Enhancing Semantic Understanding in Vision Language Models Using Meaning Representation Negative Generation

Ziyi Shou, Fangzhen Lin

HKUST-Xiaoi Joint Laboratory
Department of Computer Science and Engineering
Hong Kong University of Science and Technology

Abstract

Vision language models have been criticized for their performance resembling bag-of-words models, lacking semantic understanding. Efforts to address this concern have included the integration of composition aware negative samples into contrastive learning methodologies. However, current negative generation methods show restricted semantic comprehension, diversity, and fluency. To tackle this issue, we propose leveraging Abstract Meaning Representation (AMR), a representation of considerable interest in natural language processing research, for negative sample generation. By altering the structure of the meaning representation, we create negative samples with entirely different meanings but share close plain paraphrases. These negatives, generated using AMR, are then incorporated alongside token swap negatives during contrastive training. Our results indicate that AMR generated negatives introduce significantly diverse patterns. Furthermore, the inclusion of AMR generated negative samples enhances the models' performance across a range of compositional understanding tasks.

Keywords

Vision Language Models, Semantic Understanding, Compositional Understanding, Abstract Meaning Representation

1. Introduction

In recent years, the conspicuous development of vision language models (VLMs) across various tasks is evident [1, 2, 3]. However, VLMs have been criticized for performing akin to bag-of-words models, lacking semantic understanding, especially compositional understanding [4, 3, 5]. For instance, when some tokens in the caption of an image-caption pair are rearranged to result in an unaligned caption, a VLM may fail to notice the change. Consider the two image-caption pairs in Figure 1. In the left side pair, the phrases "Three Jack-O-Lanterns" and "flowers" in its caption are swapped, resulting in a semantically very different sentence. But CLIP fails to notice the difference and somehow gives the modified caption a slightly higher similarity score. A similar effect can be seen in the right side image-caption pair, when the phrases "Clock tower" and "a bronze statue" in its caption are swapped. These are not isolated examples. As Yuksekgonul et al. [5] pointed out, VLMs "behave like bags-of-words" because they have been mostly pretrained on large-scale web datasets for retrieval tasks where image and caption matching can often be done using keywords alone.

A straightforward and effective solution involves mining hard negative samples for contrastive learning. This entails including negative instances with similar semantic components but distinct relationships in the same batch, challenging the model to discern the correct caption amidst such variations. For example, NegCLIP [5] constructs negative image captions by swapping tokens. However, token swap methods lack semantic understanding, resulting in patterns, and lack of plausibility and fluency. Blind Models trained solely on text, without considering images, may manipulate evaluations to their advantage [6].

Meaning representations offer an alternative approach to constructing negative samples with greater diversity and fluency. Abstract Meaning Representation (AMR, [7]) stands out as a prevalent semantic representation in text tasks and is valued for its high expressiveness and human-friendly comprehension, which encodes concepts as nodes and depicts the relationships between concepts through graphical representations. We propose to utilize AMR to create negative samples that possess entirely distinct meanings but share close plain paraphrases. To achieve this, we modify the structure of meaning representation by randomly shuffling the positions of subtrees within AMR graphs and reconstructing meaning representations. Following this process, negative captions are generated from the new meaning representations using an AMR generator. We blend our generated negatives with token swap negatives to broaden the diversity of negative samples and enhance generalization. Subsequently, vision language models undergo training to distinguish between true labels and negative samples.

Our findings indicate that incorporating negative samples generated from meaning representations improves model performance across diverse compositional understanding benchmarks. Additionally, our generated nega-

KiL'24: Workshop on Knowledge-infused Learning co-located with 30th ACM KDD Conference, August 26, 2024, Barcelona, Spain
zshou@cse.ust.hk (Z. Shou); flin@cse.ust.hk (F. Lin)
© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

Aligned	Three Jack-O-Laturns of various shapes, one of which has flowers in it.	CLIP Score: 0.273
Unaligned	Flowers of various shapes, one of which has Three Jack-O-Lanterns in it.	CLIP Score: 0.288

Aligned	Clock tower with a bronze statue on top on a sunny day.	CLIP Score : 0.301
Unaligned	A bronze statue with a clock tower on top on a sunny day.	CLIP Score: 0.306

Figure 1: Example test results of the model's relational understanding. CLIP gives higher similarity scores for unaligned captions.

tives introduce various patterns, enriching the diversity of augmentations compared to token swap negatives.

2. Related Work

2.1. AMR Data Augmentation

AMR encodes concepts as nodes and illustrates the relationships between these concepts as edges. It has been shown to be advantageous in various natural language processing tasks, such as data augmentation. Token edit data augmentations in NLP often result in generating ill-formed or incoherent sentences, as they do not consider sentence structures. AMR Data Augmentation (AMR-DA) [8] suggests utilizing AMR for data augmentation. They construct positive samples by meticulously controlling minor nuances within a carefully designed framework for meaning representation. Consequently, they produce several fluent and distinct positive augmentations for the given sentences. Inspired by AMR-DA, we explore the utilization of AMR in compositional understanding tasks for vision language models. However, our approach diverges significantly; rather than focusing on careful modifications to meaning representation for positive sample generation, we propose employing AMR for negative sample generation. Our methodology involves splitting the meaning representation and shuffling its components to construct a new negative representation.

2.2. Composition-aware Hard Negatives

For generating negative captions for contrastive learning, a straightforward approach involves modifying linguistic elements. To improve compositional understanding, [5] leverage Spacy for syntactic analysis to identify and swap the positions of two elements within the caption. The token swap modifications aimed at creating variations in composition are relatively straightforward to implement but often struggle to maintain grammaticality. Moreover, they can be vulnerable to exploitation, as the patterns of modification may become predictable even without considering information from the image encoder. [9] initially parse the syntactic structure of the caption. They then randomly mask text and utilize a large language model to unmask and generate a new negative caption. While the resulting caption tends to exhibit improved grammatical correctness, the modification process lacks fine control, and the generated variants remain somewhat constrained in scope. To address the limitations of semantic modification, [10] proposes leveraging scene graphs to generate semantic negative captions. They implement a strategy where they interchange the positions of the subject and object within the same relation, as well as swap the attributes of different objects. However, the modification of scene graphs is limited. Compared to scene graphs, meaning representations encode a more extensive range of relations, especially higher-level abstract semantic relations absent in scene graphs [11]. This suggests that meaning representations have a higher potential to improve downstream tasks that require an understanding of higher-level semantic information in images.

3. Methods

3.1. Extensive Contrastive Learning

The aim of contrastive learning is to bring similar representations into closer proximity while simultaneously pushing apart dissimilar samples. This principle mirrors its application within vision language model training, exemplified by Contrastive Language-Image Pre-Training (CLIP, [1]), which has emerged as a prominent paradigm in vision language learning. The training objective of CLIP is to align text-image pairs effectively. CLIP simultaneously trains an image encoder and a text encoder to extract feature representations from each modality, denoted as I_n for image features and T_n for text features. These features are then utilized to compute scaled pairwise cosine

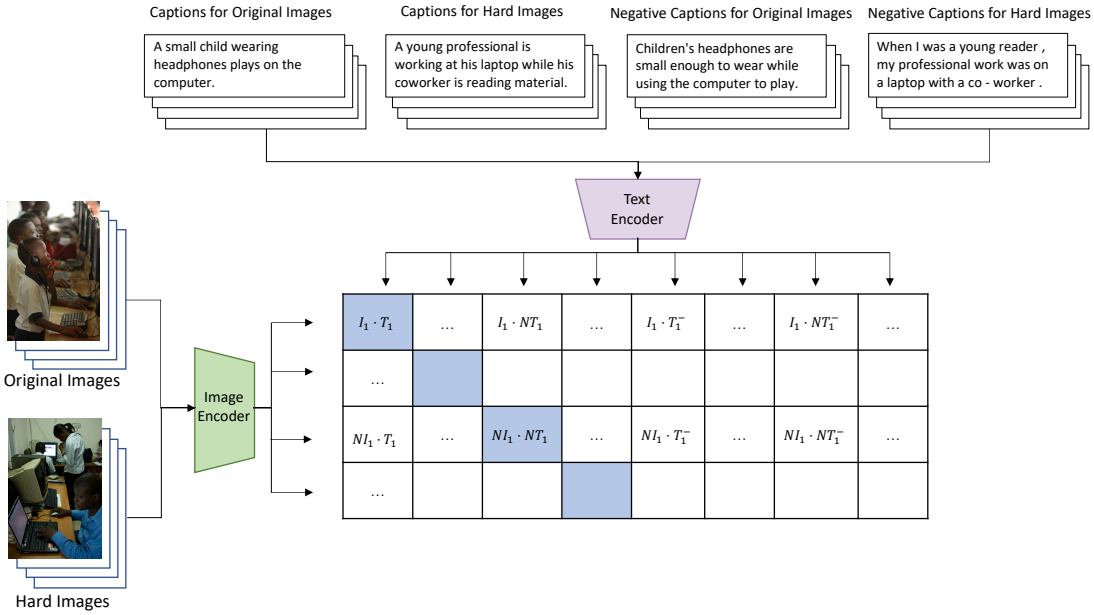

Figure 2: Extensive CLIP for compositional understanding tasks through extensive training with hard neighbor images and AMR generated hard negative captions.

similarities, serving as logits. Finally, a symmetric cross-entropy loss is computed over these similarity scores to guide the training process effectively.

In response to the challenge of vision language models struggling to comprehend text composition, we adopt the approach proposed by Yuksekgonul et al. [5], which introduced two extensive components to standard contrastive learning, aimed at increasing the complexity of model learning. This entails (1) introducing challenging images for the image encoder to extract features from, selected based on CLIP encoding and utilizing nearest neighbors of original images, and (2) incorporating hard negative captions for the text encoder to distinguish features. The difference is that we add AMR generated negative samples into hard negative captions, with modifications aimed at preserving most plain text tokens while completely distorting the semantic meaning. Figure 2 illustrates the training pipeline. In each batch, original images I_n and their nearest neighbors NI_n are included. Corresponding captions T_n and NT_n are concatenated with hard negative captions T_n^- and NT_n^-, doubling the length of captions compared to the number of images. Subsequently, a symmetric cross-entropy loss is computed as in CLIP. However, only column-wise loss for positive captions is incorporated, as negative captions lack corresponding images for comparison.

3.2. AMR for Negative Sample Generation

Contrary to token swap negative generation, we propose to the generation of negative samples using AMR. AMR encodes the semantics into graphs and has demonstrated effectiveness as an intermediate representation in natural language augmentation tasks. We adopt a similar pipeline to AMR-DA [8]: parsing sentences into AMR, modifying the AMR, and generating samples from the modified AMR. However, our objective differs significantly from that of AMR-DA. While they meticulously modify the intermediate AMR to construct positive samples, our task requires generating entirely different semantic representations, albeit with the same semantic components as given samples.

3.2.1. Meaning Representation

Abstract Meaning Representation (AMR, [7]) is a rooted, directed graph that encodes sentence concepts as nodes and the relations between these concepts as directed edges. In Figure 3, the leftmost portion depicts the AMR graph corresponding to the caption "A trunk carries a large amount of items and a few people." In this graph, the root "carry" serves as the primary predicate of the sentence, with "trunk" designated as the first argument (denoted as ARG0) of "carry," while the subtree originating from "and" represents the second argument. AMR

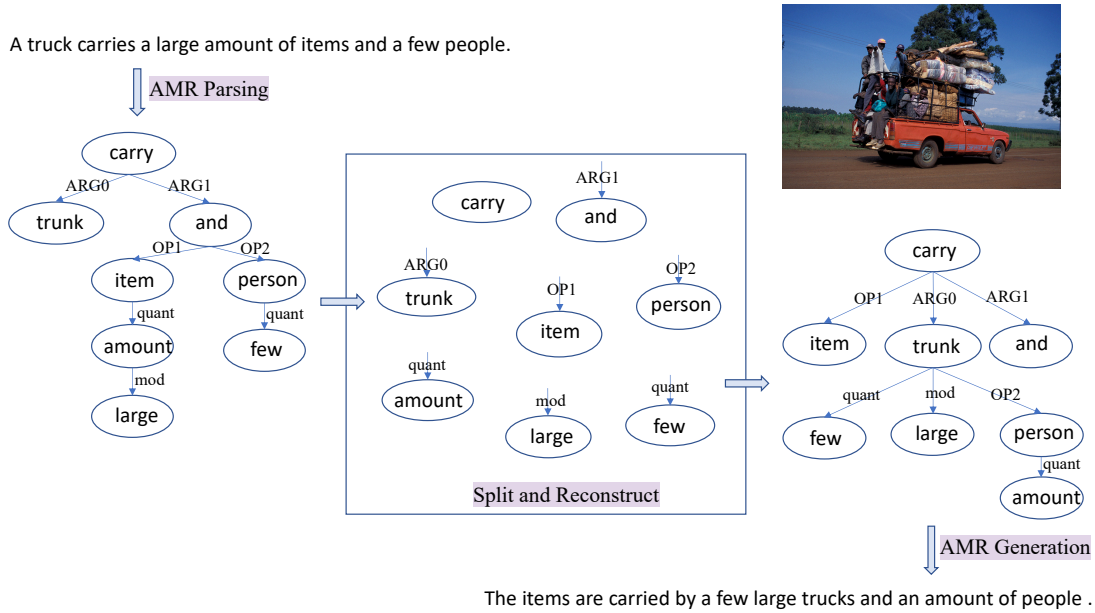

Figure 3: Negative example generated based on AMR. The shuffled AMR entails reordering all nodes along with their edges except the root node.

.

facilitates readability for both human and machine comprehension and can be adapted to various purposes as needed. In this study, our proposal involves splitting the AMR graph, shuffling its components, and then reconstructing a new AMR graph. This process aims to create a hard negative graph where all semantic parts are retained, but the overall meaning is distorted.

3.2.2. Generation Pipeline

The entire pipeline is illustrated in Figure 3. We adopt AMR-DA pipeline, which involves initially parsing the caption into an AMR graph using an AMR parser. Subsequently, we modify this AMR graph and finally utilize an AMR generator to produce negative captions based on the modified AMR. We utilize SPRING parser [12] as our AMR parser. SPRING employs a depth-first search method to linearize AMRs and utilizes a special token $< Rn >$ to manage co-referring nodes. The parser is trained based on BART model [13]. After obtaining the AMR graph for the caption, we propose a split and reconstruct algorithm to construct a new AMR graph, which is described in detail in the subsequent paragraphs. Finally, we employ PLMs-Generator [14] based on T5-base as our AMR generator to convert AMR to text. The model-based generator exhibits tolerance, allowing for the accommodation of certain unreasonable aspects within our modified graph. AMR generator can rectify to some extent and produce new samples closely resembling the given graph, this flexibility provides greater latitude for modifying the AMR graph compared to rule-based methods. For instance, in Figure 3, although the modified graph contains some illogical elements such as the node "and" lacking children, the generator is still capable of generating fluent and grammatically correct text.

3.2.3. AMR Split and Reconstruct

The key component of generating negative samples through AMR lies in our split and reconstruct algorithm. Unlike existing methods that rely on token swapping within the sentence or node swapping in the scene graph based on predefined rules, our approach offers greater flexibility by directly modifying the entire meaning representations. Modifications to AMR afford a broader range of possibilities owing to the diverse types of edges and nodes present.

In our algorithm, we split the AMR graph and regard the root node as a separate entity, while treating other nodes along with their incoming edges as edge-node pairs. As illustrated in Figure 3, the left-hand side depicts the AMR graph corresponding to the original caption "A trunk carries a large amount of items and a few people." Following the split process, we obtain a root node and a collection of edge-node pairs such as "carry, [(:ARG0, trunk), (:ARG1, and), ...]".

Algorithm 1 Negative AMR Generation

```
Ensure: Negative_G                                                    ▷ Output Negative AMR graph
Require: G                                                            ▷ Input AMR graph
  root_node, list_of_edge_node_pairs = split_graph(G)                 ▷ Split the graph
  list_of_edge_node_pairs = random.shuffle(list_of_edge_node_pairs)
  Negative_G←[(root, root_node)]
  Node_stack←[root_node]
  depth←1
  for edge, node in list_of_edge_node_pairs do
      choice = random.choice([*range(1, depth + 1, 1)])
      if choice = 1 then                                              ▷ To next level
          Negative_G.append(Node_stack[-1], edge, node)
          Node_stack.append(node)
          depth += 1
      else                ▷ choice = 2: At current level; choice = n: back to the previous N level
          move_forward_depth = choice - 2
          depth -= move_forward_depth
          while move_forward_depth > -1 do
              Node_stack.pop(-1)
              move_forward_depth -= 1
          end while
          Negative_G.append(Node_stack[-1], edge, node)
          Node_stack.append(node)
      end if
  end for
```

Next, we proceed to reconstruct a semantic tree by randomly concatenating nodes from the split parts. We shuffle the list of edge-node pairs and sequentially select edge-node pairs one by one. The process begins at layer 1 with the root node. At this stage, the first node has only one option, which is to connect to the root node and move to layer 2. Subsequently, at layer 2, the subsequent nodes have two options: either to remain at layer 2 by connecting to the root node, or to move to a deeper layer by connecting to the previous node at layer 2. If a node moves to a deeper layer, for instance, layer 3, the subsequent node has three options: to remain at the current layer, to move deeper, or to move back to the previous layer. This iterative process continues until all nodes are connected within the semantic tree. In Figure 3, when considering the pair (:mod, large), there are indeed three options available. The node "large" can either remain at the current layer by connecting to the node "trunk", proceed to a deeper layer by connecting to the node "few", or revert back to connect with the root node. The shuffled AMR entails reordering all nodes along with their edges except the root node, resulting in a new representation of meaning. Negative captions are then generated based on this shuffled AMR. The algorithm to reconstruct AMR graph is illustrated in Algorithm 1.

The distinction between negative AMR generation and AMR-DA lies in their respective objectives. AMR-DA aims to regulate modifications to avoid distorting the overall semantic meaning of the sentence by selectively adding or removing nuanced semantic components. On the other hand, negative AMR generation focuses on retaining the majority of the semantic components while generating entirely different semantic representations.

4. Experiments

We conduct experiments on different evaluation datasets to explore the impact of AMR generated negatives on the performance of vision language models in compositional understanding tasks.

4.1. Experimental Settings

We explore whether AMR generated negatives improve the performance of model compositional understanding, so we follow the training setups in NegCLIP[5], which finetune CLIP based on the ViT-B/32 [1] on the COCO dataset with token swap hard negatives.

For negative captions, we assign a specific probability to replace the original token swap caption with AMR generated negative augmentation. In the main results, the possibility of replacing negatives in NegCLIP is set at 30%. In other words, about 30% of the captions with our AMR generated hard negative captions, while the

[1] https://github.com/openai/CLIP

Table 1
ARO and SugarCrepe results comparison of AMR-NegCLIP with different models.

	ARO				SugarCrepe		
	Visual Gnome		Flickr30k	COCO	All Datasets Avg		
	Relation	Attribution	Order	Order	Replace	Swap	Add
ViT-B-32	51.1	61.3	47.2	37.1	80.8	63.3	75.1
CLIP	59.9	63.2	59.5	46.0	84.8	70.8	85.6
NegCLIP	81.0	71.0	91.0	86.0	85.4	75.3	87.3
AMR-NegCLIP	83.2	75.6	93.9	91.6	86.4	81.2	87.5

remainder with original token swap negative samples, are utilized for contrastive training. This approach ensures a diverse range of negatives is maintained. The comparison of different probabilities is included in Section 5.3. For each image, one of the three nearest negative neighbors, determined by CLIP encoding, is sampled as the hard image.

NegCLIP initially sets the batch size to 1024. However, due to device limitations, we are constrained to train the model on a single NVIDIA RTX 2080 Ti GPU, reducing our batch size to 32. Consequently, we adjust the warm-up steps to 1600. Contrastive learning relies on batch size, as it involves contrasting samples within each batch. Therefore, larger batch sizes are anticipated to yield greater improvements. We employ the AdamW optimizer with a cosine annealing schedule for a training epoch of 5. The learning rate is explored within the range of 1e-5, 5e-6, 1e-6, with reported results utilizing a learning rate of 5e-6.

4.2. Evaluation Dataset

We assess the efficacy of our approach on two widely used benchmarks for compositional understanding: ARO [5] and SugarCrepe [6]. ARO stands for **A**ttribution, **R**elation, and **O**rder, including four tasks: Visual Genome Relation (VG-Relation) and Visual Genome Attribution (VG-Attribution) tasks entail selecting the correct caption from two options, where negative captions alter either the object of the relation or the object's attribution. Flickr30k Order and COCO Order tasks demand models to accurately identify the order of captions from five options, where negative captions modify the order of tokens within the caption. SugarCrepe aims to address the issue of negative captions being implausible and non-fluent by employing large language models to generate fluent and challenging negative captions. The dataset encompasses three tasks: Replace, Swap, and Add, which entail various actions aimed at evaluating models' compositional understanding.

4.3. Main Results

We incorporate AMR generated negative samples into our contrastive training data, simplifying our method to AMR-NegCLIP. In this study, we undertake a comparative analysis of the outcomes generated by our AMR-NegCLIP approach in contrast to the results produced by several baseline models, ViT-B-32, standard CLIP finetuned with COCO dataset (CLIP), and CLIP finetuned with token-level hard negatives (NegCLIP).

From Table 1, we can find that our AMR-NegCLIP achieves superior performance across all subtasks. In Visual Gnome dataset, AMR-NegCLIP gets a 2.2% improvement in Relation task over NegCLIP and a 4.6% improvement in Attribution task. In Flickr30k Order dataset, there is a 2.9% improvement compared to NegCLIP and a substantial 34.4% improvement over CLIP. In the COCO Order dataset, there is a 5.6% improvement over NegCLIP and an impressive 45.6% improvement over CLIP. In Replace and Add tasks within SugarCrepe, AMR-NegCLIP exhibits limited improvements when contrasted with NegCLIP, with 1.0% in Replace task and 0.2% improvement in Add task. This discrepancy can be attributed to the nature of the Replace and Add tasks, which involve modifying concepts within the caption. AMR-NegCLIP generates negatives that maintain the same concepts as the positive caption, thereby not entirely aligning with the task requirements. In contrast, another notable observation is a significant improvement, 5.9% over NegCLIP, in the Swap task of SugarCrepe, a challenge that proves to be particularly daunting for pre-trained CLIP models, as highlighted in the SugarCrepe paper [6]. In their study, SugarCrepe authors evaluate over ten vision language models and note that "*all models struggle at identifying SWAP hard negatives, regardless of their pertaining dataset and model size.*". This difficulty arises from the nature of the swap action in SugarCrepe, which involves neither adding nor excluding any concepts but rather swapping objects or attributes while maintaining fluency and grammatical correctness, a task demanding a deeper understanding of composition from vision language models. This closely aligns with our motivation to employ meaning representations in the

Table 2
Example evaluation data of Visual Genome Relation, Flickr30k Order in ARO; Replace, Swap and Add in SugarCrepe. The italicized text represents a positive caption for the sample, while the other lines contain negative captions. Visual Genome includes two captions per sample, whereas Order test set includes five captions per sample.

Visual Genome Relation	*the door is to the left of the shirt.* the shirt is to the left of the door.
Flickr30k Order	*A group of people standing on the lawn in front of a building.* Many people in blue jeans stand in front of a white church. A large group of people stand outside of a church. Family members standing outside a home. People standing outside of a building.
SugarCrepe Replace	*A tan toilet and sink combination in a small room.* A white toilet and sink combination in a small room.
SugarCrepe Swap	*Three large horses eating hay while a small horse stands behind.* A small horse eating hay while three large horses stand behind.
SugarCrepe Add	*Two zebras are battling each other on hind legs.* Two striped-and-spotted zebras are battling each other on hind legs.

Table 3
Negative Sentences generated using Random Token Swap, Scene Graph Node Swap and AMR Reconstruction.

Source	A truck carries a large amount of items and a few people.
Random Token Swap	A amount carries a large truck of items and a few people .
Scene Graph Node Swap	A truck carries a few amount of items and a large people.
AMR Reconstruction	The items are carried by a few large trucks and an amount of people .
Source	A pigeon greets three bicyclists on a park path.
Random Token Swap	A park greets three bicyclists on a pigeon path .
Scene Graph Node Swap	A bicyclist greets three pigeon on a park path.
AMR Reconstruction	Greetings , three pigeon bicyclers on the path have been parkled .
Source	People walking pass a horse drawn carriage sitting at the curb.
Random Token Swap	People walking pass a horse drawn curb sitting at the carriage.
Scene Graph Node Swap	People sitting at a horse drawn carriage walking pass the curb.
AMR Reconstruction	People walking by the curb , horse sitting , carriage pulling .

negative generation. Example evaluation data for ARO and SugarCrepe are provided in Table 2.

In Order evaluation dataset, negative samples exhibit greater diversity. The introduction of Swap in SugarCrepe aims to rectify instances of textual non-fluency and implausibility, thereby rendering it more resilient against potential hacking attempts from blind models.

In conclusion, the results indicate that integrating AMR generated negative captions significantly improves VLM's performance on various composition tasks, especially dealing with high-level compositional understanding captions.

5. Analysis

5.1. Comparison with Scene Graph

Understanding the meaning of images has long been a goal. Scene graphs have emerged as a popular method for encoding objects, their attributes, and relationships within graphs. Abdelsalam et al.'s work [11] discusses the difference between AMR and Scene Graphs through detailed statistical analysis on entity and relation categorization. Their conclusion highlights that AMR encodes a broader range of relationships, particularly abstract semantic relationships absent in scene graphs.

Some studies have also explored leveraging scene graphs to construct negative samples, particularly focusing on token swapping, such as swapping asymmetric relations [15, 10, 5]. These methods have produced limited variants. However, our approach addresses the entire semantic representation rather than specific token swaps. To analyze the difference between outputs, we present the generated negative samples from Random Token Swap, Scene Graph Node Swap, and AMR Reconstruction in Table 3.

In contrast to Random Token Swap approach, leveraging scene graphs yields a richer array of syntactic and semantic cues. However, the generated negatives adhere to rule-based criteria, such as swapping exclusively between adjective words or words sharing a common relational structure. It is evident that AMR Reconstruction

Table 4
ARO performance comparison of different strategies. †: results from [10], applying semantic negative strategy; ‡: results from [15], incorporating Scene Graph Prediction in training.

	Visual Gnome		Flickr30k	COCO
	Relation	Attribution	Order	Order
CLIP	59.9	63.2	59.5	46.0
NegCLIP	81.0	71.0	91.0	86.0
AMR-NegCLIP	83.2	75.6	93.9	91.6
Semantic Negative†	79.0	77.8	-	-
CLIP-SGVL‡	-	-	82.0	78.2

introduces a wider spectrum of variations to the original captions, all while upholding the core semantic components. Our methodology thus offers enhanced flexibility in generating negative training data.

Furthermore, we compare AMR-NegCLIP with other negative augmentation-based methods, Semantic Negative [10], which constructs negative samples using scene graph node swaps, and CLIP-SGVL [15], which utilizes scene graphs in multiple ways, including positive and negative caption generation, as well as scene graph prediction tasks, in Table 4. However, the training and validation data sets of Semantic Negative are different from ours, but it can also be seen that it is challenging to improve the accuracy of both relationships and attributes by changing the negative samples. The findings indicate that AMR-NegCLIP achieves superior average performance in comparison to the Semantic Negative method. This observation underscores the efficacy of employing AMR generated negatives, which manifest more pronounced enhancements when compared to the strategy of swapping scene graph nodes. Negative sample generation rules in CLIP-SGVL are similar to those of Semantic Negative. Our AMR-NegCLIP demonstrated superior performance in Order tasks with more variants.

5.2. Case Study

We present several case studies illustrating the results of CLIP and AMR-NegCLIP across four subtasks in SugarCrepe, as depicted in Figure 4. SugarCrepe utilizes large language models to generate captions with a high degree of fluency and commonsense understanding, thereby posing a challenge for VLMs to discern negative captions effectively. For instance, in Swap Object task, VLMs must comprehend the semantics of relationships such as "in" and "background", as well as discern the object and subject of these relationships. Our test results demonstrate that while CLIP exhibits closely aligned similarity scores between captions and negative captions, AMR-NegCLIP demonstrates superior discriminatory capability. Furthermore, in Swap Attribution task, models are required to accurately identify quantities and the position of corresponding objects to succeed. CLIP returns nearly identical scores and struggles to differentiate between captions, whereas AMR-NegCLIP excels in selecting the correct option. Examples of Replace Relationship and Replace Attribution tasks highlight instances where CLIP struggles to discern subtle yet crucial concept replacements. These nuances have been effectively addressed through negative caption contrastive learning.

Table 5
Comparison of ARO performance before and after replacing a portion of original negative samples with AMR generated negative samples.

	Visual Gnome		Flickr30k	COCO	Average
	Relation	Attribution	Order	Order	
CLIP	59.9	63.2	59.5	46.0	57.2
NegCLIP	81.0	71.0	91.0	86.0	82.3
Replace Ratio					
10%	83.4	74.4	94.1	92.1	86.0
20%	82.6	76.0	92.9	90.3	85.4
30%	83.2	75.6	93.9	91.6	86.1
40%	83.8	74.8	91.3	88.3	84.5
50%	82.6	74.3	94.0	90.6	85.4
60%	81.2	75.1	91.5	87.6	83.9
70%	80.3	71.9	93.7	91.8	84.4
80%	80.2	71.2	93.2	91.5	84.0
90%	78.4	71.3	89.3	86.4	81.4
100%	75.0	69.4	83.4	80.9	77.2

5.3. Performance Impact Analysis of AMR Generated Negative Sample Ratios

AMR generated negative samples tend to distort entire semantic representations of given captions, while NegCLIP swaps the positions of tokens. Their generated negative samples address varying levels, from individual objects to complete semantics. To ensure augmented data spans different levels in the training dataset, we retain parts of negative samples from NegCLIP while replacing a ratio of NegCLIP samples with AMR generated negative samples.

To assess the impact of AMR generated negative samples on model performance, we replace NegCLIP negatives at ratios ranging from 10% to 100%, and present the results in Table 5. When replacing only 10% of NegCLIP negatives with AMR generated negative samples, the model performance exhibits noticeable improvements, particularly 6.1% in COCO Order subtasks. The best performance is achieved when 30% of the token swap negatives are replaced by AMR-generated negatives. Across replacement ratios ranging from 10% to 60%, the integration of AMR generated negatives yields improvements for NegCLIP across all subtasks. These enhancements are consistently observed, with average performance gains ranging from 1.6% to 3.8%. Beyond a 70% replacement ratio, larger ratios result in decreased model performance. Specifically, when 90% and 100% of negative samples are

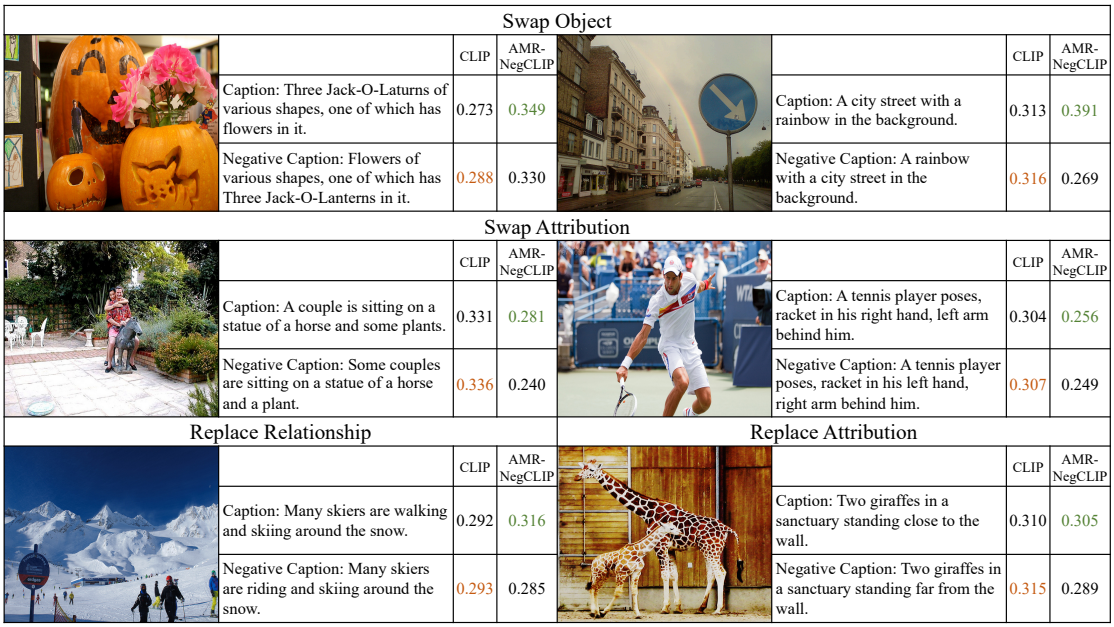

Figure 4: Predictions of CLIP and AMR-NegCLIP on SugarCrepe tasks: Swap Object, Swap Attribution, Replace Relationship and Replace Attribution. The score represents the similarity score between the (Negative) caption and the corresponding image as assessed by CLIP/AMR-NegCLIP. The model selects the caption with the higher similarity score as the correct one.

AMR generated, the performance is inferior to that of token swap negatives but still superior to CLIP. The reason for this phenomenon could be attributed to the greater diversity of AMR generated negatives compared to token swap negatives. Unlike token swap negatives, which follow a unified pattern, AMR generated negatives lack such consistency, making it challenging for models to effectively learn from them, particularly when the replacement ratio is high. Therefore, we propose that our AMR generated negative captions can effectively complement token swap generations.

6. Conclusion

To overcome the limitations of vision language models in comprehending composition and semantics, we suggest constructing hard negative samples through splitting and reconstructing AMR graphs. Compared to token and scene graph negative generation, AMR generated negatives have greater diversity and keep the fluency at the most possible. Compared to token and scene graph negative generation, AMR generated negatives exhibit greater diversity while maintaining optimal fluency. Our experimental results illustrate that incorporating our generated negatives in contrastive learning significantly boosts model performance, particularly in tasks that demand high-level comprehension. Furthermore, beyond simple shuffling, AMR offers the potential for more controlled modifications based on human instructions. For instance, users could add semantic components that are absent in the picture to deliberately confuse VLMs. We view this as a promising avenue for future research.

Limitaions Conducting AMR parsing and generation typically requires GPU acceleration, which incurs higher costs compared to direct token shuffling methods. However, when compared to tasks such as scene graph parsing or querying large language models, it remains an efficient approach. It's worth noting that splitting and shuffling AMR components introduce significant randomness in negative generation, and occasionally, this may lead to suboptimal results.

References

[1] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al., Learning transferable visual models from natural language supervision, in: International conference on machine learning, PMLR, 2021, pp. 8748–8763.

[2] J. Li, D. Li, C. Xiong, S. Hoi, Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation, in: International Conference on Machine Learning, PMLR, 2022, pp. 12888–12900.

[3] T. Zhao, T. Zhang, M. Zhu, H. Shen, K. Lee, X. Lu, J. Yin, Vl-checklist: Evaluating pre-trained vision-language models with objects, attributes and relations, arXiv preprint arXiv:2207.00221 (2022).

[4] T. Thrush, R. Jiang, M. Bartolo, A. Singh, A. Williams, D. Kiela, C. Ross, Winoground: Probing vision and language models for visio-linguistic compositionality, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 5238–5248.

[5] M. Yuksekgonul, F. Bianchi, P. Kalluri, D. Jurafsky, J. Zou, When and why vision-language models behave like bags-of-words, and what to do about it?, in: The Eleventh International Conference on Learning Representations, 2022.

[6] C.-Y. Hsieh, J. Zhang, Z. Ma, A. Kembhavi, R. Krishna, Sugarcrepe: Fixing hackable benchmarks for vision-language compositionality, Advances in Neural Information Processing Systems 36 (2024).

[7] L. Banarescu, C. Bonial, S. Cai, M. Georgescu, K. Griffitt, U. Hermjakob, K. Knight, P. Koehn, M. Palmer, N. Schneider, Abstract meaning representation for sembanking, in: Proceedings of the 7th linguistic annotation workshop and interoperability with discourse, 2013, pp. 178–186.

[8] Z. Shou, Y. Jiang, F. Lin, Amr-da: data augmentation by abstract meaning representation, in: Findings of the Association for Computational Linguistics: ACL 2022, 2022, pp. 3082–3098.

[9] S. Doveh, A. Arbelle, S. Harary, E. Schwartz, R. Herzig, R. Giryes, R. Feris, R. Panda, S. Ullman, L. Karlinsky, Teaching structured vision & language concepts to vision & language models, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 2657–2668.

[10] Y. Huang, J. Tang, Z. Chen, R. Zhang, X. Zhang, W. Chen, Z. Zhao, T. Lv, Z. Hu, W. Zhang, Structure-clip: Enhance multi-modal language representations with structure knowledge, arXiv preprint arXiv:2305.06152 (2023).

[11] M. A. Abdelsalam, Z. Shi, F. Fancellu, K. Basioti, D. J. Bhatt, V. Pavlovic, A. Fazly, Visual semantic parsing: From images to abstract meaning representation, arXiv preprint arXiv:2210.14862 (2022).

[12] M. Bevilacqua, R. Blloshmi, R. Navigli, One spring to rule them both: Symmetric amr semantic parsing and generation without a complex pipeline, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, 2021, pp. 12564–12573.

[13] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer, Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 7871–7880.

[14] L. F. Ribeiro, M. Schmitt, H. Schütze, I. Gurevych, Investigating pretrained language models for graph-to-text generation, in: Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI, 2021, pp. 211–227.

[15] R. Herzig, A. Mendelson, L. Karlinsky, A. Arbelle, R. Feris, T. Darrell, A. Globerson, Incorporating structured representations into pretrained vision & language models using scene graphs, arXiv preprint arXiv:2305.06343 (2023).