Teaching Structured Vision & Language Concepts to Vision & Language Models

Sivan Doveh[1,2], Assaf Arbelle[1], Sivan Harary[1], Eli Schwartz[1,3], Roei Herzig[1,3],
Raja Giryes[3], Rogerio Feris[4], Rameswar Panda[4], Shimon Ullman*[2], Leonid Karlinsky*[4]

[1]IBM Research, [2]Weizmann Institute of Science, [3]Tel-Aviv University, [4]MIT-IBM Watson AI Lab

Abstract

Vision and Language (VL) models have demonstrated remarkable zero-shot performance in a variety of tasks. However, some aspects of complex language understanding still remain a challenge. We introduce the collective notion of Structured Vision & Language Concepts (SVLC) which includes object attributes, relations, and states which are present in the text and visible in the image. Recent studies have shown that even the best VL models struggle with SVLC. A possible way of fixing this issue is by collecting dedicated datasets for teaching each SVLC type, yet this might be expensive and time-consuming. Instead, we propose a more elegant data-driven approach for enhancing VL models' understanding of SVLCs that makes more effective use of existing VL pre-training datasets and does not require any additional data. While automatic understanding of image structure still remains largely unsolved, language structure is much better modeled and understood, allowing for its effective utilization in teaching VL models. In this paper, we propose various techniques based on language structure understanding that can be used to manipulate the textual part of off-the-shelf paired VL datasets. VL models trained with the updated data exhibit a significant improvement of up to 15% in their SVLC understanding with only a mild degradation in their zero-shot capabilities both when training from scratch or fine-tuning a pre-trained model. Our code and pretrained models are available at: https://github.com/SivanDoveh/TSVLC

1. Introduction

Recent Vision & Language (VL) models [19, 31, 43, 44, 47, 57] achieve excellent zero-shot performance with respect to various computer-vision tasks such as detection, classification, segmentation, etc. However, recent studies [68, 82] have demonstrated that even the strongest VL models struggle with the compositional understanding of some basic Structured VL Concepts (SVLC) such as object attributes, inter-object relations, transitive actions, object states and more. Collecting specialized large scale data to *teach* VL models these missing 'skills' is impractical, as finding specialized text-image pairs for each kind and possible value of the different attributes, relations, or states, is both difficult and expensive.

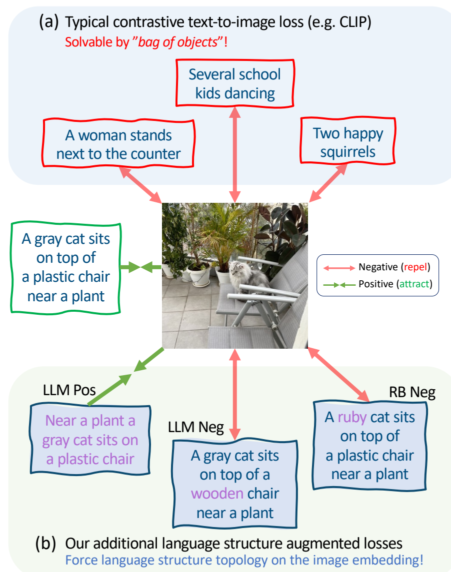

Figure 1. **Teaching language structure to VL models.** (a) Standard contrastive text-to-image loss (e.g. CLIP [57]) tends to underemphasize SVLC content of the text, likely due to the random nature of the training batches; (b) We generate modified versions of corresponding texts and use them to add losses to explicitly teach language structure (SVLC) to VL models.

Another important challenge in training VL models with new concepts is catastrophic forgetting, which is a common property to all neural models [7, 34, 37, 49, 58] and has been explored for VL models in a recent concurrent work [14]. Large VL models such as CLIP [57] and CyCLIP [19] have exhibited excellent zero-shot learning abilities in many tasks. Therefore, even given a large dataset with new concepts, it is important not to lose these abilities when performing the adaptation to the new data.

*Equal contribution

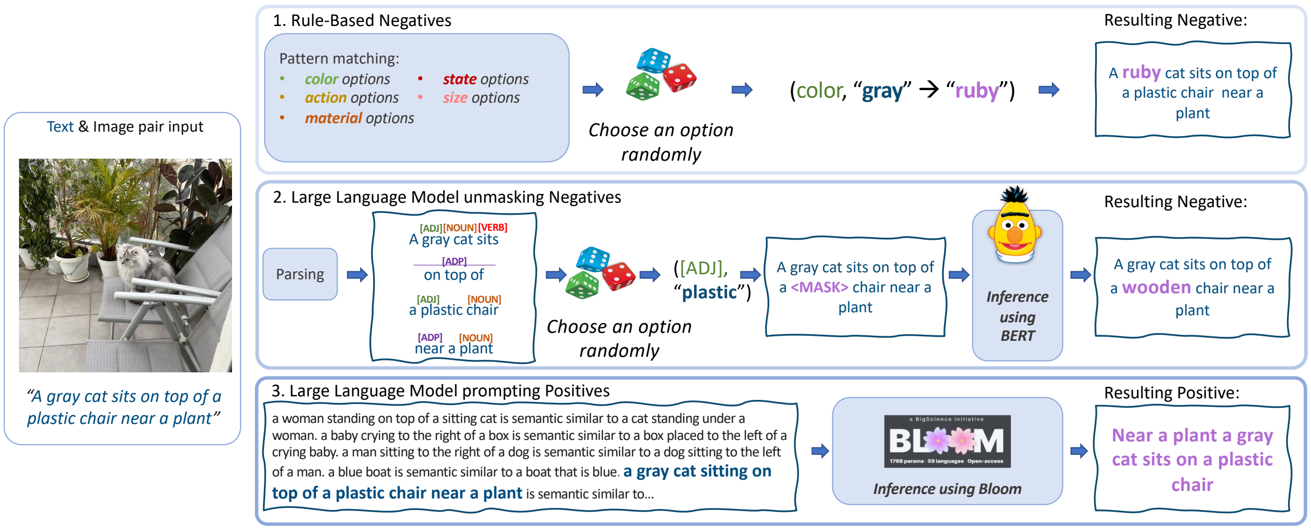

Figure 2. Teaching structured image understanding to VL models via structured textual data manipulation harnessing the power of language modeling. (1) Generating Rule-Based negative texts (Sec. 3.1.1); (2) Generating negatives using Large Language Model (LLM) unmasking (Sec. 3.1.2); (3) Generating analogies (positives) via LLM prompting (Sec. 3.1.3).

In this paper, we propose a way to leverage existing (off-the-shelf) VL pre-training data sources in order to improve the SVLC understanding skills of a given model, while at the same time maintaining its zero-shot object recognition accuracy. Naturally, succeeding in this goal would lead to potential improvement w.r.t. SVLC understanding in a wide variety of downstream tasks building upon pre-trained VL models, such as zeros-shot detection, segmentation, image generation, and many more.

Recent research [68, 82] has shown that VL models exhibit an 'object bias' partially due to the contrastive text-to-image loss used in their pre-training. For example, the popular CLIP-loss [57] is computed over a random batch of text-image pairs sampled from a large-scale and diverse VL dataset with the chance of two images in the same batch containing the same set of objects being very low. For such a loss, representing just a 'bag of objects' in each image or text is sufficient for matching the corresponding pairs. Intuitively, this leads to the 'object bias' where SVLCs like attributes, states, and relations are being underrepresented (e.g. having a much smaller amplitude in the resulting feature superposition), consequently causing the aforementioned issues with SVLC understanding.

Based on this intuition, we propose a simple data-driven technique that harnesses existing *language* parsing and modeling capabilities to enhance the importance of SVLCs in the VL model training losses. For each text in the training batch, we automatically generate alternative negative or positive text by manipulating its content to be opposite or equivalent to the original text. Using the newly generated texts, we explicitly teach SVLC to the model via additional losses (see Fig. 1) that enforce differentiating between different (original and generated) SVLC texts and are no longer satisfiable by the 'bag of objects' representation.

Towards this end, we propose several techniques for implementing this approach, including (i) rule-based priors based on classical NLP parsing and word substitution vocabulary according to attribute/relation type; (ii) prompting a Large Language Model (LLM) (e.g. [53]) for analogous text; (iii) generating different meaning (negative) alternatives by LLM-based unmasking of parsed text entities of different kinds; (iv) combinations of these methods.

We demonstrate that all these techniques can lead to significant improvements of up to 15% percent when measuring the VL models' SVLC understanding. We verify this on 5 datasets: VG [39], HAKE [48], VAW [54], SWIG [55], and all combined, using the protocol recently proposed in VL-Checklist [82]. In addition, we show that the resulting VL models largely preserve their zero-shot object recognition performance. For the latter, we also propose a variant of efficient LLM fine-tuning using low-rank residual adapters (LoRA) [27] adjusted to VL models. Finally, we show that our framework allows better harnessing of the standard available VL data, e.g. CC3M [62] and LAION [61]. This is exhibited by the aforementioned gains, both in new VL models trained from scratch, as well as in models fine-tuned using strong available VL models such as CLIP [57] and CyCLIP [19].

To summarize, we offer the following contributions: (i) We propose a data-driven approach for better harnessing the standard available VL data to improve VL models' SVLC understanding skills, such as understanding object attributes, inter-object relations, transitive actions, object states, and more, without sacrificing zero-shot object recog-

Dataset	Model	Pre	Arch	VL-Checklist Object	Attribute	Relation	21 Zero-Shot Tasks Average	
CC3M	CLIP [57]	✓	Vit-B/32	81.58%	67.60%	63.05%	56.37%	(a)
	CLIP + LoRA	✓	Vit-B/32	80.93% (-0.66%)	66.28% (-1.32%)	55.52% (-7.53%)	56.41% (+0.04%)	
	CLIP + Ours RB Neg	✓	Vit-B/32	83.89% (+2.30%)	73.35% (+5.75%)	75.33% (+12.28%)	54.32% (-2.05%)	
	CLIP + Ours LLM Neg	✓	Vit-B/32	84.44% (+2.85%)	71.63% (+4.03%)	74.82% (+11.77%)	55.60% (-0.77%)	
	CLIP + Ours RB+LLM Negs	✓	Vit-B/32	85.09% (+3.50%)	73.90% (+6.30%)	78.72% (+15.67%)	54.66% (-1.71%)	
	CLIP + Ours Combined	✓	Vit-B/32	85.00% (+3.42%)	71.97% (+4.37%)	68.95% (+5.90%)	54.77% (-1.60%)	
	CLIP [57]	✓	Vit-B/16	82.91%	67.32%	61.80%	60.00%	(b)
	CLIP + Ours RB+LLM Negs	✓	Vit-B/16	85.82% (+2.91%)	73.92% (+6.6%)	77.40% (+15.6%)	59.37% (-0.63%)	
	CLIP + Ours Combined	✓	Vit-B/16	84.75% (+1.84%)	71.18% (+3.86%)	69.68% (+7.88%)	59.87% (-0.13%)	
	CLIP	✗	Vit-B/32	71.17%	57.86%	45.20%	21.96%	(c)
	CLIP + Ours Combined	✗	Vit-B/32	71.79% (+0.62%)	63.29% (+5.43%)	58.13% (+12.93%)	20.96% (-1.00%)	
	CLIP [57]	✗	Vit-B/16	64.01%	54.27%	41.57%	15.49%	(d)
	CLIP + Ours RB+LLM Negs	✗	Vit-B/16	73.11% (+9.1%)	65.32% (+11.05)	71.93% (+30.36)	20.78% (+5.29%)	
	CLIP + Ours Combined	✗	Vit-B/16	72.99% (+8.98%)	63.01% (+8.74%)	62.95% (+21.38)	20.61% (+5.12%)	
	CyCLIP [19]	✓	R50	73.49%	59.33%	53.83%	26.00%	(e)
	CyCLIP + LoRA	✓	R50	73.30% (-0.19%)	58.89% (-0.44%)	53.03% (-0.80%)	26.30% (+0.30%)	
	CyCLIP + Ours Combined	✓	R50	74.20% (+0.71%)	63.52% (+4.20%)	59.47% (+5.63%)	26.31% (+0.31%)	
	CyCLIP	✗	R50	69.41%	57.59%	53.70%	21.02%	(f)
	CyCLIP + Ours Combined	✗	R50	71.50% (+2.09%)	65.69% (+8.10%)	70.20% (+16.50%)	20.44% (-0.42%)	
LAION	CLIP [57]	✓	Vit-B/32	81.58%	67.6%	63.05%	56.37%	(g)
	CLIP + LoRA	✓	Vit-B/32	82.18% (+0.60%)	68.48% (+0.88%)	62.72% (-0.33%)	57.15% (+0.78%)	
	CLIP + Ours Combined	✓	Vit-B/32	82.54% (+0.96%)	69.64% (+2.04%)	66.05% (+3.00%)	56.71% (+0.34%)	

Table 1. **VL-Checklist and Zero-Shot classification evaluation (ImageNet + 20 datasets)**. Finetuned models (for 5 epochs as detailed in Sec. 3.3, starting from officially released CLIP [57] and CyCLIP [19] weights) are marked with ✓in the Pre-trained (*Pre*) column, while models trained from scratch (for 10 epochs) are marked with ✗. The gains and losses of our approach (**+Ours**) are in color and are computed w.r.t. to corresponding baselines in each section. CLIP/CyCLIP + LoRA indicate finetuning in the same way and on the same data, but without our approach. Finetuning without LoRA on the same data yields significantly worse performance in all metrics. Sections are separated by **bold** horizontal lines. **(a)** *CC3M fine-tuning - CLIP - Vit-B/32*: we significantly improve the SVLC understanding, observing only small ZS performance drops, 0.77%; **(b)** *CC3M fine-tuning - CLIP - Vit-B/16*: we can observe similar improvements as in (a), with an even smaller impact on ZS performance; **(c)** *CC3M from scratch - CLIP - Vit-B/32*: we significantly improve SVLC understanding with only a small (1%) decrease in ZS performance; **(d)** *CC3M from scratch - CLIP - Vit-B/16*: compared to (c), even greater SVLC understanding improvement is observed (up to 30.36%), at no cost in ZS performance, and even improvement of over 5%; **(e)** *CC3M fine-tuning - CyCLIP*: we use CyCLIP original code (with LoRA integration) and losses, as can be seen - adding our techniques improves CyCLIP SVLC performance considerably without sacrificing ZS performance; **(f)** *CC3M from scratch - CyCLIP*: observing even largergains in SVLC understanding compared to (e) (up to 16.5%), with small reduction in ZS performance of 0.42%; **(g)** *LAION fine-tuning - CLIP - Vit-B/32*: we improve SVLC understanding without any decrease in ZS performance.

nition performance; (ii) More specifically, we propose to leverage the well-understood and well-modeled structure of language, through classical NLP parsing and/or use of the modern pre-trained LLMs, for manipulating the text part of the standard VL paired datasets to regularize VL training and teach SVLC understanding to VL models. (iii) We further propose an adaptation of efficient LLM fine-tuning technique of [27] for fine-tuning VL models, allowing for only minimal reduction in zero-shot object recognition performance after fine-tuning, while still obtaining the aforementioned SVLC understanding gains. (iv) Empirically, for the popular CLIP [57] and its most recent extension CyCLIP [19], we demonstrate SVLC understanding average improvements of up to 13% when training from scratch, and 15% when fine-tuning from a pretrained model.

2. Related Work

Vision-language (VL) Models. (*e.g.*, CLIP [57] and ALIGN [31]) show significant advances in diverse zero-shot downstream tasks. They are pre-trained using contrastive image-text alignment on a large-scale noisy dataset of text-image pairs collected from the web. Several methods [9, 46, 65] additionally employ off-the-shelf object detectors to extract region features. In order to relax this limitation, some methods [31,36,43,76] propose to use cross-attention layers with self-supervised learning objectives including image-text matching and masked/autoregressive language modeling. BLIP [43] generates synthetic captions from the language modeling head and filters noisy captions based on the image-text matching score. Recently, there have been attempts to learn finer-level alignment and relations between image and text [15, 17, 19, 47, 77]. FILIP proposes fine-grained contrastive learning to maximize the token-wise similarity between visual and textual tokens. CyClip [19] imposes additional geometrical consistency on the image and text embeddings. DeCLIP [47] introduces additional positives from the nearest neighbors. While these methods improve image-text retrieval tasks on the existing benchmarks, such as ImageNet [60] and MS-COCO [50], recent studies such as VL-CheckList [83] and the Winoground Challenge [68], show that these models cannot distinguish fine-grained language details or understand structured concepts (SVLCs) such as object attributes and relations. In this paper, we focus on the latter and propose orthogonal data-driven techniques that have the potential to improve the SVLC understanding for all VL models.

Learning Structured Representations. A full understanding of the semantics of rich visual scenes requires the ability to understand visual concepts, such as detecting individual entities and reasoning about their interactions and attributes. Structured representations have played an important role in achieving this goal, having been successfully applied to a wide range of computer vision applications: vision and language [10, 45, 46, 66], scene graphs [25, 29, 38, 56, 75], relational reasoning [4, 5], human-object interactions [16, 33, 74], action recognition [1, 2, 23, 24, 30, 52, 70], and even image & video generation from graphs [3, 22, 32]. However, most of these works rely on detailed, manually curated, supervision, often involving annotation of location information and structural details, which are very expensive to collect and scale, resulting in limited-size or synthetic data sources for training. In contrast, in our work, we focus on methods for teaching SVLC understanding to large VL models while only leveraging the available large-scale noisy VL data sources collected from the web without any use of expensive manual curation.

Data Augmentation. Augmentation plays a key role in many computer vision applications [8, 63]. Several advanced image augmentation methods (CutMix [78], mixup [79], AutoAugment [11], RandAugment [12], etc) have been proposed and greatly improved computer vision task performance. Text augmentation has been tackled trough back-translation [73], word and frame-semantic embedding augmentations [69], word replacement [81], random word insertion/deletion/swap [71], or using a text generative model [41] in diverse NLP applications. In VL tasks, previous work explores machine translation between different languages [6, 35], generating synthetic captions [43], adversarial/synthetic data augmentation for VQA [59, 67] or mixup for VL [20]. We focus on leveraging the well-understood and modeled language structure for manipulating text in a way that explicitly targets teaching SVLC semantics to VL models. To the best of our knowledge, this has not been attempted before.

3. Method

In this section, we discuss the proposed framework for improving SVLC performance of VL models using already available VL data. Sec. 3.1 and Sec. 3.2 present our main techniques for teaching SVLC to VL models. These approaches can be effectively applied both for fine-tuning existing strong VL Pre-trained models, as well as for training VL models from scratch. In both cases, they exhibit significantly improved SVLC performance as demonstrated in our experiments in Sec. 4. Sec. 3.3 presents our strategy for fine-tuning VL models on SVLC-enhanced VL data, while at the same time being parameter efficient and significantly reducing forgetting, thus, maintaining the VL model Zero-Shot (ZS) performance. Qualitative examples showing improvements attained by our proposed approach, as well as some examples of failure cases, are provided in the supplementary material.

3.1. Teaching SVLC Using NLP and LLMs

In this section, we present several ways in which the data of existing VL pre-training paired datasets can be enhanced to emphasize SVLC in the texts, and teaching them to the VL model. We propose two kinds of data enhancements - generating negative and positive text alternatives. When generating negative examples, only one word of the sentence is changed such that the semantic meaning of the sentence changes. We propose two methods for the generation of the negatives: (i) rule-based (Sec. 3.1.1); and (ii) LLM-based (Sec. 3.1.2). Positive alternatives are generated as sentences with semantically similar meanings, but different wording (Sec. 3.1.3). We then present the losses which properly take these two types of generated textual data into account during training in Sec. 3.2.

3.1.1 Generating Rule-Based (RB) Negatives

One simple yet effective method for negative text generation is using a collection of pre-defined language rules which match and replace words of a specific entity type or a pattern, such as color, material, size, etc. This method is especially useful when one has prior knowledge of a specific aspect of the language that needs to be taught. For example, if we know that our model lacks the ability to understand the colors of objects, we can easily create a rule for detecting and replacing color words in the text, for generating negative text that does not correspond to its paired image. To employ the generation of the rule-based negative, for each taught SVLC we define a list of words belonging to its characteristic. We then scan the VL data texts searching for the words within these lists. If a word is located, we simply replace it with a randomly selected word from the same list to generate a negative pair. For example, applying the color-rule to a sentence: "A big **brown** dog" can lead to "A big **yellow** dog". If a text has multiple candidates of words to be replaced, one of them is chosen randomly. We perform this process multiple times for the full list of SVLC characteristics of interest such as color, size, material, spatial relations, etc. These generated negative texts are SVLC specific and differ from the original text in only one word. For a detailed description and more examples of the RB negative generation, please refer to the supplementary material.

3.1.2 LLM-based Negative Generation via Unmasking

A natural extension of rule-based negatives technique is the generation of negatives using Large Language Models (LLMs) unmasking. Recent LLMs are explicitly trained in a self-supervised manner with the objective of "unmasking" parts of the text. Given a sentence with one missing word, models such as BERT [13], can suggest multiple words that fit the context of the sentence. Using this useful property of LLMs, we can therefore automatically create plausible negative examples without the need for prior knowledge of the SVLC characteristics of interest. In order to focus the randomly selected masked words to be likely to belong to SVLCs of interest, we use common NLP parsing techniques (such as spacy [26]) to parse the sentence into its components such as nouns, verbs, adjectives, adverbs, etc. We then randomly choose a type of sentence part and a word belonging to this part type, mask out the selected word and replace it with one of the options suggested by the LLM's unmasking. These negative examples, when used properly in the loss function (Eq. (3)) focus the network on the important details that affect the SVLC understanding. As we show in Sec. 4, this method is extremely useful and can significantly improve the VL model's understanding of different SVLCs. Further details and examples of the LLM negative generation are provided in the supplementary material.

3.1.3 Generating Text Analogies via LLM Prompting

While the goal of the negative text generation (Sec. 3.1.1 and Sec. 3.1.2) was to make minor perturbations to a given text such that the meaning changes, the goal here is exactly the opposite. We would like to make major changes to the text, while still keeping the same semantic meaning. For example "A woman standing left to a sitting cat" and "A cat sitting to the right of a standing woman" are two very different texts describing the exact same scene. One effective way to generate such semantically similar texts is by prompting the foundational LLMs. Specifically, we use the open access BLOOM [53] model. In the spirit of recently popular in-context learning [18], we present the model with a textual prompt with examples of semantically similar texts (see Fig. 2). We then append the current image caption and retrieve the BLOOMs prediction of a semantically similar text. For a detailed description, we refer the reader to the supplementary material.

3.2. Losses

All of our evaluated models (CLIP [57] and CyCLIP [19]) admit a text & image pair (T, I) and are comprised of two parts: (i) image encoder $e_I = \mathcal{E}_I(I)$; (ii) text encoder $e_T = \mathcal{E}_T(T)$. In this notation, the text-to-image similarity score is therefore computed as:

$$\mathcal{S}(T, I) = \exp\left(\frac{\tau e_T^T e_I}{||e_T||^2||e_I||^2}\right), \tag{1}$$

where τ is a learned temperature parameter.

Contrastive Loss. As most contemporary VL models, we employ the contrastive CLIP-loss [57] as one of our losses for each batch $\mathcal{B}$.

$$\mathcal{L}_{cont} = \sum_i log\left(\frac{S(T_i, I_i)}{\sum_j S(T_i, I_j)}\right) + log\left(\frac{S(T_i, I_i)}{\sum_k S(T_k, I_i)}\right). \tag{2}$$

Negatives Loss. In our ablation study in Sec. 5, we show that for a given text T_i simply adding the corresponding generated negative text T_i^{neg} to the contrastive loss $\mathcal{L}_{cont}$ is much less effective than having a separate loss individually attending to the similarity difference of T_i and T_i^{neg} w.r.t. the image I_i corresponding to T_i. We, therefore, employ the following *negatives loss*:

$$\mathcal{L}_{neg} = \sum_i -log\left(\frac{\mathcal{S}(T_i, I_i)}{\mathcal{S}(T_i, I_i) + \mathcal{S}(T_i^{neg}, I_i)}\right). \tag{3}$$

Analogy Loss. For the generated analogy texts T_i^{sim} produced from the text T_i which corresponds to image I_i, we employ the combination of the two following losses:

$$\mathcal{L}_{sim}^{text} = \sum_i -log\left(\frac{\mathcal{S}(T_i^{sim}, T_i)}{\sum_j \mathcal{S}(T_i^{sim}, T_j)}\right), \tag{4}$$

where with some abuse of notation, $\mathcal{S}(T_1, T_2)$ denotes the exponent cosine similarity between text T_1 and text T_2 text-embeddings, and:

$$\mathcal{L}_{sim}^{img} = \sum_i -log\left(\frac{\mathcal{S}(T_i^{sim}, I_i)}{\sum_j \mathcal{S}(T_i^{sim}, I_j)}\right), \tag{5}$$

which simply corresponds to the second summand of $\mathcal{L}_{cont}$ in Eq. (2), with replacing T_i by T_i^{sim}.

Our final loss is, therefore:

$$\mathcal{L} = \mathcal{L}_{cont} + \alpha \cdot \mathcal{L}_{neg} + \beta \cdot (\mathcal{L}_{sim}^{text} + \mathcal{L}_{sim}^{img}). \quad (6)$$

3.3. Fine-tuning Pre-trained VL Models

Each of the $\mathcal{E}_T$ and $\mathcal{E}_I$ networks is comprised of a mix of non-parametric functions and two types of parametric functions: linear layers and embedding layers. Roughly, each of those functions, $\mathcal{F}_k^{lin}(x)$ and $\mathcal{F}_k^{emb}(x)$ where k is layer index, is parameterized by a weight matrix $\mathcal{W}_k$ so that:

$$\mathcal{F}_k^{lin}(x) = \mathcal{W}_k \cdot x \quad (7)$$

$$\mathcal{F}_k^{emb}(x) = EMB(x; \mathcal{W}_k) \quad (8)$$

where EMB is the embedding operator assuming x is a stream of integers and picking the respective columns of $\mathcal{W}_k$. Following the idea proposed in LoRA [27] for efficient LLM fine-tuning using low-rank residual adapters, when adapting a pre-trained VL model $\mathcal{M} = (\mathcal{E}_T, \mathcal{E}_I)$ we parameterize the adapted weights $\mathcal{W}_k^*$ of the model $\mathcal{M}^*$ fine-tuned from $\mathcal{M}$ as:

$$\mathcal{W}_k^* = \mathcal{W}_k + \mathcal{A}_k \cdot \mathcal{B}_k \quad (9)$$

where for $\mathcal{W}_k$ of size $m \times l$, $\mathcal{A}_k$ and $\mathcal{B}_k$ are rank-r matrices of sizes $m \times r$ and $r \times l$ respectively. These low-rank residual adapters can be applied efficiently as:

$$\mathcal{F}_k^{*,lin}(x) = \mathcal{F}_k^{lin}(x) + \mathcal{A}_k \cdot (\mathcal{B}_k \cdot x) \quad (10)$$

$$\mathcal{F}_k^{*,emb}(x) = \mathcal{F}_k^{emb}(x) + \mathcal{A}_k \cdot EMB(x; \mathcal{B}_k) \quad (11)$$

During the fine-tuning of $\mathcal{M}^*$, we freeze all the base model $\mathcal{M}$ parameters $\forall k, \{\mathcal{W}_k\}$ and only the LoRA adapters $\forall k, \{(\mathcal{A}_k, \mathcal{B}_k)\}$ are being learned. In the above notation we disregard possible bias terms of the linear functions, if they are present, since we keep them frozen too.

There are several interesting things to note about the proposed architecture: (i) as opposed to [64], who evaluated the use of efficient LLM fine-tuning techniques for VL models adaptation, we add our LoRA adapters everywhere, i.e to all layers of both the text and image encoders, and not only to the text encoder/decoder as done in [64]; (ii) as opposed to [80] who attach a small side network only to the *output* of the adapted model, our LoRA adapters are added to all the parametric functions inside the model and affect all the intermediate computations; (iii) same as noted in the original [27] paper, at inference all the LoRA adapters can be folded back into the original weight matrices by simple summation, thus returning the number of total parameters to be the same as in the original model and hence have the same inference speed; (iv) with rank r kept low, the number of extra parameters added by all the LoRA adapters can be very low making adaptation fast and efficient; finally, (v) such form of fine low-rank adaptation allows for significantly mitigating the zero-shot performance forgetting effects as demonstrated in our results and explored in the corresponding ablation (Sec. 5).

4. Experiments

4.1. Datasets

We train the model using common Image-Text pair datasets, namely Conceptual Captions 3M [62] and LAION 400M [61] and test using the VL-Checklist [82] datasets, which will be described below.

Conceptual Captions 3M (CC3M) [62] is a dataset of three million image-text pairs automatically crawled from the internet where image descriptions are harvested from Alt-text attributes and then processed and filtered to create relatively clean descriptions.

LAION 400M [61] is a very large scale image-text pair dataset which, similarly to CC3M has been automatically harvested from the internet. One major difference between the two, apart from the size, is that LAION examples have been filtered using the pretrained CLIP model, such that the CLIP image-text similarity is high by design. Recent re-implementations of the original CLIP paper have succsefly used LAION 400M to reproduce similar capabilites as the original CLIP model.

VL-Checklist [82] combines images and annotations from the Visual Genome [39], SWiG [55], VAW [54], and HAKE [48] datasets. It is processed such that each image is annotated with two captions, one positive and one negative. The positive caption originates in its source dataset and corresponds to the image. The negative caption is constructed from the positive caption so that only one word, corresponding to the tested SVLC of interest, is changed to negate the SVLC (*e.g.*, color, size, material, etc.). VL-Checklist evaluates three main types of VL concepts further subdivided into 7 types of SVLCs total: (1) Object: its spatial location and size, (2) Attribute: color, material, size, state, and action, and (3) Relation: spatial or action relation between two objects. In the following sections we report results on a combined VL-Checklist dataset. The results on individual comprising datasets (Visual Genome [39], SWiG [55], VAW [54], and HAKE [48]) are provided in the supplementary material.

Zero-Shot Classification We evaluated our method on 21 classification dataset using the Zero-Shot classification protocol described in the ELEVATER Image Classification Toolkit [42]. The evaluation inclueds 21 different datasets, including common classification datasets such as ImageNet [60], CIFAR100 [40], EuroSat [21], and others. in Tables 1-3 we report the average results over the 21 tasks.

	VL-Checklist Object	Attribute	Relation	21 Zero-Shot Tasks Average
CLIP [57]	81.58%	67.60%	63.05%	56.37%
CLIP +LoRA	80.93% (-0.66%)	66.28% (-1.32%)	55.52% (-7.53%)	56.41%(+0.04%)
w/o Neg Loss	82.27% (+0.69%)	67.58% (-0.02%)	55.17% (-7.88%)	55.37% (-1.00%)
Ours RB+LLM Negs	85.09% (+3.50%)	73.90% (+6.30%)	78.72% (+15.67%)	54.66% (-1.71%))

Table 2. **Ablation study - separate Negative Losses (Sec. 3.2) vs adding negatives to contrastive loss.**

4.2. Implementation Details

For CLIP we use the ML-Foundation Open-CLIP repository [28] and for CyCLIP [19] we use its original code repository, which is also based on Open-CLIP. In most experiments, unless stated otherwise, we train our model for five epochs on 4 V100 NVIDIA GPUs, with a total batch size of 512. When starting from a pre-trained model, we use rank 4 LoRA adapters (Sec. 3.3) and the learning rate is set to $5E-6$. When training from scratch, for CLIP we use the default parameters set in the open-CLIP library, and for CyCLIP the defaults of its original implementation. For all CLIP experiments, we use VIT/32-B as the model architecture and ResNet-50 when training CyCLIP (following [19]). When fine-tuning we initialize with the original model weights released by the respective authors. In all experiments involving a combination of CyCLIP [19] and our method (in Sec. 4.4-4.5), in addition to Eq. (6) loss we also employ all the extra losses proposed in CyCLIP [19].

4.3. Baselines

We compare our method to two strong baselines under several configurations. The first is the CLIP [57] OpenAI pretrained model trained on 400M image-text pairs which achieves high ZS performance. The second is the very recent CyCLIP [19] method which, similarly to us, improves over the original CLIP loss. For a fair comparison all methods use the same network architecture and the same initialization from the OpenAI pretrained model. For the CyCLIP baseline we continue training from the pretrained initialization using LoRA and the CyCLIP losses. As CyCLIP losses are orthogonal to ours, we also show a unified version of the two methods (CyCLIP + Ours Combined).

4.4. Fine-tuning VL Models

In the following experiments we show the effects of fintuning a pre-trained VL model using our additional data enhancement methods and losses. All experiments are initialized from the official OpenAI CLIP. We compare our method to two baselines on the VL-Checklist tasks, and on the Zero-Shot image classification task. The first baseline is the original pretrained model without any further training. The second, is the same model with the additional LoRA parameters, trained on CC3M (Table 1a) and LAION (Table 1g) using the original CLIP loss function. Table 1a

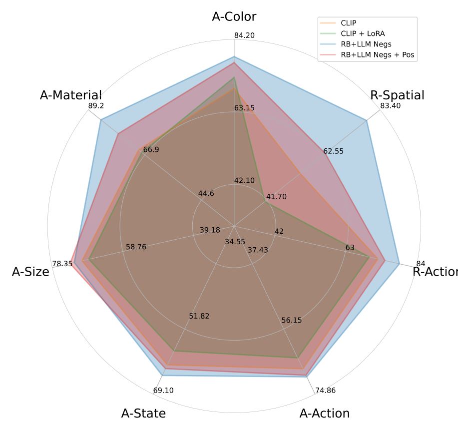

Figure 3. **CC3M fine-tuning - detailed.** Detailed results of the baselines (CLIP, CLIP+LoRA) and our models (RB+LLM Neg, RB+LLM Negs+Pos) initialized from OpenAI pretrained CLIP ViT-32B and fine-tuned (except CLIP) on CC3M.

shows several configurations of our method compared to the baselines. We see that our method shows significant improvements on all VL-Checklist tasks reaching up to 15% improvement. Figure 3 displays the relative gains on all tasks of the "Attribute" and "Relation" tests. It is clear that our gains are across all tests. These improvements come at a price of some minor degradation to the Zero-Shot performance compared to CLIP. Moreover, when fine-tuning CLIP on the LAION dataset (Table 1g) we do not see these degradations. In Tab. 1b we show consistent gains to ones in Tab. 1a when finetuning with a stronger (higher ZS) Vit-B/16 CLIP [57] pre-trained image encoder, observing an even lower drop in ZS performance. In Tab. 1e we show significant gains in fine-tuning CyCLIP on CC3M, comparing fine-tuning before and after integrating our proposed approach, this also comes at no ZS performance cost.

4.5. Training from scratch

In these experiments, we show the advantage of integrating our approach throughout the whole training procedure. To this end, we train from scratch on CC3M - both CLIP and CyCLIP, either on their own, or together with our proposed method integrated. Tables - Tab. 1c, Tab. 1d, and Tab. 1f show that, similar to finetining, our method significantly improves both CLIP and CyCLIP SVLC capabili-

Method	Attribute					Relation		21 Zero-Shot Tasks
	Color	Material	Size	State	Action	Action	Spatial	Average
Ours Pos	72.35%	69.25%	69.80%	59.35%	66.08%	70.97%	39.20%	**55.37%**
Ours RB Neg	78.45%	83.20%	69.50%	**65.95%**	69.66%	75.97%	74.70%	54.66%
Ours LLM Neg	76.00%	79.70%	72.75%	61.35%	68.36%	77.23%	72.40%	55.05%
Ours RB+LLM Negs	**79.25%**	**84.25%**	72.15%	64.05%	**69.82%**	**79.03%**	**78.40%**	54.66%%
Ours Combined	77.45%	77.35%	**73.35%**	62.30%	69.39%	74.70%	63.20%	54.77%%

Table 3. **Ablation study - component analysis.** Detailed evaluation on *Attribute* & *Relation* SVLCs.

ties, while still keeping similar ZS performance. Figure 4 shows the detailed parsing of the "Atribute" and "Relation" tests with consistent gains across the specific tasks. These findings suggest that had we trained our method for many epochs on a large dataset, such as the full LAION-9B, we would reach similar ZS performance while greatly enhancing the SVLC performance of the model.

5. Ablations

In this section we will examine several aspects and components of our proposed methods.

5.1. Negative Losses

Section 3.2 details our additional loss functions which utilize our generated texts. Specifically, Eq. (3) describes the loss which contrasts a given example **only** with its generated negatives. Table 2 clearly shows the importance of this loss as opposed to simply adding the negative examples to the original contrastive loss (Eq. (2)). Without explicitly forcing the network to attend to the small changes in the text (Eq. (3)), the generated negative examples, when inserted only to the contrastive loss (Eq. (2)) do not provide the desired gains over the baseline.

5.2. Component analysis

In Section 3 we describe several components of our proposed method. Specifically we present our RB negative generation (Sec. 3.1.1), our LLM-based negative generation (Sec. 3.1.2), and our LLM-based analogy generation, refered to here as "Pos" (Sec. 3.1.3). Tab. 3 provides a detailed analysis and comparison of the contribution of each component to the final result. Through this analysis we see two contradicting forces between the SVLC capabilities and the original Zero-Shot performance. We can see that each negative generation method plays a crucial role in some of the tasks while the use of both is usually the best performing option. On the other hand, the LLM-based analogy generation stabilizes the Zero-Shot performance and mitigates the drop with respect to the baseline. The joint version of all components ("Ours Combined") exhibits a good trade-off between the two contradicting forces.

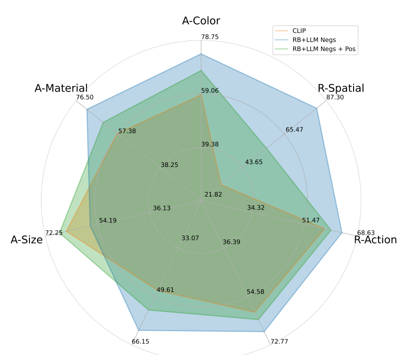

Figure 4. **CC3M train from scratch - detailed.** Detailed results of the baseline CLIP and our models (RB+LLM Neg, RB+LLM Negs+Pos), all CLIP ViT-32B, trained from scratch on CC3M.

6. Conclusions

We have presented a data-driven technique for enhancing the performance of VL models in the important task of SVLC understanding without sacrificing their impressive ZS object recognition capabilities. Our proposed method attains significant gains in multiple experiments on a variety of base VL models and datasets. It builds upon the modeling strength and knowledge of language structure to teach this structure to VL models in an orthogonal way, suggesting wide applicability to existing or future VL models.

While attaining impressive gains in SVLC understanding, we believe the small drop observed in ZS performance could be further reduced. An additional possible extension of our work is using more sophisticated sequence generation techniques for improving our batch data. One may combine annotation efforts with a language model to get improved data for training and evaluation [51]. Another possibility is adding a corrector [72] in the training that validates whether the VL model learns the correct concepts or not. We leave all of these exciting directions to future work.

Acknowledgements

We would like to thank Donghyun Kim for his invaluable help in this work. This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. FA8750-19-C-1001. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Defense Advanced Research Projects Agency (DARPA). This material is based upon work supported by the 'Data Science grant from the Israeli Council of Higher Education'. This research or RG was supported by ERC-StG SPADE grant no. 75749.

References

[1] Anurag Arnab, Chen Sun, and Cordelia Schmid. Unified graph structured models for video understanding. In *ICCV*, 2021. 4

[2] Elad Ben Avraham, Roei Herzig, Karttikeya Mangalam, Amir Bar, Anna Rohrbach, Leonid Karlinsky, Trevor Darrell, and Amir Globerson. Bringing image scene structure to video via frame-clip consistency of object tokens. In *Thirty-Sixth Conference on Neural Information Processing Systems*, 2022. 4

[3] Amir Bar, Roei Herzig, Xiaolong Wang, Anna Rohrbach, Gal Chechik, Trevor Darrell, and A. Globerson. Compositional video synthesis with action graphs. In *ICML*, 2021. 4

[4] Fabien Baradel, Natalia Neverova, Christian Wolf, Julien Mille, and Greg Mori. Object level visual reasoning in videos. In *ECCV*, pages 105–121, 2018. 4

[5] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018. 4

[6] Andrea Burns, Donghyun Kim, Derry Wijaya, Kate Saenko, and Bryan A Plummer. Learning to scale multilingual representations for vision-language tasks. In *European Conference on Computer Vision*, pages 197–213. Springer, 2020. 4

[7] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 233–248, 2018. 1

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 4

[9] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *European conference on computer vision*, pages 104–120. Springer, 2020. 3

[10] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *ECCV*, 2020. 4

[11] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018. 4

[12] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020. 4

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 5

[14] Yuxuan Ding, Lingqiao Liu, Chunna Tian, Jingyuan Yang, and Haoxuan Ding. Don't stop learning: Towards continual learning for the clip model. *arXiv:2207.09248*, 2022. 1

[15] Andreas Fürst, Elisabeth Rumetshofer, Viet Tran, Hubert Ramsauer, Fei Tang, Johannes Lehner, David Kreil, Michael Kopp, Günter Klambauer, Angela Bitto-Nemling, et al. Cloob: Modern hopfield networks with infoloob outperform clip. *arXiv preprint arXiv:2110.11316*, 2021. 4

[16] Chen Gao, Jiarui Xu, Yuliang Zou, and Jia-Bin Huang. Drg: Dual relation graph for human-object interaction detection. *ArXiv*, abs/2008.11714, 2020. 4

[17] Yuting Gao, Jinfeng Liu, Zihan Xu, Jun Zhang, Ke Li, and Chunhua Shen. Pyramidclip: Hierarchical feature alignment for vision-language model pretraining. *arXiv preprint arXiv:2204.14095*, 2022. 4

[18] Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes, 2022. 5

[19] Shashank Goel, Hritik Bansal, Sumit Bhatia, Ryan A Rossi, Vishwa Vinay, and Aditya Grover. Cyclip: Cyclic contrastive language-image pretraining. *arXiv preprint arXiv:2205.14459*, 2022. 1, 2, 3, 4, 5, 7

[20] Xiaoshuai Hao, Yi Zhu, Srikar Appalaraju, Aston Zhang, Wanqian Zhang, Bo Li, and Mu Li. Mixgen: A new multimodal data augmentation. *arXiv preprint arXiv:2206.08358*, 2022. 4

[21] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. 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*, 12(7):2217–2226, 2019. 6

[22] Roei Herzig, Amir Bar, Huijuan Xu, Gal Chechik, Trevor Darrell, and Amir Globerson. Learning canonical representations for scene graph to image generation. In *European Conference on Computer Vision*, 2020. 4

[23] Roei Herzig, Elad Ben-Avraham, Karttikeya Mangalam, Amir Bar, Gal Chechik, Anna Rohrbach, Trevor Darrell, and Amir Globerson. Object-region video transformers. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 4

[24] Roei Herzig, Elad Levi, Huijuan Xu, Hang Gao, Eli Brosh, Xiaolong Wang, Amir Globerson, and Trevor Darrell. Spatio-temporal action graph networks. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019. 4

[25] Roei Herzig, Moshiko Raboh, Gal Chechik, Jonathan Berant, and Amir Globerson. Mapping images to scene graphs with permutation-invariant structured prediction. In *Advances in Neural Information Processing Systems (NIPS)*, 2018. 4

[26] Matthew Honnibal and Ines Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 2017. 5

[27] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen.

Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 2, 3, 6

[28] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, July 2021. If you use this software, please cite it as below. 7

[29] Achiya Jerbi, Roei Herzig, Jonathan Berant, Gal Chechik, and Amir Globerson. Learning object detection from captions via textual scene attributes. *ArXiv*, abs/2009.14558, 2020. 4

[30] Jingwei Ji, Ranjay Krishna, Li Fei-Fei, and Juan Carlos Niebles. Action genome: Actions as composition of spatio-temporal scene graphs. *arXiv preprint arXiv:1912.06992*, 2019. 4

[31] 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. 1, 3, 4

[32] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1219–1228, 2018. 4

[33] Keizo Kato, Yin Li, and Abhinav Gupta. Compositional learning for human object interaction. In *ECCV*, 2018. 4

[34] Ronald Kemker, Marc McClure, Angelina Abitino, Tyler Hayes, and Christopher Kanan. Measuring catastrophic forgetting in neural networks. *AAAI Conference on Artificial Intelligence*, 2018. 1

[35] Donghyun Kim, Kuniaki Saito, Kate Saenko, Stan Sclaroff, and Bryan Plummer. Mule: Multimodal universal language embedding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11254–11261, 2020. 4

[36] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *International Conference on Machine Learning*, pages 5583–5594. PMLR, 2021. 4

[37] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 2017. 1

[38] Ranjay Krishna, Ines Chami, Michael S. Bernstein, and Li Fei-Fei. Referring relationships. *ECCV*, 2018. 4

[39] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017. 2, 6

[40] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6

[41] Varun Kumar, Ashutosh Choudhary, and Eunah Cho. Data augmentation using pre-trained transformer models. *arXiv preprint arXiv:2003.02245*, 2020. 4

[42] Chunyuan Li, Haotian Liu, Liunian Harold Li, Pengchuan Zhang, Jyoti Aneja, Jianwei Yang, Ping Jin, Yong Jae Lee, Houdong Hu, Zicheng Liu, and Jianfeng Gao. Elevater: A benchmark and toolkit for evaluating language-augmented visual models. *Neural Information Processing Systems*, 2022. 6

[43] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *arXiv preprint arXiv:2201.12086*, 2022. 1, 4

[44] 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

[45] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *ArXiv*, abs/1908.03557, 2019. 4

[46] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*, pages 121–137. Springer, 2020. 3, 4

[47] Yangguang Li, Feng Liang, Lichen Zhao, Yufeng Cui, Wanli Ouyang, Jing Shao, Fengwei Yu, and Junjie Yan. Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. *arXiv preprint arXiv:2110.05208*, 2021. 1, 4

[48] Yong-Lu Li, Liang Xu, Xinpeng Liu, Xijie Huang, Yue Xu, Mingyang Chen, Ze Ma, Shiyi Wang, Hao-Shu Fang, and Cewu Lu. Hake: Human activity knowledge engine. *arXiv preprint arXiv:1904.06539*, 2019. 2, 6

[49] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017. 1

[50] 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 *European conference on computer vision*, pages 740–755. Springer, 2014. 4

[51] Alisa Liu, Swabha Swayamdipta, Noah A. Smith, and Yejin Choi. Wanli: Worker and ai collaboration for natural language inference dataset creation, Jan. 2022. 8

[52] Joanna Materzynska, Tete Xiao, Roei Herzig, Huijuan Xu, Xiaolong Wang, and Trevor Darrell. Something-else: Compositional action recognition with spatial-temporal interaction networks. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 4

[53] Margaret Mitchell, Giada Pistilli, Yacine Jernite, Ezinwanne Ozoani, Marissa Gerchick, Nazneen Rajani, Sasha Luccioni, Irene Solaiman, Maraim Masoud, Somaieh Nikpoor, Muñoz Ferrandis Carlos, Stas Bekman, Christopher Akiki, Danish Contractor, David Lansky, Angelina McMillan-Major, Tristan Thrush, Suzana Ilić, Gérard Dupont, Shayne Longpre,

Manan Dey, Stella Biderman, Douwe Kiela, Emi Baylor, Teven Le Scao, Aaron Gokaslan, Julien Launay, and Niklas Muennighoff. Bigscience, bigscience language open-science open-access multilingual (bloom) language model. *International*, May 2021-May 2022. 2, 5

[54] Khoi Pham, Kushal Kafle, Zhe Lin, Zhihong Ding, Scott Cohen, Quan Tran, and Abhinav Shrivastava. Learning to predict visual attributes in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13018–13028, 2021. 2, 6

[55] Sarah Pratt, Mark Yatskar, Luca Weihs, Ali Farhadi, and Aniruddha Kembhavi. Grounded situation recognition. In *European Conference on Computer Vision*, pages 314–332. Springer, 2020. 2, 6

[56] Moshiko Raboh, Roei Herzig, Gal Chechik, Jonathan Berant, and Amir Globerson. Differentiable scene graphs. In *WACV*, 2020. 4

[57] 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, 2, 3, 5, 7

[58] Vinay Venkatesh Ramasesh, Ethan Dyer, and Maithra Raghu. Anatomy of catastrophic forgetting: Hidden representations and task semantics. In *International Conference on Learning Representations*, 2021. 1

[59] Arijit Ray, Karan Sikka, Ajay Divakaran, Stefan Lee, and Giedrius Burachas. Sunny and dark outside?! improving answer consistency in vqa through entailed question generation. *arXiv preprint arXiv:1909.04696*, 2019. 4

[60] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 4, 6

[61] 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. 2, 6

[62] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, 2018. 2, 6

[63] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020. 4

[64] Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5227–5237, 2022. 6

[65] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*, 2019. 3

[66] Hao Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *EMNLP*, 2019. 4

[67] Ruixue Tang, Chao Ma, Wei Emma Zhang, Qi Wu, and Xiaokang Yang. Semantic equivalent adversarial data augmentation for visual question answering. In *European Conference on Computer Vision*, pages 437–453. Springer, 2020. 4

[68] 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. 1, 2, 4

[69] William Yang Wang and Diyi Yang. That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using# petpeeve tweets. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 2557–2563, 2015. 4

[70] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *ECCV*, 2018. 4

[71] Jason Wei and Kai Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*, 2019. 4

[72] Sean Welleck, Ximing Lu, Peter West, Faeze Brahman, Tianxiao Shen, Daniel Khashabi, and Yejin Choi. Generating sequences by learning to self-correct, 2022. 8

[73] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems*, 33:6256–6268, 2020. 4

[74] Bingjie Xu, Yongkang Wong, Junnan Li, Qi Zhao, and M. Kankanhalli. Learning to detect human-object interactions with knowledge. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2019–2028, 2019. 4

[75] Danfei Xu, Yuke Zhu, Christopher B. Choy, and Li Fei-Fei. Scene Graph Generation by Iterative Message Passing. In *CVPR*, pages 3097–3106, 2017. 4

[76] Jinyu Yang, Jiali Duan, Son Tran, Yi Xu, Sampath Chanda, Liqun Chen, Belinda Zeng, Trishul Chilimbi, and Junzhou Huang. Vision-language pre-training with triple contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15671–15680, 2022. 4

[77] 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. 4

[78] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019. 4

[79] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 4

[80] Jeffrey O Zhang, Alexander Sax, Amir Zamir, Leonidas Guibas, and Jitendra Malik. Side-tuning: a baseline for network adaptation via additive side networks. In *European Conference on Computer Vision*, pages 698–714. Springer, 2020. 6

[81] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015. 4

[82] 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. 1, 2, 6

[83] 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. 4