– Independent Work Report Fall 2023 –

CompGPT: Probing the Compositional Understanding of Visual Language Models

Evan Wang
Adviser: Felix Heide

Abstract

This paper focuses on the limitations of Vision Language Models (VLMs) in understanding compositional relations, i.e. how these models comprehend and capture how different elements in a picture relate to each other. Compositional comprehension is fundamental to understanding the vision-language space, and thus it is imperative to explore this direction. In extending evaluations on new models, investigating current methods to evaluate and improve compositional understanding, and refining these methods, this paper adds to the growing knowledge about how VLMs operate and improve on comprehension tasks. Overall, this paper provides new data and directions on the fundamental question of compositional awareness in VLMs.

1. Introduction

Pioneering the rapid development of multi-modal models are Vision Language Models (VLMs). VLMs combine vision and language modalities to tackle various tasks, such as image captioning and generation.

Despite their advancements, VLMs still struggle with compositional reasoning. For instance, while humans easily understand spatial relationships in images, such as a cat being to the left of a dog, VLMs often confuse these relationships and the attributes of objects. It is no surprise that research has focused on why this weakness exists, and how to make VLMs better at this type of reasoning. However, there are still many untouched areas of research, such as specific models evaluated on specific tasks. Furthermore, the datasets and techniques used to improve compositonal understanding have many areas for improvement.

Throughout this paper, we demystify some of these areas. First, we evaluate a promising and efficient VLM on a standard compositional dataset. Beyond this, we investigate the validity of current datasets for evaluating compositional reasoning. We then improve on their question/task generation procedure to be more effective in measuring compositional reasoning, and aim to be more aligned with our human compositional reasoning. We introduce this refined caption-generation process as CompGPT.

This research addresses a broader question: In learning specific tasks, are VLMS simply memorizing tokens, or are they gaining a deeper understanding of the text-image space?? How can we guide VLMs towards a deeper comprehension? Investigating these questions not only aids in directing VLMs in grasping essential concepts but also offers valuable insights into the functioning of these models and the broader vision-language domain.

2. Problem Background and Related Work

2.1. Overview

VLMs struggle with understanding compositional concepts, i.e. linking objects to their attributes, understanding order sensitivity, and comprehending spatial relations [2]. Notably, one study prompted a dozen contemporary VLMs to identify the difference between "some plants surrounding a lightbulb" and "a lightbulb surrounding some plants"; every model failed to correctly match the images and descriptions, except one heavily-trained variant that succeeded by a 0.0001 confidence margin [10]. This phenomenon highlights the compositional weakness of many VLMs.

2.2. Vision Language Model: CLIP

The first VLM this paper focuses on is OpenAI's Contrastive Language–Image Pre-training (CLIP) model, released in 2021 [8]. CLIP's training objective is contrastive learning: to maximize the similarity between images and captions that match and minimize similarity between pairs that do not match. Thus, CLIP is a discriminative model that orients towards learning the boundaries and distinguishing features of image and text. Figure 1 diagrams this: starting with N pictures, each

with their own relevant caption, the model aims to maximize the dot product between the image embedding and the text embedding that match (along the diagonal), and minimize all other pairs. Note that both the image and text encoders utilize a transformer-based self-attention mechanism.

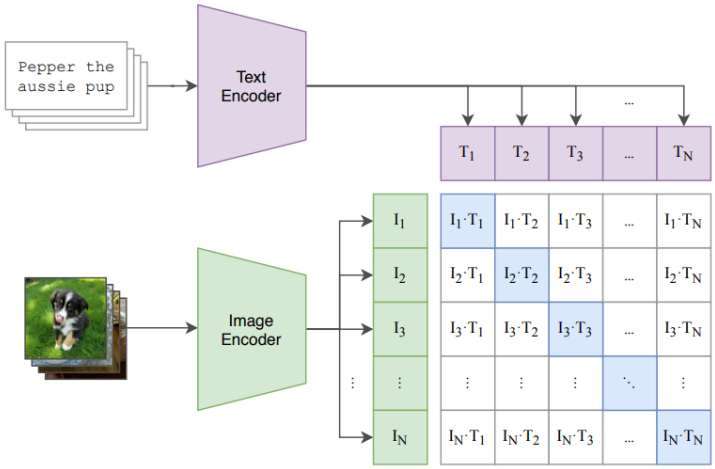

Figure 1: CLIP training visualization [8]

CLIP has demonstrated robust image-text detection capabilities, even on data it has not previously seen. Remarkably, zero-shot CLIP (i.e. no fine-tuning on the task) is able to match the performance of ResNet, a convolutional neural network with 50 layers trained on 1.28 million ImageNet examples [8]. This suggests that CLIP has a strong understanding of images and text. So, CLIP is strong on conventional tasks like visual object detection and classification, making it a compelling benchmark VLM to investigate.

2.3. Vision Language Model: Vision-and-Language Transformer (ViLT)

The second VLM this paper probes is the Vision-and-Language Transformer (ViLT), released in 2021, that utilizes the same transformer encoder for both image and text inputs. ViLT thus provides a simple and efficient architecture that eliminates the reliance on heavier, more complex convolutional neural networks as well as separate encoders for visuals and text. Figure 2 depicts the architecture of ViLT: the images are compartmentalized into patches, projected into the same dimension as text embeddings, and fed into the transformer like they are word tokens. Furthermore, unlike CLIP's

singular contrastive learning objective, ViLT utilizes multiple training objectives, including image text matching masked language modeling, and word patch alignment.

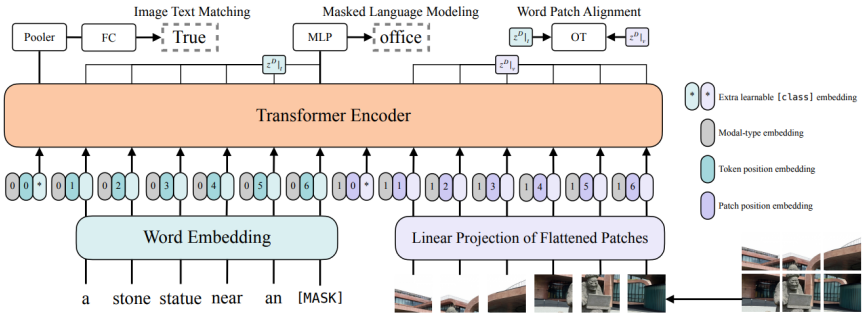

Figure 2: ViLT training, visualized [6]

Specifically, the input image $\in \mathbb{R}^3$ is flattened into a sequence of patches $\in \mathbb{R}^2$, which then goes through a trainable linear projection and positional embedding. The image and text embeddings are then concatenated together at the input level. The embeddings' hidden dimension is 768, layer depth is 12, patch size is 32, MLP depth is 3072, number of attention heads is 12.

ViLT has demonstrated super efficiency improvements compared to previous VLMs. Kim et. al. [6] show that that ViLT has between 3 to 17 times less floating point operations (FLOPS) than competitors, while achieving similar or even better performance on Visual Question Answering and Visual Reasoning tasks.

2.4. Evaluation Dataset: ARO

The evaluation dataset this paper delves into is the Attribution, Relation, and Order (ARO) dataset, created in 2022 [12]. The ARO dataset consists of images and captions; each image has a true caption that correctly describes the image, and a false caption that incorrectly reflects a compositional relationship in the image. For instance, a picture with a cat to the left of a dog will have a correct caption describing this scene, and a caption with the objects switched. The idea is that to pick the true caption, the VLM needs to correctly understand the spatial relation between the cat and dog.

ARO analyzes two main categories of compositionality: attribution and relation. Attribution concerns adjectives, e.g. "black cat and brown dog" vs. "black dog and brown cat". Relation

concerns actions/spatial orientation, e.g. "the cat is to the left of the dog" vs. "the dog is to the left of the cat." Crucially, ARO generated false captions from true captions via a strict hard rule: switch the placements of the objects (e.g. swapping cat and dog). Both Attribution and Relation have around 25,000 cases.

Remarkably, the paper found that CLIP performs little better than chance on the ARO dataset, achieving an accuracy of 0.59 on Relation and 0.63 on Attribtion respectively. This reveals a lack of compositional understanding, especially since random guessing would achieve an accuracy of 0.5, as there is one true caption and one false caption to choose per image.

2.5. Composition-Aware Training

To improve CLIP's compositional reasoning, Yuksekgonul et. al. [12] propose a stage of fine-tuning the model on composition-aware examples, created from the open-source COCO dataset. Still following the original contrastive learning approach of CLIP, compositionally incorrect captions are created for each image using the same rule-based approach of swapping parts of speech. Then, these false captions are inserted in the same training batch as the true captions. Thus, the model chooses between true and false captions during training, and to minimize error, it must learn to pick the true caption from false captions. They name this new model NegCLIP, and show that its performance on relation tasks improves from 0.59 to 0.80.

3. Approach

For the first part of our investigation, we conduct evaluations that are currently missing in the literature. Yuksekgonul et. al. [12] have already evaluated CLIP and NegCLIP on the ARO dataset, but research has not probed the robust and efficient ViLT on these tasks, so we implement evaluation here.

Next, we investigate the ARO dataset. Research indicates that datasets like ARO are inherently biased: with automatic rule-based caption generation, the wrong choice is often less logical and fluent, regardless of the image [5]. We will expand on this research by testing how probable language models consider each caption to be: bias here indicates a flawed dataset, as a significant

skew means a model can pick the right caption without actually understanding the image. We then attempt to refine these datasets, and evaluate our attempts.

Ultimately, we re-evaluate the VLMs on the new datasets and conduct appropriate analyses.

Then, based on these results, we will experiment with generating our own evaluation datasets. Since state-of-the-art chatbots like GPT have shown high capability to understand relations and natural language, prompting such bots may yield less biased captions and better datasets.

4. Implementation

4.1. Examining the ARO Dataset

First, to get a better firsthand understanding of the dataset, we manually inspected images and captions from ARO. To do this, we wrote a script to load in the dataset, randomly select images and their corresponding true and false captions, and inspected around 100 total images for both Attribution and Relation.

Next, for a quantitative assessment, we examined the biases of the ARO dataset captions. From our manual inspection, we hypothesized that the true captions and false captions were skewed in their logicality, coherency, and grammatical fluency. To test this, we calculated the perplexities of true captions and false captions, as determined by the open-source GPT-2 model [3].

Perplexity is defined as:

$$PP(W) = P(w_1, w_2, \ldots, w_N)^{-\frac{1}{N}} = \exp\left(-\frac{1}{N}\sum_{i=1}^{N} \log P(w_i | w_1, \ldots, w_{i-1})\right) \tag{1}$$

where $PP(W)$ is the perplexity of the word sequence W, $P(w_1, w_2, \ldots, w_N)$ is the probability of the word sequence, and N is the number of words in the sequence, where the probability of the sequence is determined by our model GPT-2. Intuitively, the less probability the model assigns a sequence, the more uncertain, or perplexed, the model is to observe that sequence. This metric can thus serve as a proxy for investigating caption validity: less coherent and less fluent sequences will generally have higher perplexities. GPT-2's loss function is defined as the average negative log

likelihood of a sequence, so to obtain perplexity we take the exponential of the model's loss on the desired sequences.

4.2. CompGPT: Improving the ARO Dataset

The original ARO dataset creates false captions with a simple rule-based approach: swap the positions of the nouns in the sentence. However, our hypothesis is that their approach creates biased datasets, as they do not consider the fluency, coherency, and logicality of the generated false caption. Inspired by research that has highlighted state-of-the-art chatbots' natural language capabilities, [5] we utilize GPT to create false captions. Due to resource and time constraints, focus specifically on ARO's Relation task, which involves spatial relations and verbs. This step involved a lot of prompt engineering: testing various inputs, questions, and guidelines for GPT to modify captions in the most desired way. GPT outputs have high variance, and sometimes has different outputs for the same prompt, so we took some time here to experiment with our prompting to increase consistency and faithfulness to the task. Figure 15 in the appendix shows the detailed input prompt used.

The goal of this prompt is to 1) describe the general task of creating a modified caption, break down the mechanism for doing so into specific parts (i.e. finding and replacing the relationship), enforcing requirements for the caption (i.e. ensuring it is coherent, fluent, and grammatically correct). Moreover, research and studies on chatbot effectiveness have established that few-shot prompting improves performance on specific tasks and domains. To follow this core prompt-engineering principle, we input two example captions and modifications for the model to emulate.

We follow two approaches in prompting GPT.

1. Using the chatbot interface (ChatGPT-4), following previous work we had conducted. Here, we give the instructions to the chatbot, input a JSON file of the captions, and ask for an output JSON file of the modified captions. [11]

2. Calling the OpenAI API in a script. Here, we have more fine-grain control over the system. Tunable parameters include the temperature, number of output tokens, number of output choices, system command,etc. We use default values for most parameters, but chose 0.2 as the temperature.

We wanted a low temperature value to create consistent outputs, but noticed that with a temperature of 0, the model would output identical captions for similar but not identical cases, which we thought would be counterproductive to the motivation of having multiple close but not exactly similar captions for detailed learning. Note that for this API based approach, we re-prompt GPT for each caption.

4.3. Run-Time Analysis

With our proposed methods of generating and evaluating captions, it is imperative to understand the run-time complexity of each process. We do so by varying input sizes and measuring how long it takes for GPT to output the modified captions, and how long it takes for the perplexity-calculating script to output perplexities. Timing the GPT chatbot interface approach was measured via a stop-watch and prompting on GPT-4. GPT API and Perplexity timing were measured by running the scripts and outputting the time for completion. The perplexity script was run on the Della Computing Cluster, using one 4 GB CPU and one 40 GB GPU; we converted tensors to arrays and moved from CPU to CUDA whenever possible for efficiency. The details of compute for GPT approaches is trivial since meaningful computation is done on OpenAI's servers.

4.4. Evaluation on ARO

4.4.1. Evaluating CLIP We create a modified ARO task. On the 23,000 cases, we use the original images and true captions, but incorporate our novel false captions. We then evaluate the models of interest (CLIP, NegCLIP, and ViLT) on this modified task, using the codebase of the ARO paper [12].

4.4.2. Evaluating ViLT Because of ViLT's efficiency and relative accuracy on fundamental visual-language reasoning tasks, we are interested in evaluating it on ARO. Moreover, ViLT has multiple training objectives that may allow it to "gain" a deeper understanding of compositional relations. Particularly, the word patch alignment objective is visualized in Figure 3 below.

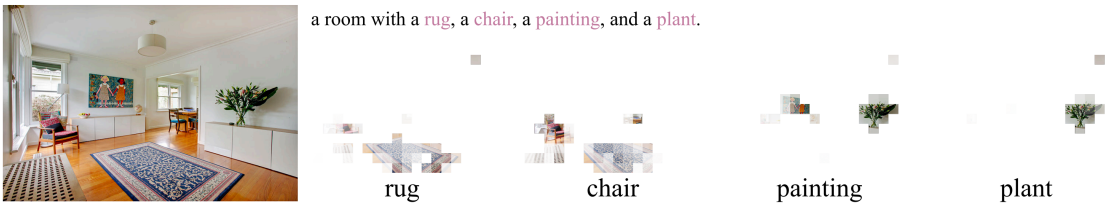

Figure 3: ViLT Word Patch Alignment Training Objective, visualized [6]

This objective encourages the model to associate words in the caption with specific regions of the image, for instance the word "rug" with the corresponding pixels. This may allow the model to better understand the spatial relations between entities and words, e.g. the plant being above the rug. Thus, we hypothesize the ViLT model will perform better on the relation task than CLIP.

We evaluated ViLT on ARO by working off of the original ARO papers' GitHub repository. However, because the architecture, input/output shapes, and overall usage of ViLT differs from the models the original paper looked at, implementation involved creating a new custom ViLTWrapper module to load in the image and caption data, call ViLT, and return calculated scores. We use the public ViltForImageAndTextRetrieval model hosted on HuggingFace [4], which includes a classifier linear layer on top of the final hidden state of the [CLS] token for image-to-text retrieval tasks.

Working off of the ARO repository's batch-based evaluation system, we code the algorithm shown below. The input is the evaluation data which consists of N data points, where each data point has K image options and L image options. In our case, $K = 1$ and $L = 2$, as we have a true caption and false caption for each image. Each image and caption pair passes through the model processor before we extract the ViLT output logits as the *batch_scores*. These scores are then compared with the ground truth values, where we take caption with the higher logit as the true caption that ViLT predicts.

Note that we evaluate ViLT on both the original ARO dataset and our modified version of ARO-both tests are novel contributions of this paper.

Algorithm 1 ViLT Evaluation

input: data
output: Tensor $\in \mathbb{R}^{N \times K \times L}$
for i **in** data **do**
  $image \leftarrow$ data[i]['image']
  $caption_0, \quad caption_1 \leftarrow$ data[i]['caption0'], data[i]['caption1']
  $input_0, \quad input_1 \leftarrow$ ViLTProcessor($image, caption_0$), ViLTProcessor($image, caption_1$)
  $score_0, \quad score_1 \leftarrow$ ViLT($input_0$).logits, ViLT($input_1$).logits
  $batch\_scores[i] \leftarrow (score_0, score_1)$
end for
return *batch_scores*

4.5. Composition-Aware Fine-Tuning

We follow Yuksekgonul et. al.'s [12] procedure for fine-tuning CLIP for compositional reasoning. We use their provided training and validation data, obtained by taking the COCO 2014 tasks and swapping various parts of speech to obtain composition-aware negative captions. We note that ideally, we would apply our proposed CompGPT procedure to create the composition-aware training data. However, due to the high scalabity costs of the procedure previously discussed, this was not in the scope of this paper.

We work off of the provided codebase for fine-tuning [12]. A visualization of the procedure is seen in Figure 4 below, where during fine-tuning, the false captions that test for compositional awareness, along with designated alternative images, are processsed in the same batch as the current test case. The model is thus pushed to maximize the embeddings' distance between the relevant image and compositionally incorrect captions. We fine-tune off the clip-ViT-B-32 model with batch size 256, 5 epochs, learning rate of 10^{-6}, 50 warmup steps.

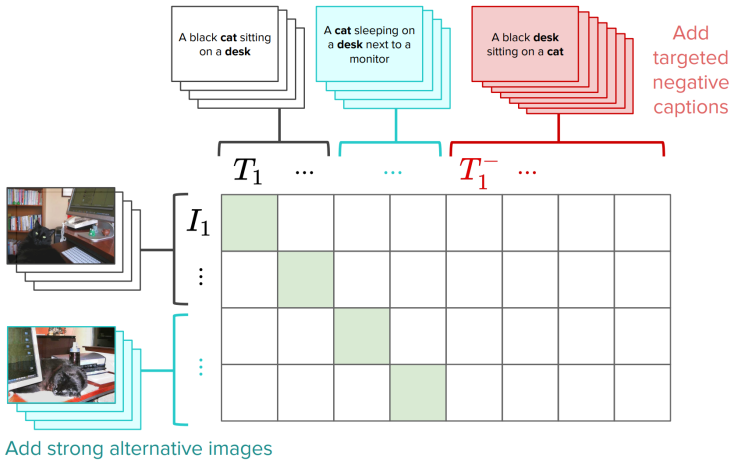

Figure 4: Visualization of Composition-Aware Fine-Tuning for CLIP [12]

5. Evaluation

5.1. Examining the ARO Dataset

Our tests show that the ARO dataset contains logically flawed captions. A notable example is shown below in Figure 14. The true caption is "the cat is on the sink"; false caption is "the sink is on the cat". We can tell the captions are biased: the false caption is illogical and implausible from our human understanding. Our quantitative approach corroborates this finding. GPT-2 assigns a perplexity of 232.0 to the true caption and 287.6 to the false caption: as expected, the perplexity of the nonsensical caption is much higher, and the CLIP model correctly picks the true caption during evaluation. But selecting the correct caption is trivial in this case: a simple language model can pass by only considering the text and picking the one with lower perplexity (the one it calculates to be more probable, i.e. closer to what the model expects to see).

The GPT-4 Chatbot approach produces the new caption "the cat is under the sink", which by inspection is much more plausible and has a perplexity of 195.7, which is actually lower than the true caption. When we substitute this new false caption in, CLIP fails the test. This case highlights the idea that the original ARO dataset has tasks that fail to genuinely probe compositional understanding, and instead may simply test the VLMs ability to calculate sequence probabilities.

Figure 5: Image with captions relating cat and sink. Rule-generated false caption is nonsensical and implausible for this image, creating a trivial evaluation task. [11]

More examples of trivial cases can be found in the Appendix.

Overall, for 59.4% of the true/false caption pairs in the ARO relation dataset, the perplexity of the false caption is greater than the perplexity of the true caption, indicating a prevalence of trivial cases like the cat and sink. This means that a completely "blind" model that only looks at the captions could achieve the same accuracy as CLIP. Furthermore, we examine the average perplexity gap, i.e. $PP(\text{false caption}) - PP(\text{true caption})$ averaged across all perplexity pairs, which is 39.7. This phenomenon points to a clearly biased dataset that doesn't actually measure compositional understanding as much as it measures the logical nature of captions.

To quantify this effect on the overall dataset, we calculate the perplexity as discussed in the methods, which is shown in the Original Captions row in Table 1. Remarkably, for 59.4% of the true/false caption pairs in the ARO relation dataset, the perplexity of the false caption is greater than the perplexity of the true caption. We denote these pairs as trivial pairs, since the boundary between true and false can be determined by a simple text-only discriminative model. Furthermore, we examine the average perplexity gap, i.e. $PP(\text{false caption}) - PP(\text{true caption})$, averaged across all perplexity pairs, which is 39.7.

The distribution of perplexity gaps can be observed in figure 6 below.

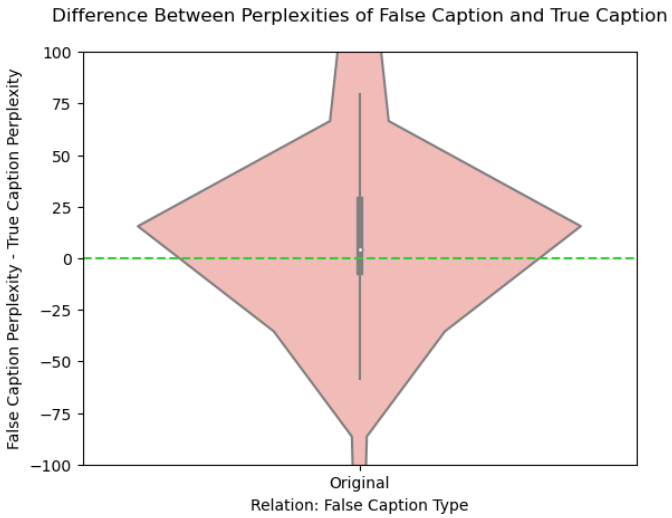

Figure 6: Violin plot and box plot showing the difference in perplexity gap distributions for original ARO captions

The distribution diagram visually confirms that there is a significant bias in ARO's original captions, where false captions have higher perplexities than true captions at an undesirable scale. We turn to GPT-4 to generate less biased false captions, as described in methods.

5.2. Improving the ARO Dataset

5.2.1. GPT-4 Chatbot Interface With our new modified captions from the GPT-4 chatbot interface, the percentage of caption pairs where $PP(\text{false caption}) > PP(\text{true caption})$, i.e. the percentage of trivial pairs, decreased from 59.4% to 48.3%. The average perplexity gap between false and true captions decreased from 39.7 to 13.5.

However, out of the 23937 total captions, 3773 of the generated captions did not change from the base true caption. Thus, the chatbot approach failed to create meaningful new captions about 15.8% of the time. For these unchanged caption pairs, we cannot draw a relevant comparison about perplexities, as the perplexities of equivalent captions is trivially the same. Accordingly, we remove the cases with unchanged captions from further analysis. After we remove these failed attempts, our new metrics are: 57.3% of pairs are trivial, average perplexity gap is 16.0. Both metrics do still both move in the desired direction, but the mitigation of trivial pairs is much less than thought, showing

that a contributing factor the decrease was due to GPT's failure to actually make a new caption, not necessarily improved logicality or coherency.

Assuming 50% trivial pairs is the ideal equilibrium, (half of the true captions have higher perplexity, half of the false captions have higher perplexity) this approach brings us 22.3% closer to our goal.

Moreover, Figure 7 shows the comparison of perplexity gap distributions. Our modified captions are grouped much closer to 0 as a whole, indicating a less biased dataset. Thus, we show that instead of strict rule-based caption generation approaches, instructional natural language leveraging state-of-the-art chat bots can successfully produce captions that are more logical and fluent.

5.2.2. GPT-3.5 API With the captions created from the GPT-3.5 API, the percentage of trivial pairs actually increased to 74.65%, and the average perplexity gap also increased to 89.05. This method was actually counterproductive in trying to refine the captions

Figure 7 shows a comparison via violin plot of the perplexity gaps. The dashed horizontal line indicates a gap of 0, which is the ideal perfect goal of equal perplexities for true and false captions. We see that the modified captions' perplexity gaps are distributed much closer to net zero, which reflects the improved dataset with less bias.

Caption Type	Trivial Pairs	Avg. Perp. Gap
Original Captions	59.40%	39.7
GPT-4 Chatbot Captions	57.30%	16.0
GPT-3.5 API Captions	74.65%	89.05

Table 1: Comparison of Metrics for Original and Modified Captions. Trivial pairs is the percentage of true/false caption pairings where the false caption perplexity is greater than true caption perplexity, indicating a trivial task of selecting the less probable caption. The average perplexity gap is the false caption perplexity - true caption perplexity, averaged across all pairs

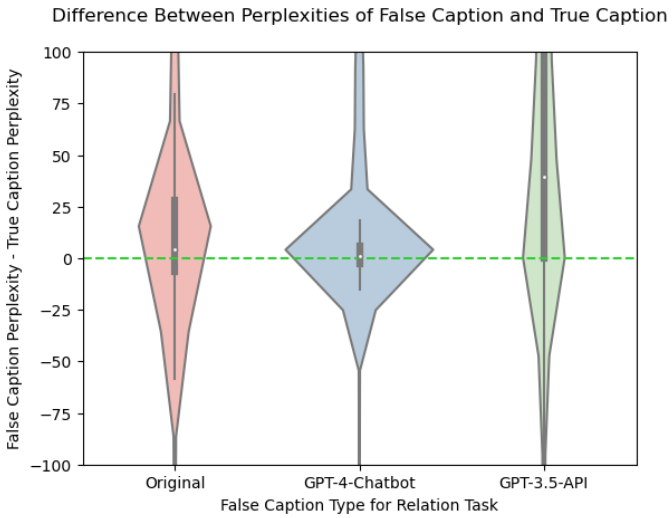

Figure 7: Violin plot and box plot showing the difference in perplexity gap distributions for original ARO, GPT4-Chatbot, and GPT-3.5 API approaches.

We think that the failure of GPT-3.5 API approach is due to that model versions' weakness in faithfully following the given instructions. The high cost for accessing the GPT-4 API (as outlined in the cost analysis section further on) made it difficult to conduct any meaningful quantitative analyses for that model version, but we note that from manual inspection, GPT-4 does seem to provide meaningfully higher quality modified captions than GPT-3.5. Take this illuminating example: for the original caption of "the boot is to the left of the man," GPT-4 outputs "the boot is on top of the man," while GPT-3.5 outputs "the boot is inside of the man." We think that GPT-4's generation is able to remain faithfully coherent and plausible with respect to the original caption, while GPT-3.5's generation does not seem very plausible.

Testing using the chatbot interfaces, another indicative example is for the original caption "the tent is to the left of the lady." ChatGPT 4 outputs "the tent is sheltering the lady", which is logical, plausible, and describes a scene different from the original. ChatGPT 3.5 outputs "the tent is walking beside the lady", which is implausible and does not successfully describe a different scene. Moreover, after prompting ChatGPT 3.5 to reconsider its answer, it correctly evaluates its own answer, but revises it to "the tent is surrounded by the lady", which is yet another implausible caption as it doesn't make sense for a singular human to surround a tent.

These results agree with the general public consensus that GPT-4 is significantly stronger than GPT-3.5 at critical thinking, analysis, and following specific instructions. However, due to feasibility of costs, we use GPT-3.5 as our main caption generation model.

5.3. Run-Time Analysis

Because our paper proposes novel general methods of generating captions and evaluating the validity of these captions, which would be utilized on large-scale datasets, it is crucial to understand the run-time complexities of the processes.

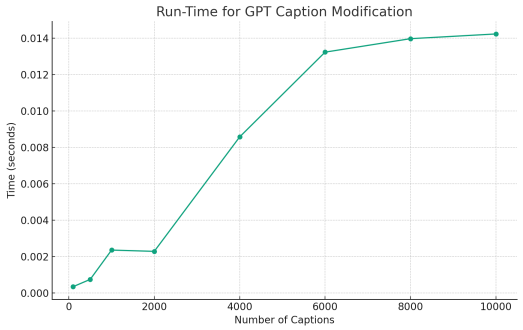

Figure 8: Run-Time for GPT Chatbot Interface

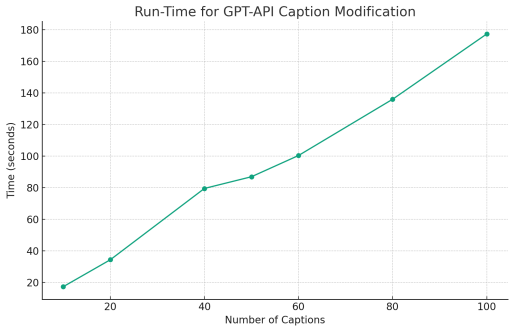

Figure 9: Run-time for GPT OpenAPI API

Figure 8 above shows the run-time complexity for using the GPT chatbot interface approach to modify captions, where the x-axis is the number of captions and the y-axis is the run-time in seconds. Visually, the shape does not neatly fall into linear, polynomial, or exponential time, as the increase in run-time seems to fall-off at larger inputs, suggesting a logarithmic or sub-linear relationship. However, based on our understanding of the task we are asking GPT to do, which is to modify each caption based on a set of rules, a linear relationship is more intuitive-logarthmic or sub-linear models doesn't match our intuition; perhaps with each test GPT was becoming more "used to" the task and was able to speed up. Regardless, we can provide a run-time upper bound of $O(n)$ with respect to the number of captions. The linear fit for this has $R^2 = 0.92$ and slope 1×10^{-6} seconds per caption.

Figure 9 shows the run-time complexity for the GPT OpenAI API approach. We see a linear

relationship, and can establish an upper bound of $O(n)$ here as well. The linear fit has a $R^2 = 0.993$ and a slope of 1.73 seconds per caption.

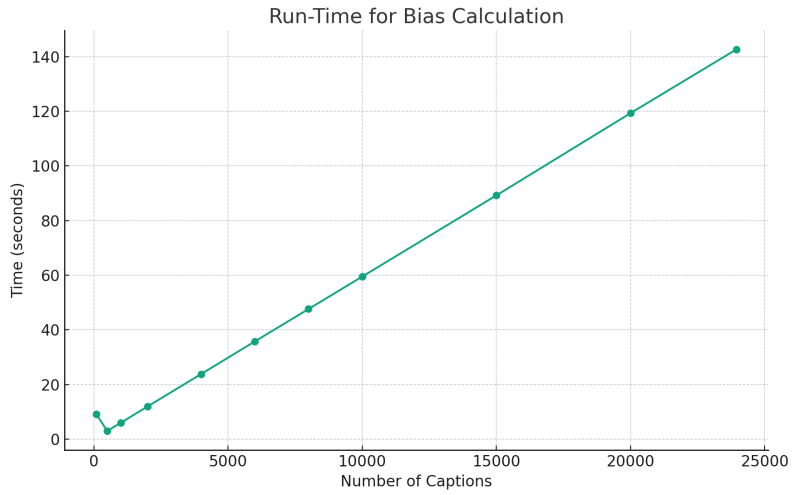

Figure 10: Run-Time for Bias Calculation

Figure 10 above shows the run-time complexity for using GPT-2 to calculate the perplexity of a list of captions, where the x-axis is the number of captions and the y-axis is the run-time in seconds. We see a clear linear relationship here. The linear fit for this has $R^2 = 0.997$ and 5×10^{-3} seconds per caption.

Thus, for both approaches of caption generation, and caption evaluation, we provide a run-time upper bound of $O(n)$, which allows for scalability. However, the raw time spent for the GPT API approach is costly. For instance, taking 1.73 seconds per caption would amount to around 38 hours of run-time on the popular COCO 2014 training dataset that consists of 80,000 image examples [7]. This is also only assuming one caption per image, and does not factor in the validation/test sets. Crucially, more compute cannot cut down this cost, since the bulk of the run-time is dependent on the network connection to OpenAI's servers and the response time of OpenAI themselves-that is where the computation is done.

5.3.1. Cost Analysis While the cost of the chatbot interface approach is constant ($20 per month) with respect to the number of captions, each call to the OpenAI API incurrs a fee, so considering this cost is crucial for the API approach. At the time of writing, the API pricing is $0.03 per 1K input tokens and $0.06 per 1K output tokens for GPT-4; $0.001 per 1K input tokens and $0.002 per 1K output tokens for GPT-3.5-turbo [1]. Continuing the cost estimate: our prompt has 149 words, and the average output has 24 words, so following the general estimate of 0.75 words per token [9], we arrive at 200 input tokens and 32 output tokens. So, one prompt using GPT-4 costs $0.008; one prompt using GPT-3.5 costs $0.0003. Running on the 23,000 captions of the ARO Relation dataset thus costs $184 and $7 respectively for GPT-4 and GPT-3.5.

5.4. Evaluations

Results of evaluating our main models of interest (CLIP, NegCLIP, and ViLT) are shown below in 2

Table 2: Accuracy of Vision-Language Models on Different Datasets

Dataset	CLIP	NegCLIP	ViLT
Original	59.0%	80.0%	56.2%
GPT-4 Chatbot	55.5%	64.6%	63.5%
GPT-3.5 API	62.9%	70.1%	43.2%

Visualization of this data is shown below in Figure 11.

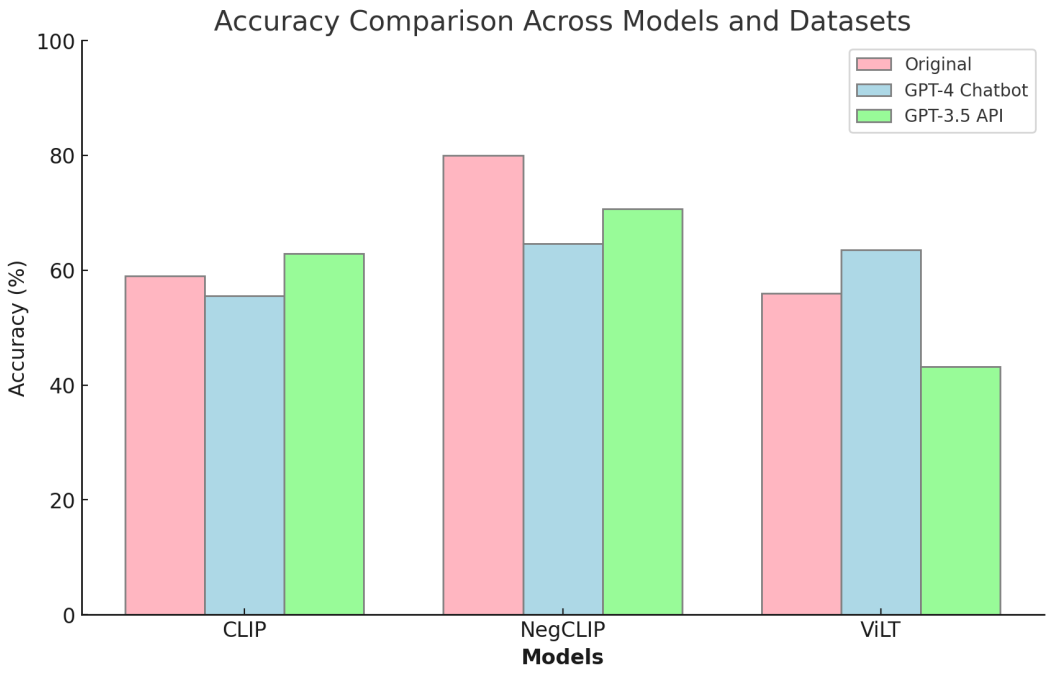

Figure 11: Accuracy for CLIP, NegCLIP, and ViLT, across the original, GPT 3.5 Chatbot, and GPT 4 API datasets

There a few key take-aways from these results.

First, for ViLT: based on the original ARO dataset, ViLT is not better than CLIP at compositional reasoning. This goes against our original hypothesis that ViLT's word patch alignment training objective would allow it to gain a deeper understanding of compositional relations when compared to CLIP's purely contrastive approach. However, we note that on the GPT-4 Chatbot dataset, which according to our perplexity metrics is the least biased dataset, ViLT outperforms CLIP by almost 8 percentage points, and is competitive with NegCLIP which had the advantage of specifically fine-tuning on the compositionally-swapped captions.

In terms of the datasets, we see that CLIP and NegCLIP perform better on GPT-3.5 API than on GPT-4. Since GPT-3.5 API is the extremely flawed dataset, this could be due the CLIP models taking advantage of the "blind" text plausibility approach. Interestingly, ViLT performs significantly worse on GPT-3.5 API than the other datasets, which could mean that ViLT does not try to take advantage of the "blind" approach, and instead the implausible captions are a hindrance to correct

predictions.

Furthermore, we do observe that NegCLIP is able to achieve the best performances overall, on all three datasets. This corroborates that composition-aware fine-tuning is a valid approach to improving compositional reasoning, and that fine-tuning on strict rule-based compositional swaps can still refine compositional understanding on non rule-based generated captions.

5.5. Composition-Aware Fine-Tuning

We fine-tune CLIP following the same procedure as Yuksekgonul et. al., where we train on the 2014 COCO dataset. Composition aware hard negatives are generated with strict rules by swapping parts of speech, as in the original ARO method. Results are shown below in Figure 12

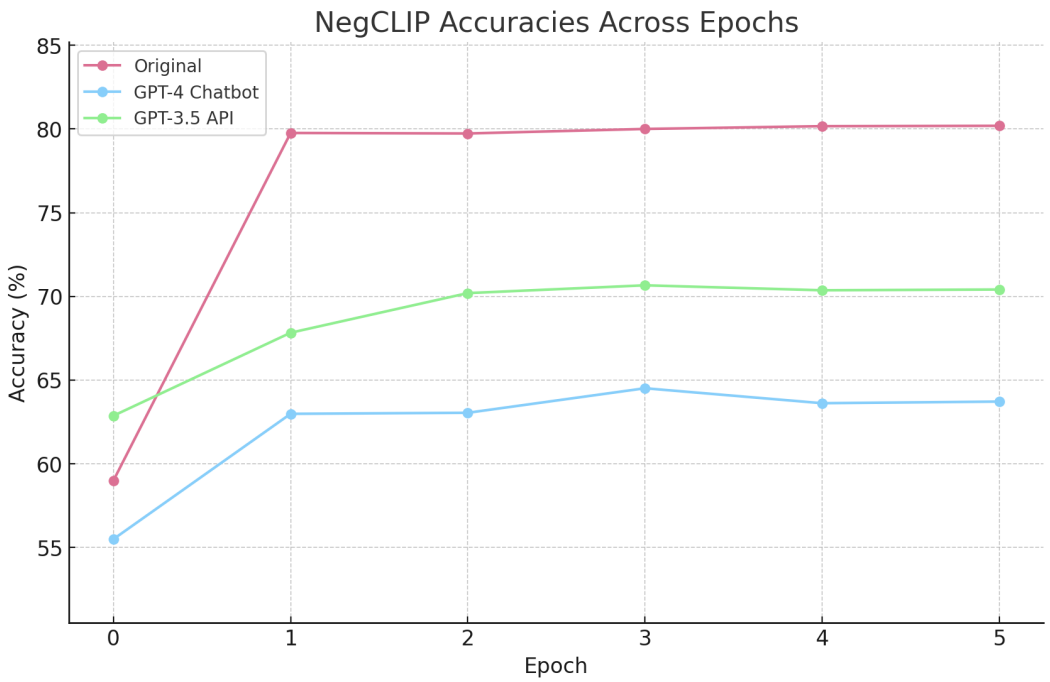

Figure 12: Evaluation on Checkpoints of NegCLIP Fine-Tuning

As expected, improvement on the original dataset is the most apparent, as the fine-tuning dataset uses a strict part-of-speech swapping procedure, just like the original ARO evaluation dataset. The benefit of fine-tuning goes down for the GPT based approaches, although there is still some increase in accuracy, showing that compositional-aware fine-tuning is relatively effective.

Notably, the base accuracy and improvement for GPT-3.5 API is better than GPT-4 Chatbot. This is interesting since the GPT-3.5 API dataset is heavily skewed according to our perplexity analysis.

6. Conclusion

This paper demonstrates that even models like ViLT that train for word patch alignment do not have a strong grasp of compositional relations. Moreover, existing datasets that create false captions with hard strict swapping can have significant unwanted biases. In particular, the false caption may be implausible or illogical, in which case the VLM does not need to even consider the image to answer correctly. Modern large language models show promise in enhancing these datasets to alleviate these biases. However, more work is needed to refine these approaches, as they may either fail to create plausible captions or fail to create a new caption altogether. Approaches should also consider the tradeoff between cost (monetary and run-time) and quality. Composition-aware fine-tuning is a valid approach for improving compositional reasoning. However, the dataset used for fine-tuning is crucial to consnider as well; improvements may not fully carry over to compositional questions generated in different ways.

Overall, we take a step in understanding VLMs' weakness in compositional awareness, and how we can begin to mitigate these holes. And thus we contribute towards a fundamental ability of artifical intelligence in the vision-language space.

6.1. Future Work

There are many valuable potential next steps. First is to have enough resources to bypass the high costs of the GPT-4 API, in order to take advantage of its higher quality outputs for better caption generation. Next would be to have some human evaluation of the datasets. We were only able to look at certain cases and quantitative metrics like perplexity, but since our goal is to align with our human compositional understanding, it would be invaluable to have a manual survey to compare the validity and coherency of captions.

Another promising avenue is to experiment more with ViLT. Focusing on the word patch alignment

objective during training may improve compositional awareness, as well as building the code infrastructure to fine-tune ViLT with composition-aware hard negatives as we did with CLIP.

Moreover, we would ideally apply CompGPT to the composition aware fine-tuning dataset. This would allow us to observe how our proposed caption generation methods compare in terms of improving compositonal understanding via fine-tuning.

7. Acknowledgements

I would first like to thank Dr. Felix Heide. His enlightening seminar, thoughtful questions, and astute guidance gave me the ability to complete this project at the level I desired. I would also like to thank Ilya Chugunov for guiding us through neural radiance fields and the Della cluster. Thank you as well to the TAs, Michael Tang and Shazra Raza, for their assistance. And finally, thank you to the rest of the students and my friends in their seminar for their support.

References

[1] "Openai pricing," accessed: [2024-01-06]. [Online]. Available: https://openai.com/pricing
[2] A. C. A. M. de Faria *et al.*, "Visual question answering: A survey on techniques and common trends in recent literature," 2023.
[3] H. Face, "Gpt-2," https://huggingface.co/gpt2, 2024, accessed: 2024-01-09.
[4] H. Face, "Transformers: State-of-the-art natural language processing," https://huggingface.co/docs/transformers/model_doc/vilt#transformers.ViltForImageAndTextRetrieval, 2024, accessed: 2024-01-06.
[5] C.-Y. Hsieh *et al.*, "Sugarcrepe: Fixing hackable benchmarks for vision-language compositionality," 2023.
[6] W. Kim, B. Son, and I. Kim, "Vilt: Vision-and-language transformer without convolution or region supervision," 2021.
[7] T.-Y. Lin *et al.*, "Microsoft coco: Common objects in context," *European conference on computer vision*, pp. 740–755, 2014.
[8] A. Radford *et al.*, "Learning transferable visual models from natural language supervision," 2021.
[9] Raf, "What are tokens and how to count them?" 2023, accessed: [2024-01-06]. Available: https://help.openai.com/en/articles/4936856-what-are-tokens-and-how-to-count-them
[10] T. Thrush *et al.*, "Winoground: Probing vision and language models for visio-linguistic compositionality," 2022.
[11] E. Wang, "Investigating clip's understanding of the vision-language space," 2023, final project for Computational Models of Cognition, Griffiths, Princeton University.
[12] M. Yuksekgonul *et al.*, "When and why vision-language models behave like bags-of-words, and what to do about it?" 2023.

8. Appendix

The code for this paper can be found at `https://github.com/ewang-pu/CompVLMs`, which we built off of the original ARO codebase.

Fine-tuning CLIP was done in this repo: https://github.com/ewang-pu/NegTrain, built off the corresponding ARO code. [12]

I will be working on commenting and cleaning these repositories. Please email me with any questions or clarifications!

Figure 13: Image from ARO with captions relating elephant and path. Rule-generated false caption is nonsensical for this image. The true caption is "the elephant is walking on the path"; false caption is "the path is walking on the elephant". We can tell something is severely biased here: the false caption is illogical and implausible from our human understanding. Our quantitative approach corroborates this finding. GPT-2 assigns a perplexity of 145.4 to the true caption and 396.7 to the false caption. GPT-4 Chatbot approach produces the new caption "the elephant is walking to the left of the path", which has a perplexity of 63.6. [11]

Figure 14: Image frmo ARO with captions relating man and shirt. Rule-generated false caption is nonsensical for this image. The true caption is “the man is wearing the shirt”; false caption is ”the shirt is wearing the man”. We can tell something is severely biased here: the false caption is illogical and implausible from our human understanding. Our quantitative approach corroborates this finding. GPT-2 assigns a perplexity of 89.1 to the true caption and 380.5. The GPT-4 Chatbot approach produces the new caption “the man is holding the shirt”, which by inspection is much more plausible and has a perplexity of 136.8.[11]

```
(visual): VisualTransformer(
  (conv1): Conv2d(3, 768, kernel_size=(32, 32), stride=(32, 32), bias=False)
  (ln_pre): LayerNorm((768,), eps=1e-05, elementwise_affine=True)
  (transformer): Transformer(
    (resblocks): ModuleList(
      (0-11): 12 x ResidualAttentionBlock(
        (attn): MultiheadAttention(
          (out_proj): NonDynamicallyQuantizableLinear(in_features=768, out_features=768,
    bias=True)
        )
        (ln_1): LayerNorm((768,), eps=1e-05, elementwise_affine=True)
        (mlp): Sequential(
          (c_fc): Linear(in_features=768, out_features=3072, bias=True)
          (gelu): QuickGELU()
          (c_proj): Linear(in_features=3072, out_features=768, bias=True)
```

```
        )
        (ln_2): LayerNorm((768,), eps=1e-05, elementwise_affine=True)
      )
    )
  )
  (ln_post): LayerNorm((768,), eps=1e-05, elementwise_affine=True)
)
(transformer): Transformer(
  (resblocks): ModuleList(
    (0-11): 12 x ResidualAttentionBlock(
      (attn): MultiheadAttention(
        (out_proj): NonDynamicallyQuantizableLinear(in_features=512, out_features=512,
    bias=True)
      )
      (ln_1): LayerNorm((512,), eps=1e-05, elementwise_affine=True)
      (mlp): Sequential(
        (c_fc): Linear(in_features=512, out_features=2048, bias=True)
        (gelu): QuickGELU()
        (c_proj): Linear(in_features=2048, out_features=512, bias=True)
      )
      (ln_2): LayerNorm((512,), eps=1e-05, elementwise_affine=True)
    )
  )
)
(token_embedding): Embedding(49408, 512)
(ln_final): LayerNorm((512,), eps=1e-05, elementwise_affine=True)
```

Listing 1: CLIP Model Architecture

```
I will give you an input caption describing a scene. Your task
is to:
1. Find any verb or spatial relationships between two
objects in the caption. If there are no such
relationships, return an empty list.
2. Randomly pick one relationship.
3. Replace the selected relationship with a new
relationship to make a new caption.
The new caption must meet the following three
requirements:
1. The new caption must be describing a scene that is
as different as possible from the original scene.
2. The new caption must be coherent, fluent, and grammatically
correct.
3. The new caption must make logical sense.
Here are some examples:
Original caption: the man is in front of the building
Relationships: ["in front of"]
Selected relationship: "in front of"
New relationship: behind
New caption: the man is behind the building
Original caption: the horse is eating the grass
Relationships: ['eating']
Selected relationship: eating
New relationship: jumping over
New caption: the horse is jumping over the grass
```

Figure 15: Input used for GPT-4 False Caption Generation