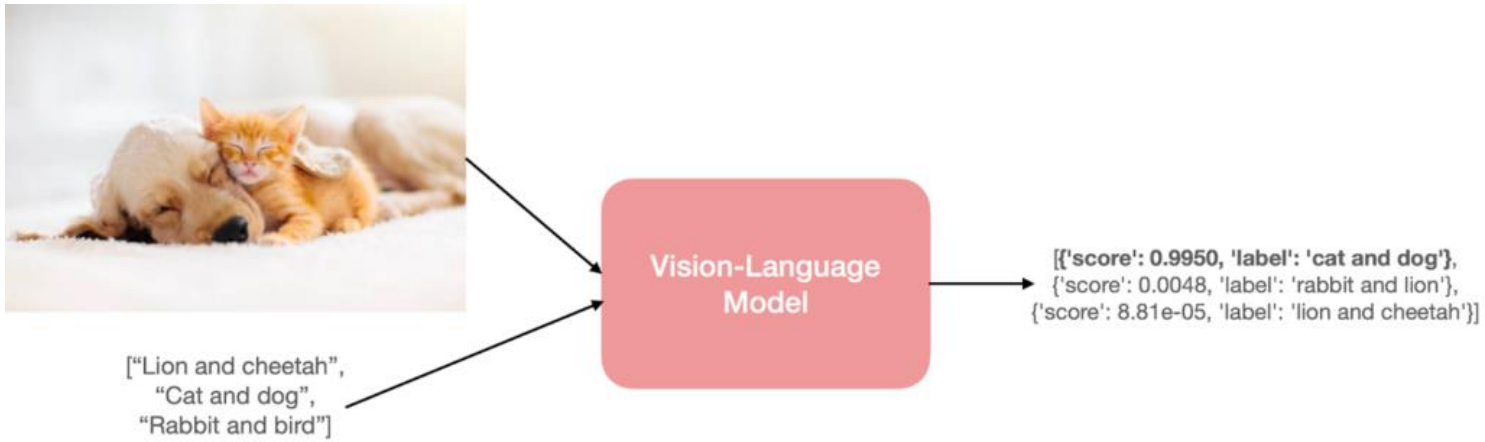
Seeing what's not there: Negation understanding needs more than training

Honda R&D

Bhuvan Aggarwal

Amit More, Mudit Soni, S Divakar Bhat

What are VLMs?



Vision Language Models (VLMs) like CLIP, BLIP, Grounding DINO bridge the gap between visual perception and natural language understanding.

Why are they exciting?

- Reasoning about visual scenes
- Image Generation and Multi-modal Retrieval
- Promptable Perception and Region Selection

VLMs fail to understand Negation

A text-to-image and grounded detection example using various VLMs:

A person crossing the road **not** being fully visible.

CLIP

*NegCLIP**

Desired Result

A motorbike rider **without** a helmet

CLIP

*NegCLIP**

Desired Result

Baseline CLIP MCQ accuracy for negations: 30% (near random chance!)

VLMs fail to understand Negation

But why?

- **Under-representation in data:** Only ~1% of training data contains negations.
- **Bag-of-Words understanding:** Due to the contrastive pretraining approach, these models treat the text as a bag-of-words without any compositional understanding.
- **Affirmation Bias:** Models learn to ignore words like 'no', 'not', 'without' etc.

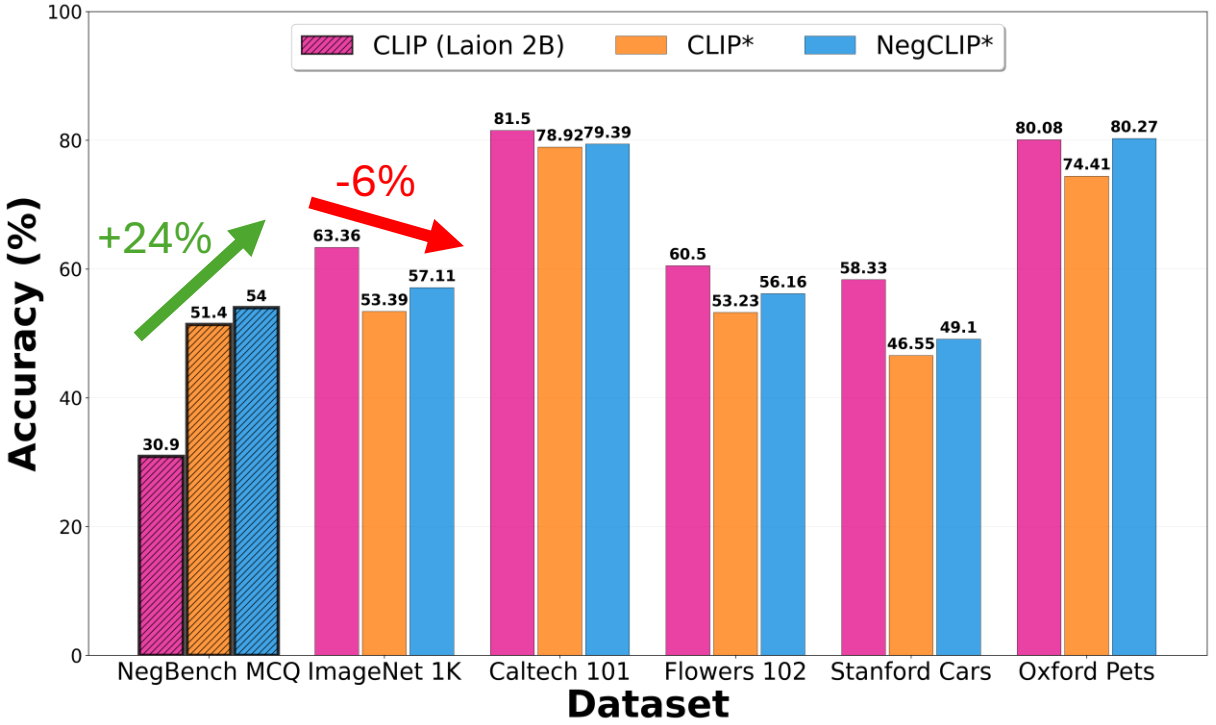
Result 😢

”A car without tires” generates same image as ”A car with tires”

Previous approaches: The Data-Centric Path!

Common Solution: More Training Data!

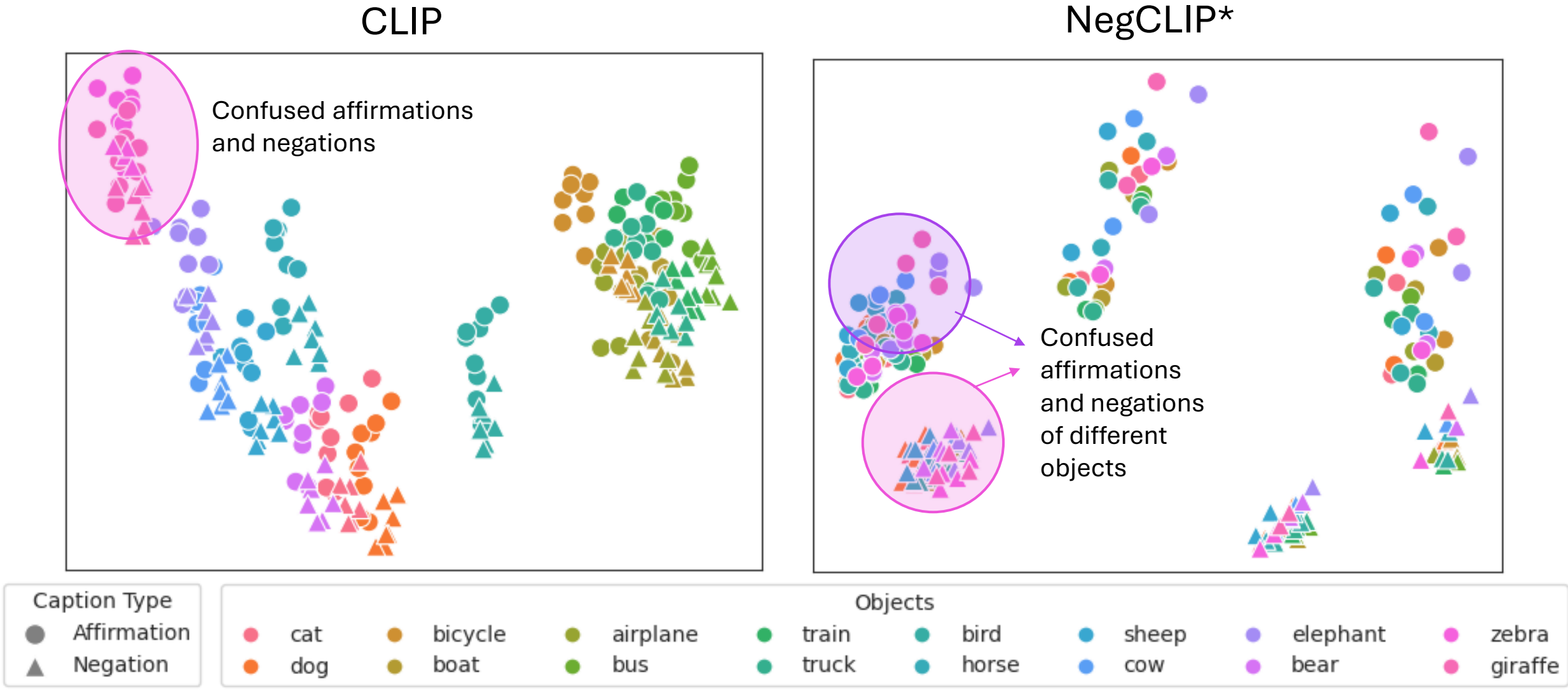
- Create datasets with hard-negative samples
- Fine-tune models on millions of image-text pairs
- Examples: CC-Neg, CC12M-NegFull, TripletCLIP



Problems with this approach

- **Catastrophic Forgetting**: Performance drops on general tasks
- **Expensive Training:** Requires massive computational resources
- **Still Incomplete:** Fine-tuned models still struggle with negations

Previous approaches: The Data-Centric Path!



(*) represents models trained on CC12M-NegFull

So the key question is:

Do we need more training or just

better embeddings?

Can we fix negation understanding without retraining?

Our Approach

Our approach is inspired by the **vector-offset method** in NLP, so we explore semantics directly in CLIP's embedding space.

GLoVe and Word2Vec embeddings follow compositional Arithmetic!
E(King) - E(Man) + E(Woman) ~ E(Queen)

Addition works:
E(cat) + E(flower) ~ E(cat and flower)

Subtraction works too:
E(Paris) - E(France) + E(China) ~ E(Beijing)

So, in CLIP also, can we use direct subtraction to remove the negated concepts from captions? **Not really**

The main reason for this is that CLIP embeddings are entangled and not fully compositional so we cannot use direct arithmetic in the CLIP semantic space!
BUT.....

Our Approach

Instead of directly following compositional arithmetic:

The naïve formulation:
$e^* = e^c - e^{neg}$

- **Compute directional offset:** The projection of caption on the negated concept tells us what needs to be removed.
- **Anchor correction:** To maintain the non-informative bias of the model for correct positioning of the embeddings.

$$e^* = e^c - \lambda \left(\frac{\langle e^c, e^{neg} \rangle}{\langle e^{neg}, e^{neg} \rangle} e^{neg} - a \right)$$

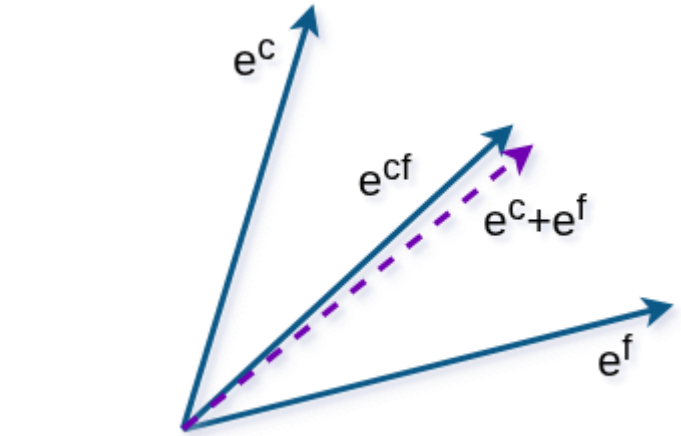
Original Confused Embedding

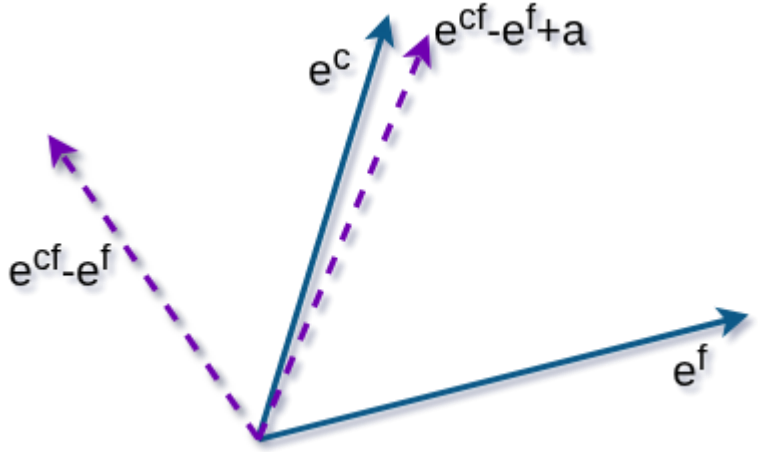
Directional Offset of the Negated Concept

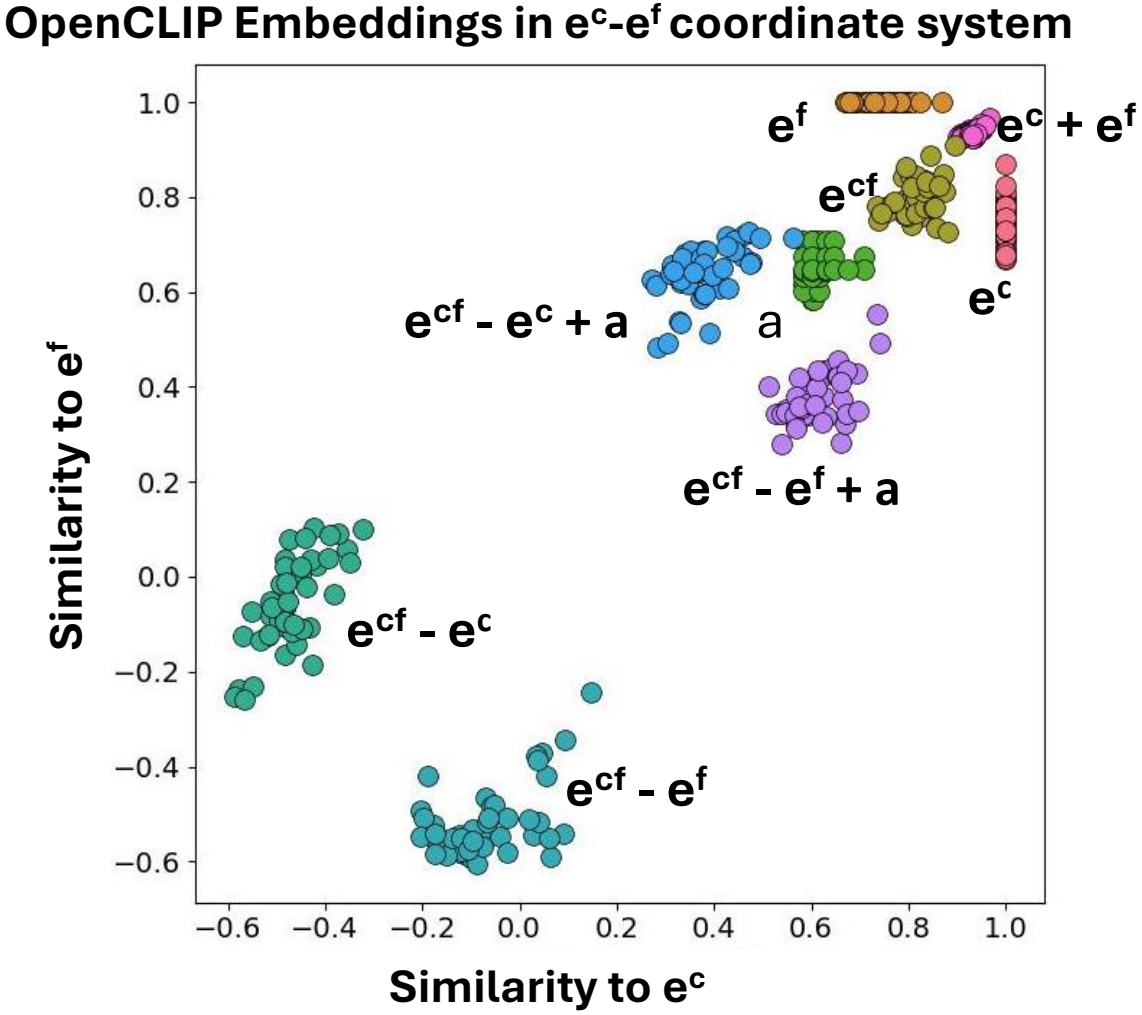
Semantic Anchor

Correction Strength

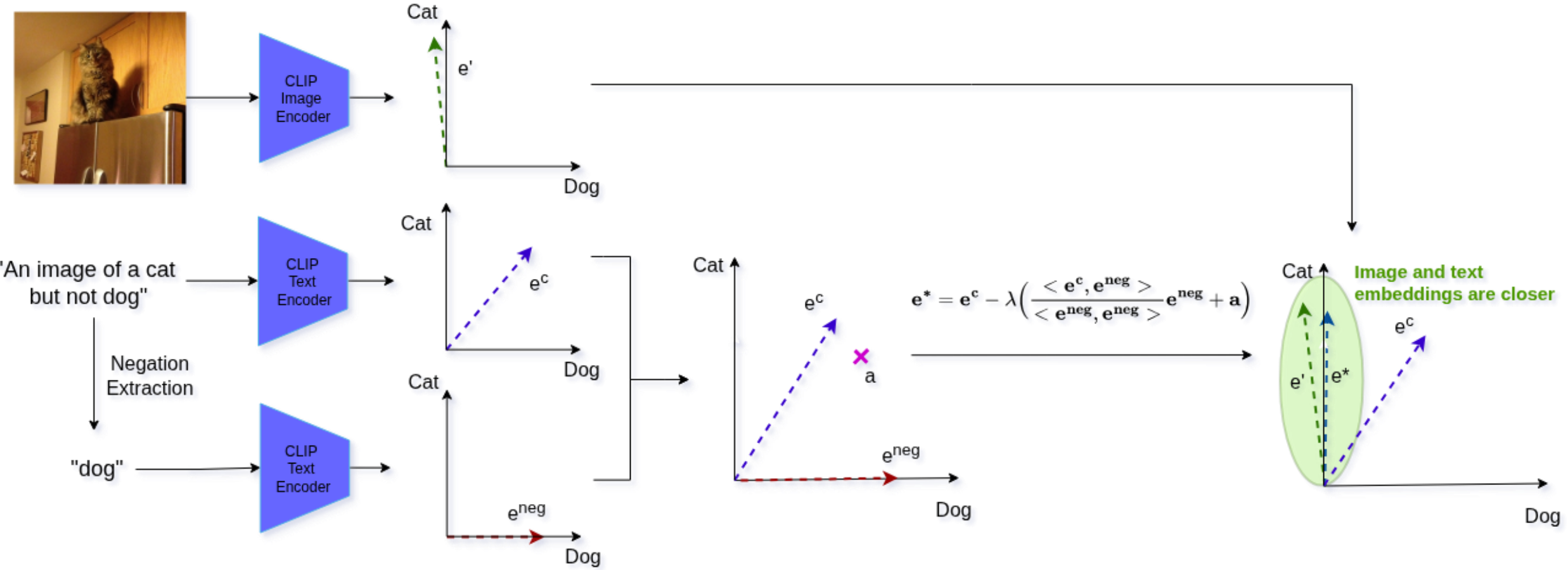
Our Approach







Our Approach



Results

Our method outperforms trained models results (56.2/59.7/46.2/67.0/51.5) by **(+13.3,+15.4,+7.9,+0.9,+0.7)** even without any model training.

On many different backbones, we see similar trend and results are better than trained models

	MCQ			Retrieval			
	COCO	VOC2007	MSRvtt	COCO		MSRvtt	
Model				**R@5**	**R-Neg@5**	**R@5**	**R-Neg@5**
SigLIP	28.9	30.8	30.7	72.1	64.4	51.4	44.7
+ Ours	**61.15** (↑32.25)	**66.9** (↑36.1)	**46.8** (↑16.1)	–	**68.7** (↑4.2)	–	**47.6** (↑2.9)
SigLIP2	27.2	27.1	30.1	72.8	64.0	51.6	45.3
+ Ours	**66.3** (↑39.1)	**70.2** (↑43.1)	**50.1** (↑20)	–	**70.2** (↑6.2)	–	**49.4** (↑4.1)
AlignCLIP	32.7	28.4	23.6	44.8	35.6	35.8	31.1
+ Ours	**60.1** (↑27.4)	**69.2** (↑40.8)	**44.3** (↑20.7)	–	**41.2** (↑5.6)	–	**34.7** (↑3.6)
TripletCLIP	33.8	23.7	30.2	52.3	44.3	42.4	35.8
+ Ours	**61.8** (↑28.0)	**57.2** (↑33.5)	**46.7** (↑16.5)	–	**47.9** (↑3.6)	–	**39.9** (↑4.1)

Table 2: Performance evaluation on other backbones show that our approach generalizes across different model architectures and training paradigm.

	MCQ			Retrieval			
	COCO	VOC2007	MSRvtt	COCO		MSRvtt	
Model				**R@5**	**R-Neg@5**	**R@5**	**R-Neg@5**
CLIP openai	39.3	38.7	32.1	54.8	48.6	50.6	45.8
CLIP laion400M	31.0	31.6	30.0	59.4	51.7	43.1	37.5
NegCLIP	28.7	30.5	27.3	68.7	64.4	53.7	51.0
CLIP*	54.4	54.8	44.9	54.2	51.9	46.9	43.9
NegationCLIP	36.0	44.1	34.5	62.2	61.4	43.0	41.6
CLIP	24.7	24.3	27.5	64.8	57.3	49.7	44.5
Ours + CLIP	**72.5** (↑47.8)	**78.6** (↑54.3)	**50.0** (↑22.5)	-	63.2 (↑5.9)	-	49.0 (↑5.5)
NegCLIP*	56.2	59.7	46.2	69.0	67.0	54.0	51.5
Ours+NegCLIP	69.5 (↑13.3)	75.1 (↑15.4)	54.1 (↑7.9)	-	**67.9** (↑0.9)	-	**52.2** (↑0.7)

Table 1: **Performance evaluation on NegBench.** We evaluate our embedding correction method against baseline and fine-tuned CLIP models across Multiple Choice Questions (MCQ) and Retrieval tasks. (*) indicates the models fine-tuned on the CC12M-NegFull. We show improvements over original models in brackets.

Results

Model	ConCLIP	NegCLIP*	CLIP	CLIP+Ours
Acc.	57.2	78.7	89.8	**10.2**

Table 3. Zero-shot accuracy on CIFAR10 with distractor captions.

Positive Captions:

This image depicts a truck. (Ground Truth)

“This image depicts a {cls}.” for all CIFAR-10 classes

Negative Caption: (distractor)

This image does not depicts a truck.

“This image does not depicts a {cls}.” for all CIFAR-10 classes

CLIP	NegCLIP*	CoNCLIP	Ours+CLIP
89.5	86.8	83.8	89.5
89.8	78.7	57.2	10.2
✖	✖	✖	✅

Same as Random chance (10%): shows good understanding of negations.

Results

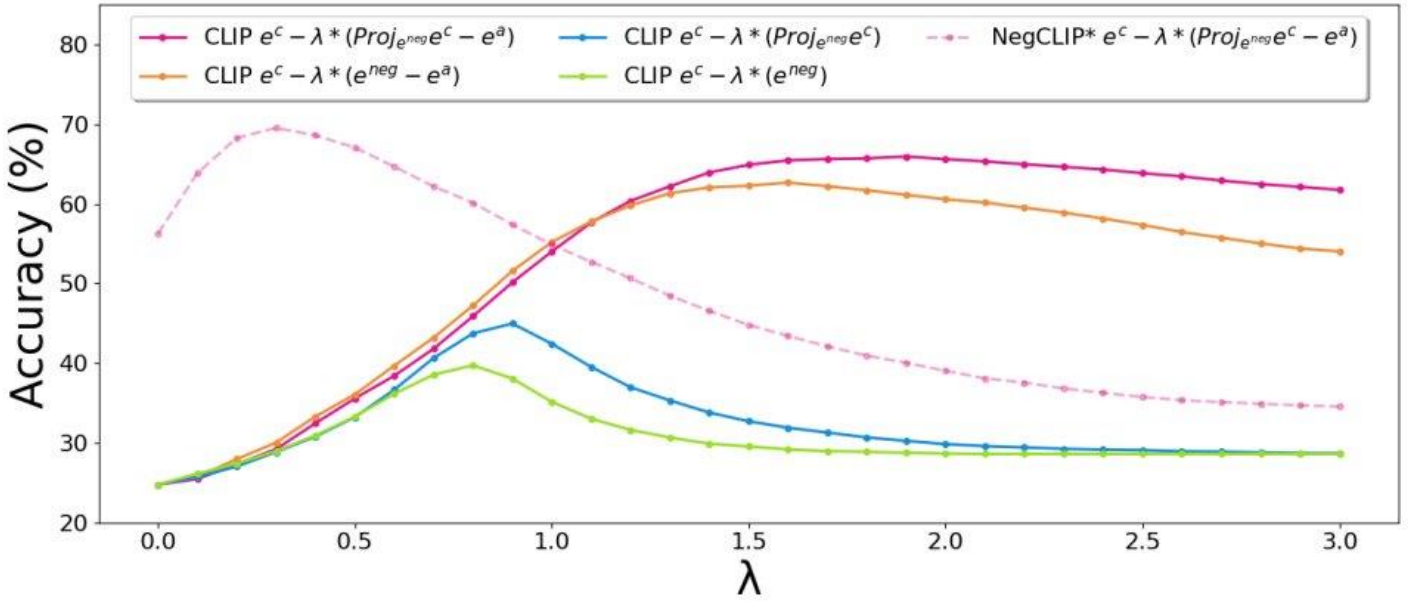
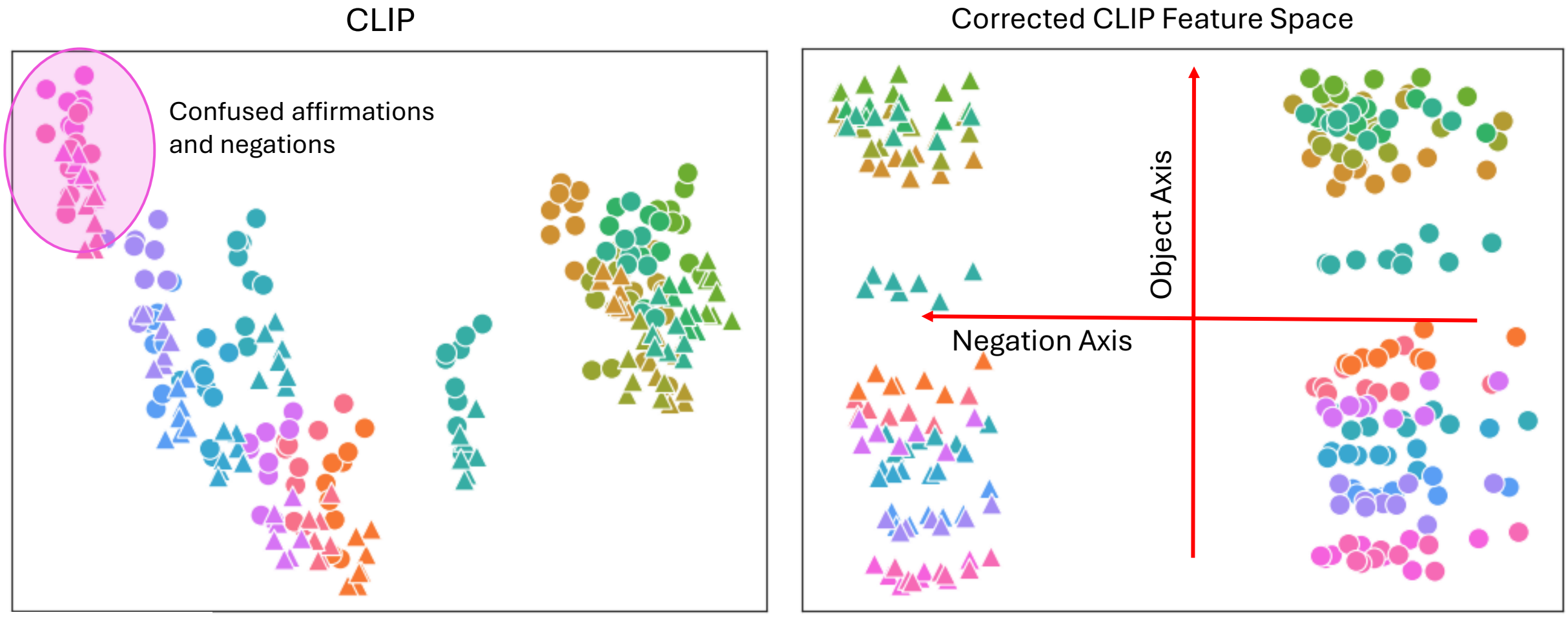
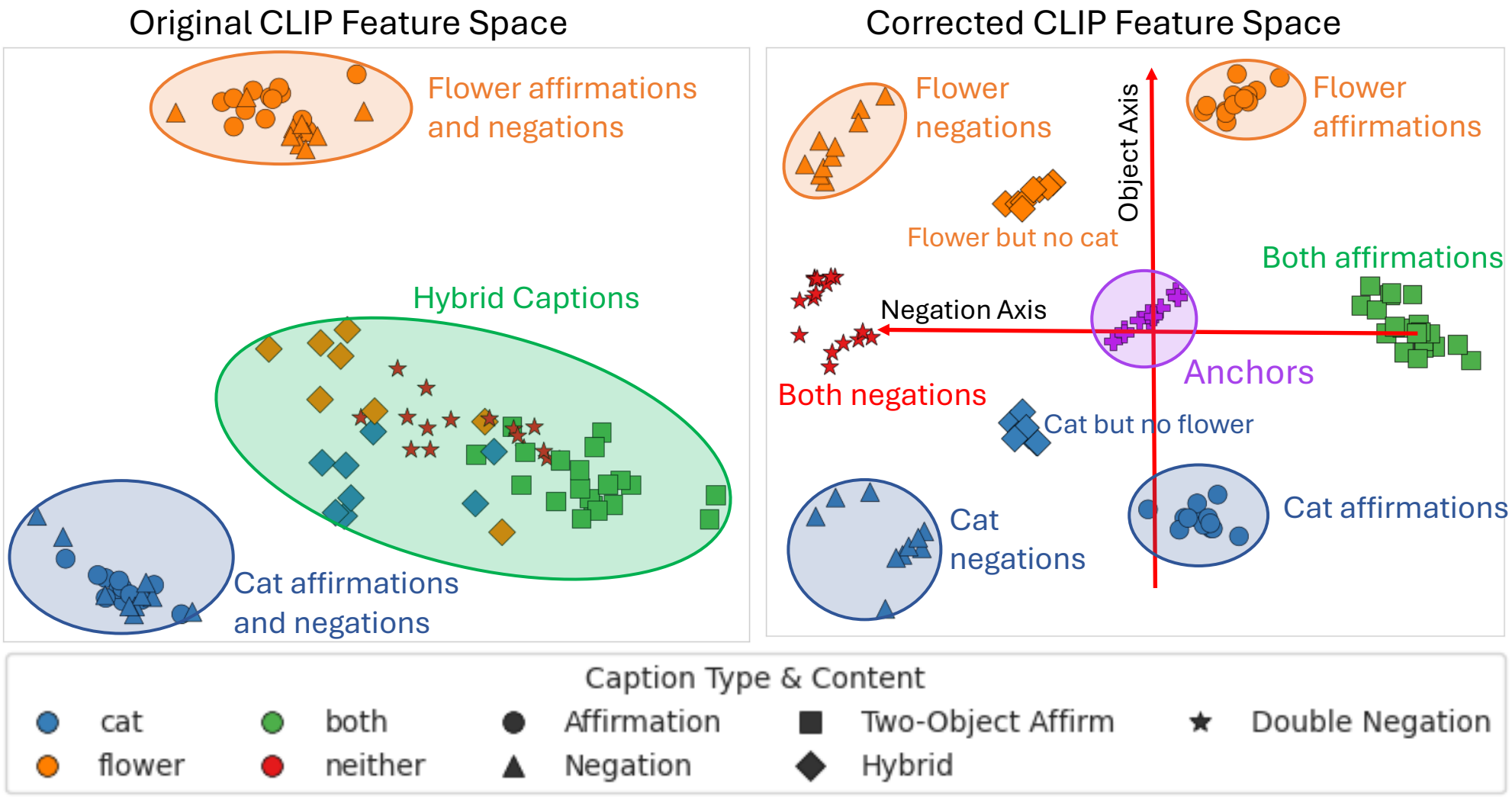


Figure 7. We analyze the performance for CLIP model different values of λ. Projection of $\mathbf{e^c}$ on $\mathbf{e^{neg}}$ is denoted by $\mathbf{Proj_{e^{neg}}}$ Results for NegCLIP* are shown in dashed line.

Results


CLIP
Confused affirmations and negations
Corrected CLIP Feature Space
Object Axis
Negation Axis
Caption Type
Affirmation
Negation
Objects
cat
dog
bicycle
boat
airplane
bus
train
truck
bird
horse
sheep
cow
elephant
bear
zebra
giraffe

Results


Original CLIP Feature Space
Flower affirmations and negations
Hybrid Captions
Cat affirmations and negations
Corrected CLIP Feature Space
Flower negations
Flower affirmations
Object Axis
Flower but no cat
Both affirmations
Negation Axis
Anchors
Both negations
Cat but no flower
Cat negations
Cat affirmations
Caption Type & Content
cat
flower
both
neither
Affirmation
Negation
Two-Object Affirm
Hybrid
Double Negation

Qualitative examples

Image	Ground Truth Caption	Caption selected by Neg-CLIP*	Caption selected by our alg.
	A cup is nowhere to be found in this image.	A cup is present in this image, but there is no pizza. ✗	A cup is nowhere to be found in this image. ✓
	A dining table is not present in this image.	There is no hot dog in this image. ✗	A dining table is not present in this image. ✓

Limitations and Future work

Syntactic Brittleness:
Current rule-based parsers struggle with complex sentence structures like “A large, decorative, but clearly not present, table.”
Implicit Negations:
Rule-based parsers also struggle with implicit negations

Fix: LLM Integration
Future work will replace the rule-based parser with a lightweight LLM.

Conclusion

- VLMs are a foundational part of current end-to-end pipelines but lack fundamental negation understanding.
- All the approaches till now have tried to introduce negation understanding through finetuning on additional negation datasets which leads to improved performance on benchmarks while still having incomplete negation understanding.
- We introduce a zero-shot embedding arithmetic-based correction to improve the understanding of VLMs showing both qualitative and quantitative improvements
- The current bottleneck of our negation extraction approach and the benchmarks is the lack of implicit negations.