 2021). Note that, for SugarCrepe, we employ the macro-average to compute the coarse-grained task results like in (Tschannen et al., 2023) and unlike the original paper, since we are interested in measuring the global phenomena instead of giving importance to the task sample sizes. See Appendix A for the performance on SugarCrepe for each fine-grained task. In Table 4, we compare the same methods in other types of benchmarks. These are: ImageNet (Deng et al., 2009), Stanford Cars (Krause et al., 2013), CIFAR10 (Krizhevsky, 2009), CIFAR100 (Krizhevsky, 2009), MNIST (LeCun et al., 1994), EuroSAT (Helber et al., 2019, 2018), Oxford Flowers 102 (Nilsback and Zisserman, 2008), and Describable Textures (DTD) (Cimpoi et al., 2014). Following Radford et al. (2021), we employ the top-1 accuracy metric for them, except for Oxford Flowers 102, where we use the mean per class.

Our method presents a high compositionality recognition performance overall while having comparable performance to the pre-trained model in the rest of the benchmarks. Existing methods achieve high numbers on compositionality at the cost of a significant drop in other tasks.

5 Conclusions

In this paper, we introduced CLoVe – a method to considerably improve the compositionality of CLIP-like pre-trained models while preserving their performance on other tasks. The method consists of fine-tuning contrastive VLMs with hard negative texts by leveraging synthetically captioned images, as they can provide a great trade-off between quality and quantity. Subsequently, our method patches the original model with the fine-tuned one to convey the best of two worlds – compositional skills while maintaining the performance on other tasks.

We showed experimentally that CLoVe improves the performance of such models on multiple tasks, both compositionality-related and non-compositionality-related. We ablated the different components of our method and showed their importance: the data quality, the use of hard negatives in training, and the model patching.

Our code and pre-trained models are publicly available at `http://anonymous.edu`. Our code

	ARO				SugarCrepe			SVO-Probes			
	Attr.	Rel.	C-Ord.	F-Ord.	Repl.	Swap	Add.	Subj.	Verbs	Obj.	Avg.
pre-trained	63.5	59.8	47.7	59.9	80.1	62.3	72.8	84.0	79.3	87.8	69.7
NegCLIP	70.5	**80.1**	87.0	90.1	85.1	75.3	85.9	90.9	84.7	92.3	84.2
REPLACE	**71.2**	72.9	80.1	86.7	88.2	74.8	89.5	**92.0**	84.6	93.0	83.3
Ours w/o patching	69.0	77.4	**91.7**	**93.6**	**88.6**	**76.1**	**90.5**	88.2	83.7	91.6	**85.0**
Ours ($\alpha = .6$)	69.7	72.7	86.6	92.1	87.0	74.6	85.8	90.5	**86.4**	**93.3**	83.9

Table 3: Zero-shot results on three compositional benchmarks. The best results are in **bold**. An underline indicates results within 1% of best.

	IN	Cars	CIFAR10	CIFAR100	MNIST	EuroSAT	Flowers	DTD	Avg.
pre-trained	**63.4**	**59.7**	89.8	64.2	**48.9**	50.5	**66.6**	**44.4**	**60.9**
NegCLIP	55.8	45.6	85.9	60.9	45.3	32.9	55.9	39.0	52.7
REPLACE	52.9	42.7	84.6	60.2	36.6	34.3	51.9	34.5	49.7
Our w/o patching	53.1	48.7	88.5	62.0	40.4	46.9	43.2	36.3	52.4
Ours ($\alpha = .6$)	62.8	56.8	**91.4**	**68.1**	48.7	**57.4**	61.1	41.2	**60.9**

Table 4: Zero-shot results on eight image classification tasks. The best results are in **bold**. An underline indicates results within 1% of best.

will allow for an easy replacement of CLIP-like weights with the ones we provide, considerably boosting the language composition performance.

Limitations

Our work is limited in the following ways.

Our method does not solve the compositionality problem completely. The performance of our method on the compositionality benchmarks still presents a gap regarding the human performance reported by the papers associated with each of the employed benchmarks.

Employing synthetic captions can introduce undesired noise. Image captioners may sometimes hallucinate, introducing incorrect concepts or inaccurate descriptions of such objects. This is especially true for quantities, such as when there are four horses in the scene, but the synthetic caption mentions three. Future work can focus on methods to improve the synthetic caption quality.

We did not study the effect of the performance of the patched models on different demographics. It could be the case that some demographics are misrepresented in some task performance (compositional or not) after the model has been patched. Users should be careful about this aspect.

In this work, we focus on two-tower models because of their efficiency for classification and retrieval. We leave the study of single-tower models for future work.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikoł aj Bińkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. 2022. Flamingo: a visual language model for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 35, pages 23716–23736. Curran Associates, Inc.

James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, Wesam Manassra, Prafulla Dhariwal, Casey Chu, Yunxin Jiao, and Aditya Ramesh. 2023. Improving image generation with better captions.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media, Inc.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, S. Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S. Chen, Kathleen A. Creel, Jared Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren E. Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas F. Icard, Saahil

Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, O. Khattab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir P. Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Benjamin Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, J. F. Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Robert Reich, Hongyu Ren, Frieda Rong, Yusuf H. Roohani, Camilo Ruiz, Jack Ryan, Christopher R'e, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishna Parasuram Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei A. Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. 2021. On the opportunities and risks of foundation models. *ArXiv*.

Minwoo Byeon, Beomhee Park, Haecheon Kim, Sungjun Lee, Woonhyuk Baek, and Saehoon Kim. 2022. COYO-700M: Image-text pair dataset. `https://github.com/kakaobrain/coyo-dataset`.

Santiago Castro and Fabian Caba. 2022. Fitclip: Refining large-scale pretrained image-text models for zero-shot video understanding tasks. In *33rd British Machine Vision Conference 2022, BMVC 2022, London, UK, November 21-24, 2022*. BMVA Press.

Santiago Castro, Oana Ignat, and Rada Mihalcea. 2023. Scalable performance analysis for vision-language models. In *Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (*SEM 2023)*, pages 284–294, Toronto, Canada. Association for Computational Linguistics.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C. Lawrence Zitnick. 2015. Microsoft COCO Captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*.

Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. 2023. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2818–2829.

M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. 2014. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Casper da Costa-Luis, Stephen Karl Larroque, Kyle Altendorf, Hadrien Mary, richardsheridan, Mikhail Korobov, Noam Raphael, Ivan Ivanov, Marcel Bargull, Nishant Rodrigues, Guangshuo Chen, Antony Lee, Charles Newey, CrazyPython, JC, Martin Zugnoni, Matthew D. Pagel, mjstevens777, Mikhail Dektyarev, Alex Rothberg, Alexander Plavin, Daniel Panteleit, Fabian Dill, FichteFoll, Gregor Sturm, HeoHeo, Hugo van Kemenade, Jack McCracken, MapleCCC, and Max Nordlund. 2023. tqdm: A fast, Extensible Progress Bar for Python and CLI.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Anuj Diwan, Layne Berry, Eunsol Choi, David Harwath, and Kyle Mahowald. 2022. Why is winoground hard? investigating failures in visuolinguistic compositionality. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2236–2250, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Christiane Fellbaum. 2010. *Theory and Applications of Ontology: Computer Applications*, chapter WordNet. Springer Netherlands, Dordrecht.

Alon Hafri, E. J. Green, and Chaz Firestone. 2023. Compositionality in visual perception. *Behavioral and Brain Sciences*, 46:e277.

Charles R Harris, K Jarrod Millman, Stéfan J Van Der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, et al. 2020. Array programming with NumPy. *Nature*, 585(7825):357–362.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. 2018. Introducing eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. In *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*, pages 204–207. IEEE.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. 2019. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*.

Lisa Anne Hendricks and Aida Nematzadeh. 2021. Probing image-language transformers for verb understanding. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3635–3644, Online. Association for Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Cheng-Yu Hsieh, Jieyu Zhang, Zixian Ma, Aniruddha Kembhavi, and Ranjay Krishna. 2023. SugarCrepe: Fixing hackable benchmarks for vision-language compositionality. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

John D Hunter. 2007. Matplotlib: A 2D graphics environment. *Computing in science & engineering*, 9(03):90–95.

Gabriel Ilharco, Mitchell Wortsman, Samir Yitzhak Gadre, Shuran Song, Hannaneh Hajishirzi, Simon Kornblith, Ali Farhadi, and Ludwig Schmidt. 2022. Patching open-vocabulary models by interpolating weights. In *Advances in Neural Information Processing Systems*, volume 35, pages 29262–29277. Curran Associates, Inc.

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. 2021. Openclip. If you use this software, please cite it as below.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 4904–4916. PMLR.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. 2017. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica Hamrick, Jason Grout, Sylvain Corlay, Paul Ivanov, Damián Avila, Safia Abdalla, Carol Willing, and Jupyter development team. 2016. Jupyter Notebooks – a publishing format for reproducible computational workflows. In *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, pages 87–90, Netherlands. IOS Press.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 2013. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*.

Alex Krizhevsky. 2009. Learning multiple layers of features from tiny images. Technical report, University of Toronto.

Hugo Laurençon, Lucile Saulnier, Leo Tronchon, Stas Bekman, Amanpreet Singh, Anton Lozhkov, Thomas Wang, Siddharth Karamcheti, Alexander M Rush, Douwe Kiela, Matthieu Cord, and Victor Sanh. 2023. OBELICS: An open web-scale filtered dataset of interleaved image-text documents. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Yann LeCun, Corinna Cortes, and Christopher J.C. Burges. 1994. The MNIST database of handwritten digits.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *Fortieth International Conference on Machine Learning*.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 12888–12900. PMLR.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common objects in context. In *Computer Vision – ECCV 2014*, pages 740–755, Cham. Springer International Publishing.

Ilya Loshchilov and Frank Hutter. 2017. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

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

TorchVision maintainers and contributors. 2016. TorchVision: PyTorch's computer vision library. https://github.com/pytorch/vision.

Antoine Miech, Jean-Baptiste Alayrac, Ivan Laptev, Josef Sivic, and Andrew Zisserman. 2020. RareAct: A video dataset of unusual interactions. *arXiv preprint arXiv:2008.01018*.

Maria-Elena Nilsback and Andrew Zisserman. 2008. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

OpenAI. 2023. GPT-4V(ision) System Card. Technical report, OpenAI.

Letitia Parcalabescu, Michele Cafagna, Lilitta Muradjan, Anette Frank, Iacer Calixto, and Albert Gatt. 2022. 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)*, pages 8253–8280, Dublin, Ireland. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Fernando Pérez and Brian E. Granger. 2007. IPython: a system for interactive scientific computing. *Computing in Science and Engineering*, 9(3):21–29.

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

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

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. Zero-shot text-to-image generation. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8821–8831. PMLR.

Arijit Ray, Filip Radenovic, Abhimanyu Dubey, Bryan A. Plummer, Ranjay Krishna, and Kate Saenko. 2023. Cola: A benchmark for compositional text-to-image retrieval. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

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

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. 2022a. LAION-5B: An open large-scale dataset for training next generation image-text models. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Christoph Schuhmann, Andreas Köpf, Theo Coombes, Richard Vencu, Benjamin Trom, and Romain Beaumont. 2022b. LAION COCO: 600M synthetic captions from LAION2B-EN.

Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. 2021. LAION-400M: Open dataset of CLIP-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*.

Ravi Shekhar, Sandro Pezzelle, Yauhen Klimovich, Aurélie Herbelot, Moin Nabi, Enver Sangineto, and Raffaella Bernardi. 2017. FOIL it! find one mismatch between image and language caption. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 255–265, Vancouver, Canada. Association for Computational Linguistics.

Robyn Speer. 2019. ftfy. Zenodo. Version 5.5.

Ole Tange. 2011. GNU Parallel - the command-line power tool. *;login: The USENIX Magazine*, 36(1):42–47.

The Pandas development team. 2023. pandas-dev/pandas: Pandas.

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

Michael Tschannen, Manoj Kumar, Andreas Peter Steiner, Xiaohua Zhai, Neil Houlsby, and Lucas Beyer. 2023. Image captioners are scalable vision learners too. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Subhashini Venugopalan, Lisa Anne Hendricks, Marcus Rohrbach, Raymond Mooney, Trevor Darrell, and Kate Saenko. 2017. Captioning images with diverse objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. 2020. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature methods*, 17(3):261–272.

Michael L. Waskom. 2021. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021.

Ross Wightman. 2019. PyTorch image models. https://github.com/rwightman/pytorch-image-models.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Omry Yadan. 2019. Hydra – a framework for elegantly configuring complex applications. Github.

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

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

A SugarCrepe Fine-Grained Performance

In Table 5, we show SugarCrepe's fine-grained task results.

	Replacement				Swap			Addition				
	Obj.	Att.	Rel.	Avg.	Obj.	Att.	Avg.	Obj.	Att.	Avg.	Task Avg.	Avg.
pre-trained	90.8	80.2	69.1	80.1	61.0	63.8	62.3	77.1	68.5	72.8	71.7	72.9
NegCLIP	92.6	85.9	76.8	85.1	**75.6**	75.1	<u>75.3</u>	88.8	83.0	85.9	82.1	82.5
REPLACE	<u>93.5</u>	<u>90.2</u>	<u>80.9</u>	<u>88.2</u>	74.0	75.5	74.8	**90.9**	88.0	<u>89.5</u>	<u>84.2</u>	<u>84.7</u>
Ours w/o patching	<u>93.0</u>	**91.0**	**81.6**	**88.6**	74.4	**77.9**	**76.1**	86.2	**94.7**	**90.5**	**85.1**	**85.5**
Ours ($\alpha = .6$)	**93.8**	89.1	78.2	87.0	74.4	74.8	74.6	84.4	87.3	85.8	82.5	83.1

Table 5: Results on SugarCrepe. The best results are in **bold**. An <u>underline</u> indicates results within 1% of best.