Vision-Language Models Do *Not* Understand Negation

Shaden Alshammari*[1] Kumail Alhamoud*[1] Yonglong Tian[2] Guohao Li[3]

Philip Torr[3] Yoon Kim[1] Marzyeh Ghassemi[1]

[1]Massachusetts Institute of Technology (MIT) [2]Google Research [3]University of Oxford

Abstract

Negation is crucial for precise communication, yet vision-language models (VLMs) often fail to understand it accurately. This study addresses two key questions: how well VLMs currently understand negation and how different training objectives affect their ability to understand negation. We introduce NegBench, a new benchmark for evaluating multimodal negation understanding through two tasks: Multiple Choice Questions with Negated Captions (MCQ-Neg) and COCO Retrieval with Negation (Retrieval-Neg). Our evaluation of six models reveals significant shortcomings in their negation understanding, with performance often at or below chance level. Further analysis shows that models employ various shortcut strategies, such as clustering sentences by template, failing to grasp negation properly. Our findings emphasize the need for improved training methods to enhance negation understanding in VLMs.

1. Introduction

Negation is an integral part of precise communication, allowing the speaker to specify what is false, prohibited, absent, or undesirable [5, 6]. For example, a user seeking unobstructed images of the beach for a phone wallpaper might search for images of the beach with *no* people'. Similarly, a radiologist seeking help diagnosing a novel X-ray image might search for cases displaying bilateral consolidation with *no* evidence of pneumonia'. Without understanding negation, a system could mistakenly retrieve records showing both conditions, leading to erroneous diagnoses. This work investigates two main questions:

Q1: How good are current vision-language models at understanding negation?
Q2: How do different VLM training objectives result in different negation understanding failure modes?

To answer the first question, we introduce NegBench, a new framework to evaluate multimodal negation understanding. Our NegBench benchmark includes two tasks: (i) the Multiple Choice Question with Negated Captions (MCQ-Neg) task, which provides a fine-grained diagnosis of negation understanding by evaluating models using structured caption templates; and (ii) the COCO Retrieval with Negation (Retrieval-Neg) task, which evaluates the model's ability to handle freeform negated captions in a larger-scale text-to-image retrieval setting. We test six models that include three different pretraining datasets, two model sizes, and three training algorithms. Our findings indicate that evaluated models, including one explicitly trained for negation understanding, perform at or below chance level in the realistic MCQ-Neg task, revealing significant shortcomings in their ability to comprehend negation.

To address the second question, we design a diagnostic tool to investigate how VLMs encode negated prompts. Our analysis shows that models trained with different objectives learn various shortcut strategies, such as cluster-

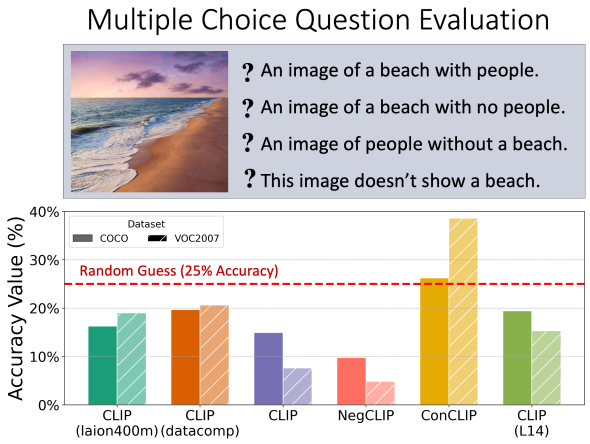

Figure 1. **Can Vision-Language Models Understand Negation?** We test six models on multiple-choice questions with negated captions (top). Most perform at or below random guess accuracy, revealing their struggle with negation comprehension (bottom).

ing sentences by template in the embedding space. Current VLMs struggle with negation understanding because they associate visual scenes with object keywords, often overlooking negation words like 'no'. Recent work hypothesized that models trained with the Contrastive Language-Image Pretraining (CLIP) objective may employ a shortcut strategy [4] that ignores the order of words in a sentence, resulting in a model that behaves like a bag-of-words [14]. While this explains CLIP's failure at understanding negation, it remains unclear why models specifically trained for negation, such as CoNCLIP [13], and those trained for order and compositionality, such as NegCLIP [14], also struggle. Our tool categorizes the types of errors each model makes, providing insights into potential improvements.

2. The Negation Benchmark (NegBench)

Existing benchmarks studying negation understanding have limitations: they focus on LLMs, not VLMs [3]; prioritize compositional reasoning over negation [9, 14]; and lack detailed evaluation of model biases towards negation-related linguistic templates [3, 9, 14]. To address these issues, we introduce NegBench, a benchmark specifically designed to evaluate multimodal negation understanding.

2.1. MCQ with Negated Captions (MCQ-Neg)

Our evaluation begins with a classification task. Given an image containing a set of positive elements $\{pos\}$ and excluding a set of negative elements $\{neg\}$, we test whether the model can correctly identify image descriptions that include negated statements. To describe an image, we generate captions using three templates based on three linguistic concepts: Affirmation, Negation, and Conjunction [7].

1. **Affirmation**: "This image includes x (and y)."
2. **Negation**: "This image does not include x."
3. **Hybrid** (combines Affirmation and Negation with the Conjunction 'but'): "This image includes x but not y."

Here, x and y are elements from $\{pos\}$ or $\{neg\}$. A caption accurately describes the image if it correctly affirms the presence of $\{pos\}$ elements and/or negates the presence of $\{neg\}$ elements. We use these templates to assess model biases and comprehension failures. A False Affirmation (e.g., "This image includes x", where $x \in \{neg\}$) indicates a misidentification of objects, revealing a vision failure. A False Negation (e.g., "This image does not include x", with $x \in \{pos\}$) indicates a lack of negation understanding. Finally, a False Hybrid (e.g., "This image includes x but not y", with $x \in \{neg\}$ and $y \in \{pos\}$) indicates an inability to handle negation or conjunction.

We utilize the COCO [8] and VOC2007 [1] datasets, which provide object-level annotations. We identify positive elements $\{pos\}$ as objects present in the image. To obtain negative elements $\{neg\}$, we pre-compute object co-occurrence statistics across the dataset and select the top three objects that frequently co-occur with the positive elements but are not present in the current image. By explicitly negating these commonly associated yet absent objects in our captions, we test the model's ability to grasp the concept of absence rather than being biased by frequent associations. We generate multiple random MCQs for each image. Each MCQ consists of one correct answer and three incorrect answers. The incorrect answers are designed to test model bias using Affirmation, Negation, and Hybrid templates. The correct answer is randomly assigned to one of these three templates. An example is shown in Figure 1.

2.2. Retrieval with Negation (Retrieval-Neg)

The MCQ-Neg task provides a detailed analysis of a VLM's ability to understand negation by examining the specific types of errors it makes with template-based multiple-choice questions. In Retrieval-Neg, we evaluate a model's ability to perform text-to-image retrieval with negated captions. This task introduces two unique challenges compared to the MCQ task: (i) *Realistic User Input*: Users can input unconstrained hybrid captions that combine affirmation and negation without relying on templates, and (ii) *Scalability*: The model must retrieve the correct image from a database containing thousands of images.

To perform this evaluation, we modify the traditional COCO retrieval task. For each caption, we add a negated statement, resulting in captions like: "There is no x in the image. [Original Caption]." or "[Original Caption]. There is no x in the image." Here, $x \in \{neg\}$ represents an object that is not present in the image but is commonly associated with the present objects. The model ranks images from the database according to how well they match the caption.

3. Evaluating on NegBench

We evaluate six models with differences in pretraining datasets, model sizes, or training approaches. Notably, CLIP [10], CLIP-laion400m [12], and CLIP-datacomp [2] vary in their pretraining datasets. In contrast, CLIP, CoNCLIP [13] (designed to improve negation understanding), and NegCLIP [14] (designed to improve compositional reasoning) differ in training strategies. Finally, CLIP-L14 is a larger version of CLIP.

Models struggle with negation. Figure 1 shows performance on MCQ-Neg. Most models perform worse than random guessing. The base CLIP model achieves 15% on COCO and 8% on VOC2007. This subpar performance likely stems from CLIP's pre-training objective, which biases the model to associate visual concepts with specific words. This bias is a disadvantage in tasks involving negated captions. The NegBench answer choices are designed to expose such biases, often misleading models like CLIP. For instance, as shown in Figure 1, all three incorrect answers mention 'beach', which might mislead CLIP.

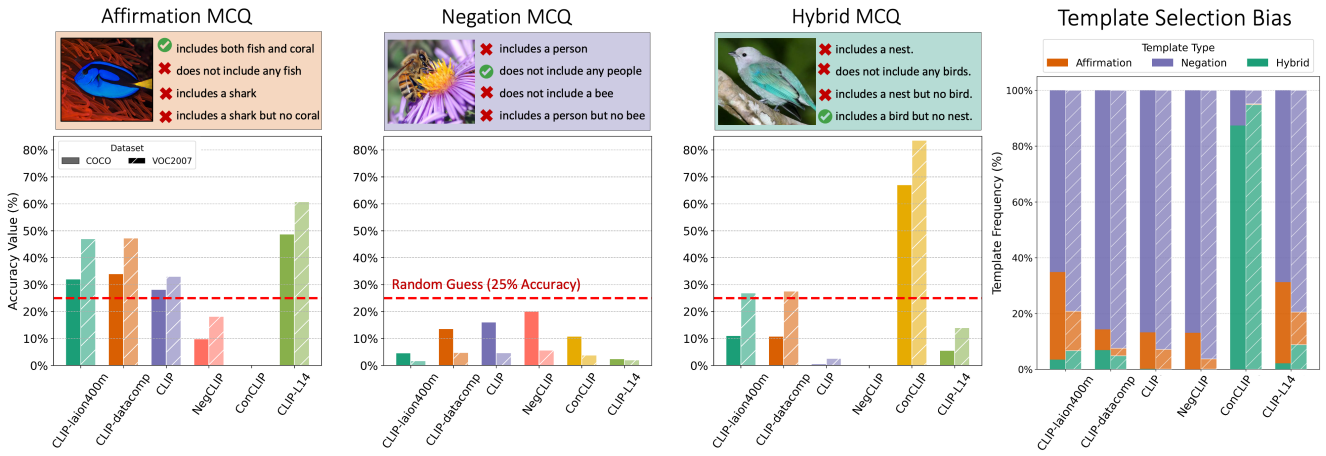

Figure 2. **How Do Models Perform Across Different MCQ Types?** We compare accuracies on three types of multiple-choice questions (MCQs): Affirmation (left), Negation (middle), and Hybrid (right). All models underperform on Negation MCQs. The fourth panel shows the frequency with which models select specific templates. For instance, ConCLIP shows a strong bias towards hybrid sentence structures.

ConCLIP performs better, especially on VOC2007, where it achieves an accuracy above 40%. Yet, as shown in Figures 2 and 4, the improvements of ConCLIP are due to another shortcut strategy rather than genuine negation understanding. To understand why some models perform below random chance, we categorize the MCQs and show how models with different biases fail certain question types.

Models show varied performance across MCQ types. Each MCQ has a correct answer assigned to one of three templates: Affirmation, Negation, or Hybrid. This results in three MCQ types: Affirmation MCQs, Negation MCQs, and Hybrid MCQs. Figure 2 compares model accuracies for these MCQ types: Affirmation MCQs on the left, Negation MCQs in the middle, and Hybrid MCQs on the right.

CLIP-Large improves on CLIP's Affirmation and Hybrid MCQ accuracies but has lower accuracy on Negation MCQs. ConCLIP shows a strong preference for hybrid sentences, completely failing in correctly identifying affirmative sentences with 0% Affirmative MCQ accuracy on both datasets. Conversely, NegCLIP scores 0% on Hybrid MCQs, indicating a bias against hybrid sentences. All models perform poorly on Negation MCQs, indicating a general difficulty with negation understanding.

Template Selection Frequency To explain the disparities in accuracy by MCQ type, we analyze model biases towards linguistic structures in Figure 2. The Y-axis shows the percentage of times a model selects captions corresponding to the Affirmation, Negation, or Hybrid templates.

We make two key observations. First, ConCLIP overpredicts the Hybrid template and never predicts the Affirmation template. This explains its high Hybrid MCQ accuracy and 0% Affirmation MCQ accuracy. In Section 4, we will show that all Hybrid captions ("This image includes X but not Y") are mapped to the same location in the ConCLIP embedding space. The overprediction of the Hybrid template suggests that the image encoder maps most images to where the text encoder maps the Hybrid captions. ConCLIP is likely overly tuned to recognize Hybrid descriptions.

Second, most models frequently select the Negation template. This tendency can be attributed to the task design, where 67% of the MCQs (Negation and Hybrid MCQs) do not present the choice "This image includes pos." CLIP-like models, which associate visual concepts with specific words, default to selecting the negation-based answer "This image does not include pos" when the affirming option is unavailable. Overall, this analysis suggests that models are learning linguistic shortcut strategies, a hypothesis that will be further explored in Section 4.

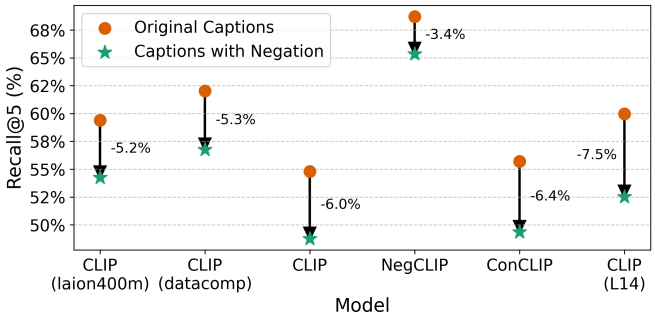

Figure 3. **How Does Negation Impact Retrieval Performance?** We compute the recall@5 of six models on the original COCO retrieval task (orange circles) and the Retrieval-Neg task with negated captions (green stars). All models show a significant drop in performance when negation is introduced, with the percentage decrease indicated by the black arrows.

Evaluating VLMs on Retrieval-Neg We compare the per-

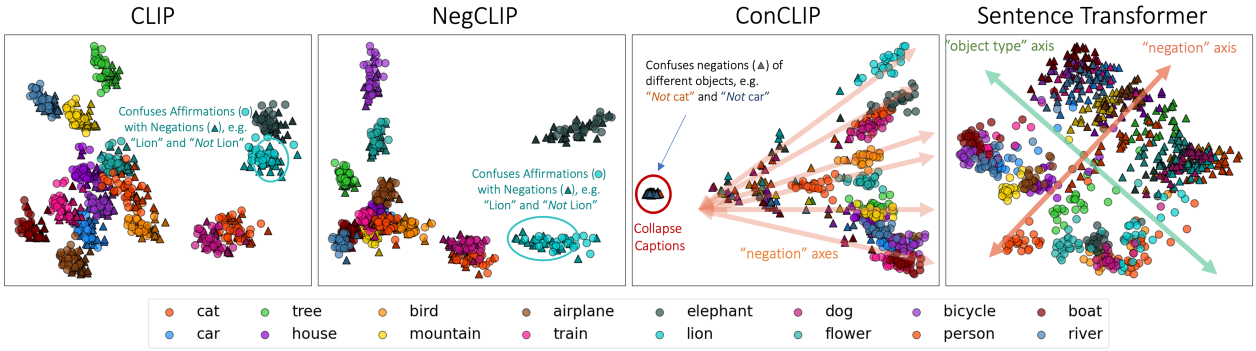

Figure 4. **PCA projections of Affirmative (dots) and Negative (triangles) Caption Embeddings for Multiple Objects.** The Sentence Transformer shows clear separation along 'object type' and 'negation' axes, while CoNCLIP demonstrates compression along negation axes (treating all negated captions as identical), and both CLIP and NegCLIP lack separation between affirmative and negated captions.

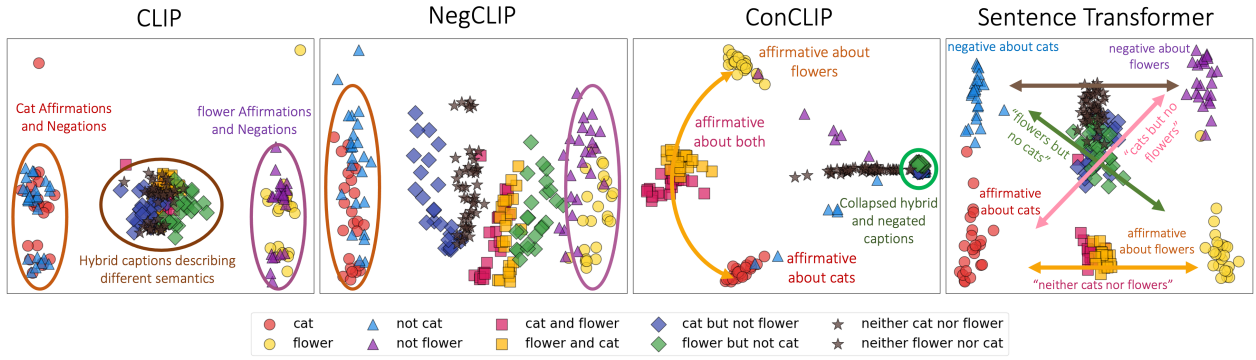

Figure 5. **PCA of Affirmative (dots and squares), Negated (triangles and stars), Hybrid captions (diamonds).** ConCLIP compresses negated and hybrid captions (green circle), while CLIP and NegCLIP fail to separate affirmative and negated captions (red and purple ovals). Sentence Transformer exhibits clear separability and semantic representation, exemplifying the desired behavior.

formance of six models on the original COCO retrieval task and the Retrieval-Neg task, where captions include negated statements. Figure 3 shows this comparison. All six models perform worse on the Retrieval-Neg task than on the original COCO task, indicating that they struggle with processing negation in the context of multimodal retrieval.

4. Why Do VLMs *Not* Understand Negation?

To analyze shortcut strategies in models, we use 24 Affirmative ("X") and 24 Negative ("Not X") templates to create 48 captions per object. These captions are embedded by different models and analyzed via PCA in Figure 4.

We observe varying behaviors among the models. The overlapping embeddings for affirmative and negated captions in *CLIP* and *NegCLIP* suggest that these models do not effectively distinguish between positive and negative statements, possibly due to a "bag-of-words" shortcut strategy [4, 14] that overlooks negation words. This explains why both models incorrectly select the Negation template, which negates positive objects, in Figure 2. *CoNCLIP* separates positive and negative captions but fails to distinguish between negative captions of different objects, collapsing all negative caption embeddings toward a single point (red circle). We further show an example of an embedding space for a model that understands negation. The text-only *Sentence Transformer* model [11] effectively separates affirmative and negated captions along distinct "object type" and "negation" axes, indicating better differentiation.

Figure 5 extends the previous analysis to hybrid captions that combine affirmations and negations. It provides further evidence that *ConCLIP* employs a shortcut strategy for embedding linguistic negation, with hybrid and negated captions collapsing towards a single point (green circle), indicating significant compression along the negation axis. While *CLIP* and *NegCLIP* struggle to distinguish affirmative from negative statements, *NegCLIP* shows better sepration for hybrid captions, which appear collapsed in the CLIP embedding space. This suggests that NegCLIP's poor performance on Hybrid MCQs might be due to a misalignment between the text and image encoders, rather than an inability to understand hybrid sentence structure. In contrast,

the *Sentence Transformer* effectively distinguishes between different caption types and provides semantically guided representations. For example, it aligns "flowers but not cats" along the line connecting "cats" and "not flowers." We hypothesize that future works aiming to improve negation understanding in multimodal retrieval may benefit from creating a similarly separable embedding space.

References

[1] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88, 2010. 2

[2] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, Eyal Orgad, Rahim Entezari, Giannis Daras, Sarah M Pratt, Vivek Ramanujan, Yonatan Bitton, Kalyani Marathe, Stephen Mussmann, Richard Vencu, Mehdi Cherti, Ranjay Krishna, Pang Wei Koh, Olga Saukh, Alexander Ratner, Shuran Song, Hannaneh Hajishirzi, Ali Farhadi, Romain Beaumont, Sewoong Oh, Alex Dimakis, Jenia Jitsev, Yair Carmon, Vaishaal Shankar, and Ludwig Schmidt. Datacomp: In search of the next generation of multimodal datasets. In *NeurIPS Datasets and Benchmarks Track*, 2023. 2

[3] Iker García-Ferrero, Begoña Altuna, Javier Alvez, Itziar Gonzalez-Dios, and German Rigau. This is not a dataset: A large negation benchmark to challenge large language models. In *EMNLP*. Association for Computational Linguistics, 2023. 2

[4] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11), 2020. 2, 4

[5] Laurence R. Horn. *A Natural History of Negation*. University of Chicago Press, 1989. 1

[6] Michael P Jordan. The power of negation in english: Text, context and relevance. *Journal of pragmatics*, 29(6), 1998. 1

[7] Miren Itziar Laka Mugarza. *Negation in syntax–on the nature of functional categories and projections*. PhD thesis, Massachusetts Institute of Technology, 1990. 2

[8] 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 *ECCV*, 2014. 2

[9] Zixian Ma, Jerry Hong, Mustafa Omer Gul, Mona Gandhi, Irena Gao, and Ranjay Krishna. Crepe: Can vision-language foundation models reason compositionally? In *CVPR*, 2023. 2

[10] 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 *ICML*. PMLR, 2021. 2

[11] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP*. Association for Computational Linguistics, 2019. 4

[12] 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

[13] Jaisidh Singh, Ishaan Shrivastava, Mayank Vatsa, Richa Singh, and Aparna Bharati. Learn" no" to say" yes" better: Improving vision-language models via negations. *arXiv preprint arXiv:2403.20312*, 2024. 2

[14] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? In *ICLR*, 2023. 2, 4