What CLIP Knows but Cannot Say: Recovering Negation from Frozen Intermediate Features

Chen-Yi Lu, Yueh-Shao Chen, and Somali Chaterji

Purdue University

Abstract. Contrastive vision-language models such as CLIP map semantically opposite phrases (*e.g.*, "a dog" *vs.* "not a dog") to nearly identical embeddings, rendering them insensitive to negation. We attribute this failure to a phenomenon we call *Representational Collapse*: by tracking compositional divergence and visual alignment across the CLIP text encoder, we show that middle layers build compositional syntax, but the final layers collapse this structure as visual alignment rises, producing a syntax-blind final representation. To recover the lost negation signal without altering pretrained weights, we propose PeakPatch, a lightweight post-hoc correction system that intercepts the encoder at its *compositional peak* while keeping CLIP fully frozen. An Embedding Correction Network (ECN) uses cross-attention to extract a negation-specific signal from the peak layer, anchored to a stable baseline, and predicts a deviation vector that re-injects the lost syntax into the final-layer embedding space. A complementary Score Correction Network (SCN) predicts bounded scalar score offsets for discriminative tasks. Both modules are trained jointly end-to-end while all CLIP parameters remain frozen, adding only 5.2M parameters (3.5% of the backbone) and preserving the standard cosine similarity interface. On NegBench, PeakPatch achieves 74.3% on COCO MCQ (+35.1 over CLIP, +17.8 over the best encoder fine-tuning method) and 65.5% on VOC MCQ, while outperforming all fine-tuning baselines on fully out-of-distribution negation retrieval despite training only 3.5% of the parameters. The corrected embeddings also transfer to text-to-image generation (+18.4 negation score) and generalize across ViT-B/32, ViT-L/14, and SigLIP backbones.
Project page: `https://stevencylu.github.io/PeakPatch/`

1 Introduction

Vision-language models (VLMs) such as CLIP [35] learn a shared embedding space where images and text can be directly compared via cosine similarity. This simple yet powerful formulation has enabled a broad range of applications, including zero-shot classification [35, 57], cross-modal retrieval [30, 52], text-to-image generation [36,38], open-vocabulary detection [19,24,27,50], and semantic segmentation [29, 48]. Subsequent models such as ALIGN [17], OpenCLIP [5], SigLIP [53], and EVA-CLIP [41] have scaled this paradigm to larger datasets and architectures.

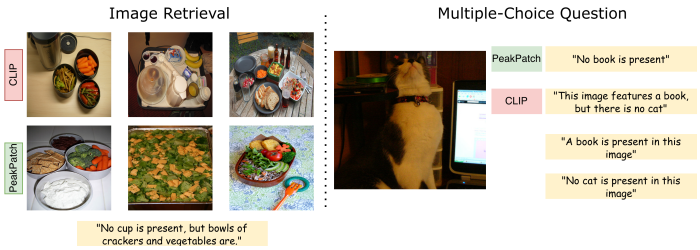

Fig. 1: CLIP's negation blindness on two NegBench protocols. (*Left*) Retrieval: CLIP retrieves images containing the negated object (cup), treating "No cup is present" as similar to "cup present." PeakPatch correctly retrieves cup-free images. (*Right*) MCQ: given a cat image with no book, CLIP selects a caption mentioning both nouns while ignoring negation; PeakPatch selects the correct answer ("No book is visible in this image.").

Despite this progress, these models share a fundamental limitation: they cannot distinguish negated descriptions from affirmative ones. Semantically opposite phrases (*e.g.*, "a dog" *vs.* "not a dog") are mapped to nearly identical embeddings, making the model effectively blind to negation. As Fig. 1 illustrates, this causes CLIP to retrieve images containing the negated object and to select wrong captions in multiple-choice settings. On the NegBench VOC MCQ task [3], this manifests as a universal *affirmation bias*: 81% accuracy on affirmative captions collapses to just 3% on negated ones, worse than random chance. This limitation has practical consequences: applications routinely require negation, from a radiologist searching for "bilateral consolidation with no evidence of pneumonia" to a safety inspector querying "construction sites with no barriers" to content moderation prompts such as "no weapon."

Why does this happen, and can it be fixed? Kang *et al.* [20] prove that the problem is fundamental: no CLIP-like joint embedding space can correctly handle even *any two* of basic semantics, attribute binding, spatial relations, and negation, as the geometry is overconstrained. Fine-tuning methods [33,40,52] improve negation accuracy without large drops in zero-shot classification, but they remain subject to this geometric ceiling—they push negation into a space that provably cannot fully accommodate it. Moreover, fine-tuning permanently alters the pretrained weights, requiring every downstream system that builds on frozen CLIP features (*e.g.*, LLaVA [26], BLIP-2 [22], text-to-image generators [38]) to be re-validated or re-adapted. We therefore ask: *Can we correct CLIP's negation blindness without modifying the encoder or relying on an external model?*

To answer this, we probe the frozen CLIP text encoder, layer by layer. We introduce the Layer-wise Compositional Divergence (LCD) metric, which tracks how well each layer separates negated captions from their affirmative counterparts (Fig. 2). Jointly measuring this compositional divergence alongside per-layer visual alignment to the paired image reveals a clear trajectory. LCD rises through early and middle layers as the encoder builds compositional structure, then peaks at a *compositional peak* layer l_p. Visual alignment, meanwhile, re-

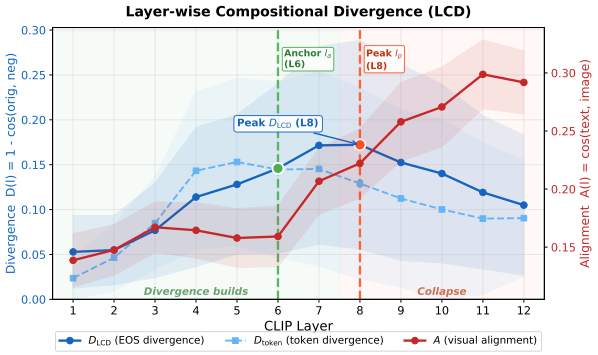

Fig. 2: Representational Collapse in CLIP's text encoder. We track two layer-wise signals: compositional divergence (LCD; blue, left axis), defined as $D(\ell) = 1 - \cos(\text{orig}, \text{neg})$, and visual alignment (red, right axis), defined as $A(\ell) = \cos(\text{text}, \text{image})$. Divergence peaks at Layer l_p but drops sharply as alignment rises, collapsing into a syntax-blind representation. PeakPatch extracts negation features at l_p (compositional peak) and anchors them to l_a (L6) to correct the collapsed final-layer output.

mains low through an anchor layer l_a and then rises sharply. Beyond l_p, visual alignment dominates while LCD drops. The final layers collapse the compositional structure into a syntax-blind representation optimized for cross-modal matching. We call this phenomenon *Representational Collapse*.

Representational Collapse suggests a simple strategy: extract negation information before the encoder discards it, then inject it back into the standard final embedding interface. Since the final layer is strongly visually aligned but syntax-blind, while the compositional peak is syntax-aware but not yet visually aligned, neither representation suffices alone. Based on this insight, we propose PeakPatch, a lightweight post-hoc correction system that intercepts the frozen CLIP encoder at its compositional peak and corrects the collapsed final-layer output. An Embedding Correction Network (ECN) uses cross-attention to extract a negation-specific signal from Layer l_p, anchors it to a stable baseline at Layer l_a, and predicts a deviation vector that injects the lost syntax into the final-layer embedding space. A Score Correction Network (SCN) contrasts the peak-layer representation with the ECN-corrected embedding and the image to predict bounded scalar corrections for discriminative tasks. Both modules are trained jointly end-to-end while all CLIP parameters remain frozen, adding 5.2M parameters (3.5% of the backbone) and preserving the standard cosine similarity interface. Overall, we correct CLIP's negation blindness entirely post-hoc, with a frozen encoder and no external inference-time model. Our contributions are as follows:

1. We introduce the Layer-wise Compositional Divergence (LCD) metric and use it to identify *Representational Collapse* in CLIP's text encoder: negation separability peaks at an intermediate layer and then degrades as visual align-

ment increases, revealing that the encoder actively discards compositional structure in its final layers.
2. Motivated by this finding, we propose PeakPatch, a lightweight post-hoc correction system that keeps all of CLIP frozen and requires no external model. An Embedding Correction Network (ECN) extracts negation signals from the compositional peak via cross-attention and corrects the final text embedding, while a Score Correction Network (SCN) applies bounded corrections to the similarity score.
3. PeakPatch achieves 74.3% on COCO MCQ (+35.1 over CLIP, +17.8 over the best encoder fine-tuning method) with only 5.2M parameters (3.5% of the backbone). In the fully OOD retrieval setting, it outperforms all fine-tuning baselines despite modifying only the text side, and its corrected embeddings transfer to text-to-image generation (+18.4 negation score).

2 Related Work

2.1 Post-Hoc Adaptation of Vision-Language Models

Contrastive VLMs such as CLIP [35], ALIGN [17], OpenCLIP [5], SigLIP [53], and FLAVA [39] enable strong zero-shot transfer, but adapting them without damaging pretrained representations remains an open challenge. Existing approaches operate at three levels of the inference pipeline. *Prompt learning* [18, 21, 56, 57] optimizes continuous input tokens while keeping weights frozen, yielding gains on few-shot classification but leaving the similarity function unchanged. *Feature adapters* [9, 45, 54] apply residual or cache-based transformations to output embeddings, demonstrating that small feature-space perturbations can produce large task-level improvements. *Score- and classifier-level corrections* [16, 42, 51, 58] modify classifier weights or logits to steer predictions without retraining the backbone. Complementing these, WiSE-FT [47] shows that interpolating fine-tuned and zero-shot *model weights* preserves robustness, and parameter-efficient methods such as low-rank adaptation [14] have become standard for updating large models with minimal overhead. Notably, generative VLMs that use CLIP as a frozen visual encoder [2, 6, 22, 23, 26] inherit any compositional failure in CLIP's representations, making upstream corrections especially consequential.

Despite their diversity, these methods share a common assumption: the embedding geometry is fundamentally sound and merely needs task-specific steering. PeakPatch draws on the idea of residual corrections [9, 51] but addresses a qualitatively different problem: the text encoder itself produces systematically wrong representations for negated inputs, requiring correction at the representation level before any downstream scoring can be meaningful.

2.2 Negation Understanding in Vision-Language Models

Among the compositional failures exposed by recent benchmarks [7, 13, 15, 31, 44, 55], negation stands out as uniquely severe. While negation has been studied

extensively in natural language understanding [12], its impact on VLMs has only recently been quantified: NegBench [3] reveals that CLIP's 81% affirmative accuracy collapses to 3% on negated captions, and CC-Neg [40] confirms this failure at scale with 228K pairs. The root cause appears structural rather than data-driven: Kang *et al.* [20] prove a geometric impossibility theorem showing that no single embedding space can correctly handle even any two of basic semantics, attribute binding, spatial relations, and negation without overconstraining the geometry, while Quantmeyer *et al.* [34] provide mechanistic evidence that negation processing is distributed across layers and concentrated in a small fraction of negator-selective attention heads (8%).

Existing fixes fall into two camps, each with clear tradeoffs. *Encoder fine-tuning* methods [33, 40, 52] retrain the text encoder on negation-aware data. This improves negation understanding but risks degrading the broad representations that make CLIP useful, precisely because it forces negation into an embedding space that provably cannot accommodate it [20,47]. *Inference-time methods* avoid modifying CLIP but introduce other constraints: DCSM [20] replaces cosine similarity with token-to-patch CNN scoring, breaking the standard CLIP interface and preventing use in downstream pipelines that rely on embeddings. Two concurrent methods fall outside these camps but introduce their own constraints: SpaceVLM [37] and Aggarwal *et al.* [1] explicitly extract the negated concept at inference time, the former through an LLM parser and the latter through a rule-based parser, making them not directly comparable to methods that operate on the raw caption without external parsing. We provide detailed comparisons in the supplementary material.

2.3 Interpreting CLIP's Internal Representations

A growing body of work probes what CLIP's layers encode. Gandelsman *et al.* [8] project each ViT layer's output through CLIP's final projection head, decomposing the image representation into per-layer contributions. On the text side, CLIP behaves largely as a bag of words [52], and Quantmeyer *et al.* [34] use causal tracing to establish that negation awareness *does* exist in intermediate layers but spans only 8% of attention heads and does not survive to the final representation. These studies reveal *what* CLIP fails at and *where* information resides, but stop short of quantifying the layer-wise dynamics or translating diagnostics into a correction strategy.

Our LCD analysis (Sec. 3.2) builds on these findings. Adopting the projection methodology of Gandelsman *et al.* [8] on the text encoder side, we track the competition between compositional divergence and visual alignment across every layer, revealing a consistent collapse of negation separability in the final layers—a phenomenon we term *Representational Collapse*. This directly motivates PeakPatch: lightweight correction modules that read negation signals from intermediate layers where prior mechanistic analyses localize them [34], sidestep the impossibility theorem [20] by operating *outside* the joint embedding space, and apply corrections at both the embedding and score levels—all while keeping CLIP frozen.

3 Method

3.1 Preliminaries

CLIP encodes images and text into a shared d-dimensional space. Given an image I and text T, their similarity is computed as:

$$s(I,T) = \cos(\mathbf{f}_I, \mathbf{f}_T) = \frac{\mathbf{f}_I^\top \mathbf{f}_T}{\|\mathbf{f}_I\| \, \|\mathbf{f}_T\|}, \tag{1}$$

where $\mathbf{f}_I = E_I(I) \in \mathbb{R}^d$ and $\mathbf{f}_T = E_T(T) \in \mathbb{R}^d$ are the image and text embeddings, respectively. The text encoder is an L-layer transformer [46]; each layer applies multi-head self-attention and a feed-forward sub-network, producing hidden states $\mathbf{H}^l \in \mathbb{R}^{N \times d}$ at layer l, where N is the sequence length. The final text embedding is obtained by projecting the `[EOS]` token of the last layer through a layer norm and linear head into the shared space: $\mathbf{f}_T = W_T \, \mathrm{LN}(\mathbf{h}^L_{\texttt{[EOS]}})$, followed by ℓ_2 normalization. Following Gandelsman *et al.* [8], who project intermediate ViT layer outputs through CLIP's final head to interpret per-layer contributions, we extend this to the text encoder by defining $\mathbf{f}_T^l = W_T \, \mathrm{LN}(\mathbf{h}^l_{\texttt{[EOS]}})$ (with ℓ_2 normalization), so that $\mathbf{f}_T^L \equiv \mathbf{f}_T$. Because W_T was trained to maximize cosine similarity between $\mathbf{f}_T^L$ and $\mathbf{f}_I$, the score $\cos(\mathbf{f}_T^l, \mathbf{f}_I)$ at intermediate layers measures the degree to which layer l has already specialized for cross-modal alignment.

For a negated caption T^- (*e.g.*, "not a dog") and its affirmative counterpart T^+ (*e.g.*, "a dog"), CLIP produces $\cos(\mathbf{f}_{T^+}, \mathbf{f}_{T^-}) \approx 1$, meaning that the embeddings are nearly identical despite opposite semantics. Kang *et al.* [20] prove that this is a geometric constraint of the shared space, not merely a training failure. However, our analysis (Sec. 3.2) reveals that compositional awareness *does* exist in the frozen encoder's intermediate layers but is overwritten by the final layers.

3.2 Layer-wise Compositional Divergence

To formalize *Representational Collapse*, we define two complementary per-layer metrics:

Compositional divergence. Given a diagnostic set $\mathcal{D} = \{(I_i, T_i^+, T_i^-)\}$ of image–caption triples, where T^+ is the affirmative caption and T^- its negated counterpart, we measure how well each layer separates their projected `[EOS]` representations:

$$D_{\mathrm{LCD}}(l) = \frac{1}{|\mathcal{D}|} \sum_i \Big(1 - \cos\big(\mathbf{f}_T^l(T_i^+), \, \mathbf{f}_T^l(T_i^-)\big)\Big). \tag{2}$$

Visual alignment. We complement D_{LCD} with a visual alignment score that measures how closely each layer's projected `[EOS]` representation matches the paired image embedding:

$$A(l) = \frac{1}{|\mathcal{D}|} \sum_i \cos\big(\mathbf{f}_T^l(T_i^+), \, \mathbf{f}_{I_i}\big). \tag{3}$$

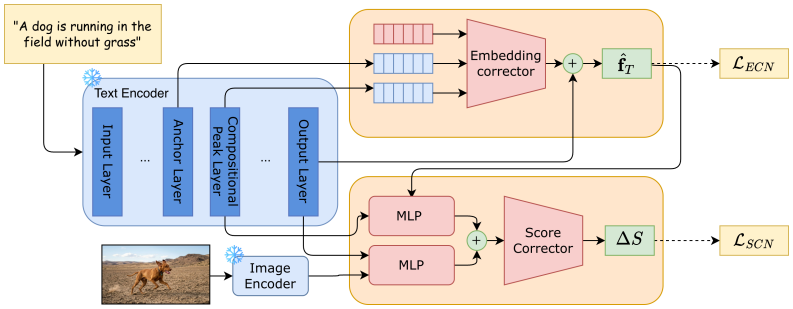

Fig. 3: Method overview. With all CLIP parameters frozen, we extract hidden states at three layers identified by the LCD analysis (Sec. 3.2): anchor l_a, compositional peak l_p, and final layer L. The ECN (Sec. 3.3) attends over the full token sequence at l_p with a learned query, fuses the result with peak- and anchor-layer embeddings, and predicts a deviation vector $\boldsymbol{\delta}$ added to the collapsed layer-L embedding. The SCN (Sec. 3.3) contrasts the corrected embedding $\hat{\mathbf{f}}_T$ with the l_p representation and the image embedding $\mathbf{f}_I$ to predict a bounded scalar correction Δs to the cosine similarity. Both modules are trained jointly (Sec. 3.4) while all CLIP parameters remain frozen.

Because both $\mathbf{f}_T^l$ and $\mathbf{f}_I$ lie in the shared embedding space, the cosine similarity is well-defined at every layer; $A(l)$ is not a claim about inherent visual content at layer l, but a measure of how strongly the contrastive alignment pressure has reshaped that layer's output.

The collapse trajectory. Figure 2 plots both metrics across the L transformer layers, revealing three phases. In the early and middle layers, D_{LCD} rises steadily as the encoder builds syntactic structure, while $A(l)$ remains low. In the upper-middle layers, $A(l)$ increases sharply as the encoder begins to specialize for cross-modal matching. Once alignment dominates, D_{LCD} drops and the final layers collapse compositional distinctions into a syntax-blind, visually aligned representation.

This collapse is a direct consequence of the InfoNCE [32] training objective. Because negation is largely absent from web-crawled image–caption data, the contrastive loss is dominated by object-level semantics: the final layers learn to maximize visual alignment based on *what* objects are present, with no gradient signal to preserve the distinction between "a dog" and "not a dog." The compositional structure built by the middle layers is therefore overwritten as the encoder specializes for object-driven matching. We verify that this pattern is negation-specific through a control experiment in Sec. 4.4.

This trajectory identifies three functionally distinct zones that motivate our layer extraction strategy (Fig. 3):

Anchor zone (layer l_a). D_{LCD} is rising but $A(l)$ has not yet begun its steep ascent, providing a stable linguistic baseline uncontaminated by the visual prior.

Compositional peak (layer l_p). D_{LCD} reaches its maximum, the point at which syntactic and visual signals coexist before coming into conflict.

Collapse zone (layers l_p+1 through L). $A(l)$ dominates and D_{LCD} drops sharply; by layer L, the `[EOS]` embedding is visually aligned but syntax-blind.
Deterministic layer selection. Both layers are read directly off the LCD curve rather than tuned: the compositional peak is $l_p = \arg\max_\ell D_{\mathrm{LCD}}(\ell)$, and the anchor l_a is the latest pre-peak layer still in the rising-D_{LCD}, pre-alignment zone—the closest stable baseline to l_p, which shares the most processing context with it. Computing D_{LCD} requires only forward passes over a small diagnostic set and takes minutes, so layer selection is a one-time characterization, not a hyperparameter search. The same rule transfers across backbones: re-running the analysis on each encoder relocates l_p automatically, with no manual intervention (the supplementary material reports a peak shift to Layer 4 for a LAION-2B-pretrained encoder, with comparable gains).

3.3 PeakPatch

The compositional peak retains negation information but lacks visual alignment, while the final layer is visually aligned but syntax-blind; an effective correction must bridge both representations. PeakPatch does so through two lightweight modules that operate on frozen CLIP features (Fig. 3). The Embedding Correction Network (ECN) extracts a negation-specific deviation from the peak layer and injects it into the final text embedding, producing a corrected representation suitable for retrieval and generation. However, embedding correction alone may not suffice for discriminative tasks such as MCQ, where the decision hinges on the *relative* ranking of K candidate scores against a single image rather than on absolute embedding quality. A complementary Score Correction Network (SCN) therefore predicts a bounded scalar offset to the cosine similarity, allowing fine-grained adjustment of pairwise comparisons that the embedding shift cannot fully resolve.
Embedding Correction Network (ECN). The ECN corrects the collapsed layer-L embedding by recovering the negation signal from the compositional peak (Fig. 3, top branch). It predicts a deviation vector $\boldsymbol{\delta} \in \mathbb{R}^d$ that shifts the final embedding to separate negated from affirmative descriptions:

$$\hat{\mathbf{f}}_T = \frac{\mathbf{f}_T + \alpha \cdot \boldsymbol{\delta}}{\|\mathbf{f}_T + \alpha \cdot \boldsymbol{\delta}\|_2}, \tag{4}$$

where α is a learned scalar controlling the correction magnitude.

The deviation $\boldsymbol{\delta}$ is computed in two stages. All intermediate hidden states are first passed through CLIP's final layer normalization ($\mathrm{LN}_{\mathrm{final}}$) but *not* the text projection W_T, preserving the full token-level structure while ensuring numerical compatibility across layers; for notational brevity, we continue to write $\mathbf{H}^l$ and $\mathbf{h}^l$ for these layer-normalized representations in the equations below. In the first stage, a learned query $\mathbf{q} \in \mathbb{R}^d$ attends over the full token sequence at the peak layer via multi-head cross-attention [46] to produce a negation-aware summary:

$$\mathbf{a} = \mathrm{LN}\big(\mathrm{CrossAttn}(\mathbf{q}, \mathbf{H}^{l_p}, \mathbf{H}^{l_p})\big) \in \mathbb{R}^d, \tag{5}$$

where LN denotes a separate learned layer normalization. Because the query is learned end-to-end, it discovers negation-relevant token positions automatically, handling negation through a single mechanism without requiring a syntactic parser.

In the second stage, the cross-attention summary is concatenated with the `[EOS]` tokens from the peak and anchor layers and a mean-pooled representation of the peak layer, then passed through a bottleneck MLP:

$$\boldsymbol{\delta} = \mathrm{MLP}_{\mathrm{ECN}}\big([\mathbf{h}^{l_p}_{\texttt{[EOS]}};\, \mathbf{a};\, \mathbf{h}^{l_a}_{\texttt{[EOS]}};\, \bar{\mathbf{h}}^{l_p}]\big) \in \mathbb{R}^d, \tag{6}$$

where $\bar{\mathbf{h}}^{l_p}$ is the mean-pooled token representation at the peak layer and $\mathrm{MLP}_{\mathrm{ECN}} : \mathbb{R}^{4d} \to \mathbb{R}^d$. The anchor token $\mathbf{h}^{l_a}_{\texttt{[EOS]}}$ serves as a stable reference from before alignment pressure distorts the representation, while $\bar{\mathbf{h}}^{l_p}$ provides a global summary of the peak layer's full token sequence, complementing the query-focused cross-attention output. Since $\hat{\mathbf{f}}_T$ is ℓ_2-normalized (Eq. (4)), it preserves CLIP's cosine similarity interface and can serve as a drop-in replacement for retrieval and generation tasks.

Score Correction Network (SCN). While the ECN operates in embedding space, the SCN provides a complementary correction at the score level (Fig. 3, bottom branch). For discriminative tasks such as multiple-choice question answering, the SCN predicts a bounded scalar adjustment to the cosine similarity between an image and a candidate caption.

The SCN reads from the compositional peak region, where negation separability is high but the representation is not yet dominated by alignment pressure. This provides a compositional cue complementary to the token-level features the ECN extracts, making the score correction more robust on ambiguous inputs.

The SCN aggregates three sources of information. A *text context* encoder maps the concatenation of the projected peak-layer embedding $\mathbf{f}^{l_p}_T$ and the ECN-corrected embedding $\hat{\mathbf{f}}_T$ to a compact representation $\mathbf{t} \in \mathbb{R}^{d_s}$ (where $d_s \ll d$ is the SCN's internal dimension), capturing the discrepancy between the compositionally aware peak representation and the corrected output. A *cross-modal context* encoder maps the element-wise product $\mathbf{f}_I \odot \hat{\mathbf{f}}_T$ to $\mathbf{c} \in \mathbb{R}^{d_s}$, encoding how the corrected text relates to the image. Finally, two scalar *similarity features* $\mathbf{s} = [\cos(\mathbf{f}^{l_p}_T, \mathbf{f}_I);\ \cos(\hat{\mathbf{f}}_T, \mathbf{f}_I)]$ directly compare the peak-layer and corrected similarities; disagreement between these two scores signals that negation information was lost during the collapse. These three streams are concatenated and mapped to a bounded scalar correction:

$$\Delta s = \alpha_{\max} \cdot \tanh\big(\mathrm{MLP}_{\mathrm{SCN}}([\mathbf{s};\, \mathbf{t};\, \mathbf{c}])\big), \tag{7}$$

where the tanh bounds the output to $[-\alpha_{\max}, \alpha_{\max}]$, preventing the SCN from overriding the embedding-level signal. The final corrected similarity is:

$$\hat{s}(I, T) = \cos\big(\mathbf{f}_I,\, \hat{\mathbf{f}}_T\big) + \Delta s. \tag{8}$$

Because the SCN takes $\hat{\mathbf{f}}_T$ rather than the original layer-L embedding as input, embedding-level corrections propagate into the score-level module, coupling the two streams end-to-end.

Inference. At test time, PeakPatch applies the ECN and SCN uniformly to every input: it performs no polarity classification, gating, or parsing to decide whether a caption is negated, so affirmative and negated queries pass through the identical correction path. The ECN always returns a drop-in ℓ_2-normalized embedding $\hat{\mathbf{f}}_T$ and the SCN always returns a bounded offset Δs; on affirmative inputs the learned corrections are small, preserving standard retrieval (Sec. 4.4) and zero-shot classification (supplementary material).

3.4 Joint Training

Both modules are trained jointly end-to-end while all CLIP parameters remain frozen.

ECN objective. The ECN is trained with a symmetric (image-to-text + text-to-image) InfoNCE loss [32]. For a batch of B images, each paired with an affirmative and a negated caption, the contrastive denominator sums over all $2B$ corrected embeddings, so negated captions act as hard negatives that the loss explicitly pushes apart. We denote this loss $\mathcal{L}_{\text{ECN}}$.

SCN objective. The SCN is trained with a K-way softmax cross-entropy loss over multiple-choice questions, each consisting of one correct caption and $K-1$ negated distractors. An ℓ_2 penalty on the correction magnitude $(\Delta s)^2$, averaged over K options within each MCQ, regularizes the SCN to keep adjustments small. We denote this loss $\mathcal{L}_{\text{SCN}}$.

Joint optimization. The total objective combines both losses:

$$\mathcal{L} = \mathcal{L}_{\text{ECN}} + \lambda_{\text{SCN}}\, \mathcal{L}_{\text{SCN}}. \tag{9}$$

Because the SCN operates on ECN-corrected embeddings, gradients from $\mathcal{L}_{\text{SCN}}$ flow back into the ECN, creating a coupled training dynamic: the ECN must produce embeddings that are useful for both retrieval (via $\mathcal{L}_{\text{ECN}}$) and discrimination (via $\mathcal{L}_{\text{SCN}}$). We use separate learning rates for the two modules to balance the contrastive and discriminative objectives.

4 Experiments

4.1 Setup

Architecture and training. We build on CLIP ViT-B/32 [35] (d=512). The ECN and SCN add ∼4.7M and ∼0.5M parameters respectively, totaling 5.2M trainable parameters (3.5% of the frozen backbone). Training data is constructed from CC12M [4] following the negation generation protocol of NegBench [3]: for each original caption, LLaMA 3.1-8B [11] generates a semantically negated counterpart by inserting explicit negation (e.g., "no," "not," "without"), producing ∼1.06M image–caption pairs for the ECN contrastive objective and ∼313K four-option MCQ samples for the SCN. Both modules are trained jointly for 10 epochs with AdamW [28] (weight decay 10^{-2}, gradient clipping at norm 1.0), cosine

Table 1: Negation MCQ accuracy (%) on NegBench [3]. Aff, Neg, and Hyb denote affirmation, negation, and hybrid template types; Avg is computed over all samples across template types. All methods use CLIP ViT-B/32 [35] unless noted.

	COCO				VOC			
Method	Aff	Neg	Hyb	Avg	Aff	Neg	Hyb	Avg
CLIP [35]	70.0	6.6	38.4	39.2	80.9	3.0	58.0	37.9
Encoder fine-tuning								
NegCLIP [52] [ICLR'23]	49.2	13.9	16.3	26.8	70.5	4.6	42.3	30.2
CoN-CLIP [40] [WACV'25]	15.6	32.9	25.3	24.4	24.8	23.2	56.7	38.2
CLIP + NF [3] [CVPR'25]	73.1	33.2	54.7	54.2	85.0	31.7	79.5	60.1
NegCLIP + NF [3] [CVPR'25]	81.0	25.9	60.1	56.5	81.0	21.1	**83.7**	58.2
Post-hoc correction								
DCSM† [20] [ICCV'25]	71.2	6.6	68.0	48.6	68.5	5.4	73.1	49.0
PeakPatch	**98.1**	**63.2**	**60.7**	**74.3**	**99.7**	**57.9**	62.2	**65.5**

† Uses ViT-B/16 backbone. NF = NegFull [3] fine-tuning data.

annealing, and separate learning rates (10^{-5} for the ECN, 10^{-4} for the SCN). Batch size is 1024; training takes ∼2.7 hours on a single NVIDIA A100.

Evaluation. We evaluate on NegBench [3] under two protocols: *Negation MCQ*—four-choice accuracy on COCO [25] and VOC 2007 splits (affirmation, negation, hybrid templates); and *Negation retrieval*—text-to-image Recall@1/5 with negated queries on COCO and MSR-VTT [49]. We also evaluate on the *text-to-image generation* benchmark of Park *et al*. [33]: ECN-corrected embeddings are fed into a frozen GALIP [43] generator on 107 negation prompts; Gemma-3-27B [10] judges whether the affirmative object is present (Aff) and the negated attribute absent (Neg); combined (Comb = Aff × Neg) counts a sample as correct only when both hold. Zero-shot classification on CIFAR-100 is reported in the supplementary material.

Baselines. *Encoder fine-tuning*: NegCLIP [52], CoN-CLIP [40], NegationCLIP [33], and CLIP/NegCLIP + NF [3]. *Post-hoc*: DCSM [20] (dense token-to-patch matching, ViT-B/16).

4.2 Main Results

Negation MCQ. Table 1 reports multiple-choice accuracy on the COCO and VOC splits of NegBench. Baseline CLIP scores only 39.2% on COCO, with 70.0% on affirmative captions but 6.6% on negated ones. The best fine-tuning method (NegCLIP + NF) reaches 56.5%; among post-hoc methods, DCSM achieves 48.6%. PeakPatch achieves **74.3%** on COCO (+35.1 over CLIP, +17.8 over the best fine-tuning baseline) with a large negation gain (6.6%→63.2%). On VOC, PeakPatch reaches 65.5%, surpassing all fine-tuning baselines.

Negation retrieval. Table 2 reports text-to-image retrieval with negated queries. In the fully OOD setting (trained without COCO), PeakPatch outperforms all fine-tuning baselines across all metrics despite training only 5.2M parameters *vs*. ∼151M for full encoder fine-tuning. In-domain, PeakPatch reaches 37.1%

Table 2: Negation retrieval (Text→Image, %) on NegBench [3]. R@k: Recall@k on negated queries. All methods use CLIP ViT-B/32 [35] unless noted. FT = encoder fine-tuning; PH = post-hoc (frozen backbone).

			COCO		MSR-VTT	
Method		#P	R@1	R@5	R@1	R@5
CLIP [35]	–	–	25.0	47.9	23.8	45.9
Trained on COCO (COCO = in-domain)						
NegCLIP [52] [ICLR'23]	FT	151M	41.0	68.6	28.0	50.2
NegationCLIP [33] [ICCV'25]	FT	151M	38.6	65.8	**29.3**	**53.8**
NegCLIP + NF [3] [CVPR'25]	FT	151M	**41.3**	**69.0**	29.2	51.5
DCSM† [20] [ICCV'25]	PH	3.0M	10.6	28.6	19.4	41.2
PeakPatch	PH	5.2M	37.1	64.3	26.2	49.0
Trained w/o COCO (fully OOD)						
CoN-CLIP [40] [WACV'25]	FT	151M	25.7	50.1	23.3	45.4
CLIP + NF [3] [CVPR'25]	FT	151M	30.4	55.0	28.4	51.6
PeakPatch	PH	5.2M	**31.4**	**56.9**	**29.1**	**53.7**

† Uses ViT-B/16 backbone. NF = NegFull [3] fine-tuning data.

Table 3: Text-to-image generation with negated prompts (%). Image generated with frozen GALIP [43] generator on 107 prompts [33], correctness judged by Gemma-3-27B [10].

Method	Aff	Neg	Comb
CLIP [35]	97.4	29.2	28.3
NegCLIP [52]	**98.8**	24.5	23.7
NegationCLIP [33]	**98.8**	45.2	44.5
PeakPatch (ECN only)	98.1	**47.6**	**45.8**

Table 4: Generalization across architectures (MCQ Avg, %). PeakPatch consistently improves negation understanding across diverse VLM backbones without architecture-specific tuning.

	Base		+ PeakPatch	
Architecture	COCO	VOC	COCO	VOC
ViT-B/32 [35]	39.2	37.9	**74.3**	**65.5**
ViT-L/14 [35]	40.6	38.0	**63.8**	**52.5**
SigLIP [53]	28.9	30.8	**66.4**	**55.3**

R@1 and 64.3% R@5 on COCO, closing much of the gap to NegCLIP + NF (41.3% R@1) while modifying only 3.5% of the backbone. The remaining gap is expected: encoder fine-tuning reshapes the joint space for both modalities, whereas our correction operates only on text embeddings.

Text-to-image generation. Table 3 evaluates ECN-corrected embeddings in a downstream generation pipeline. The corrected embeddings raise the negation score from 29.2% to 47.6% (+18.4) while maintaining affirmative accuracy (98.1%), yielding a combined score of 45.8% *vs*. 28.3% for CLIP—surpassing NegationCLIP [33] (44.5%) without modifying encoder weights.

4.3 Ablation Study

Component contribution. Table 5 isolates the contribution of each module on COCO. The standalone ECN improves MCQ Avg from 39.2% to 51.2% and retrieval R@5 from 47.9% to 58.2%, showing that embedding-level correction benefits both tasks. The standalone SCN provides a larger MCQ gain (71.6%) but barely improves retrieval (48.1%), as score-level correction does not alter

Variant	MCQ Neg	MCQ Avg	Retrieval R@5
CLIP (baseline)	6.6	39.2	47.9
ECN only	16.5	51.2	58.2
SCN only	**67.3**	71.6	48.1
w/o anchor (l_a)	61.7	68.7	60.5
Detached SCN	57.3	67.1	**64.8**
PeakPatch (joint)	63.2	**74.3**	64.3

Table 5: Ablation study on COCO (%). Top: component contribution. Bottom: design choices. MCQ: negation accuracy and overall average. Ret: negation retrieval R@5.

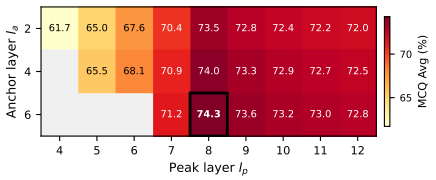

Fig. 4: ECN layer selection. COCO MCQ Avg as a function of peak layer l_p and anchor layer l_a. Accuracy peaks at l_p=8, coinciding with the LCD maximum (Sec. 3.2), and is robust to the anchor choice.

the embedding space. Joint training yields the best overall system (74.3% MCQ, 64.3% R@5), with the ECN and SCN co-adapting through shared gradients.

Design choices. Table 5 also ablates two key architectural decisions. Removing the anchor layer (l_a) reduces accuracy by −5.6 (74.3%→68.7%), confirming that the stable baseline from the anchor zone is critical for measuring compositional change. Detaching the SCN from ECN gradients reduces accuracy by −7.2 (74.3%→67.1%), showing that the cooperative training dynamic between the two modules is essential.

Layer selection. Figure 4 sweeps the ECN peak layer l_p and anchor layer l_a across CLIP's text-encoder layers. Accuracy peaks at l_p=8, the same layer where the LCD trajectory reaches its maximum (Sec. 3.2), and decreases for both earlier and later choices. Performance is robust to the anchor choice, varying by $<$1 pp across $l_a \in \{2, 4, 6\}$ for a fixed peak, confirming that the peak layer is the critical design decision and that the LCD analysis provides a principled criterion for selecting it.

4.4 Analysis

Generalization across architectures. Table 4 evaluates PeakPatch on additional backbones. Negation blindness persists regardless of scale or objective—CLIP ViT-L/14 (40.6%) and SigLIP ViT-B/16 (28.9%) match ViT-B/32 (39.2%)—and PeakPatch consistently improves COCO MCQ: +35.1 on ViT-B/32, +23.2 on ViT-L/14 (to 63.8%), and +37.5 on SigLIP (to 66.4%). The ViT-L/14 LCD trajectory shows an identical collapse pattern, and the layer-selection rule transfers across pretraining data and depth (OpenCLIP ViT-B/32 on LAION-2B, ViT-g/14); see supplementary material.

Standard (non-negation) retrieval. Because PeakPatch corrects every query without gating (Sec. 3.3), we check that it does not harm ordinary retrieval. On standard (affirmative) MSCOCO 5K and Flickr30K 1K text-to-image retrieval (Tab. 6), ECN-corrected embeddings slightly *improve* over frozen CLIP (R@1

Table 6: Standard (non-negation) text-to-image retrieval (%). ECN-corrected embeddings preserve—and slightly improve—retrieval on affirmative captions.

	MSCOCO 5K		Flickr30K 1K	
Method	R@1	R@5	R@1	R@5
CLIP [35]	29.9	54.1	57.9	83.0
PeakPatch (ECN)	**32.1**	**57.0**	**60.4**	**83.9**

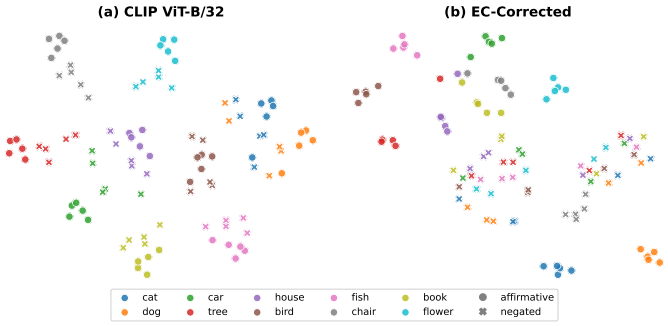

Fig. 5: t-SNE of text embeddings for 10 object categories. Circles: affirmative captions; crosses: negated captions; colors: object categories. *Left*: original CLIP embeddings (affirmative and negated overlap within each object). *Right*: after ECN correction (affirmative captions stay in tight per-category clusters; negated captions separate into a looser shared region).

+2.2/+2.5), and zero-shot classification is likewise preserved (supplementary material).

Embedding space visualization. Figure 5 shows t-SNE projections of text embeddings for 10 object categories, each with 5 affirmative and 5 negated caption templates. In the original CLIP space (*left*), affirmative and negated embeddings for the same object cluster together, confirming that the encoder collapses the syntactic distinction. After ECN correction (*right*), the affirmative captions keep their tight per-category clusters while the negated captions separate from them into a distinct, looser region. We discuss negation's set-valued nature in the supplementary material.

LCD control experiment. To verify that the LCD trajectory reflects a negation-specific phenomenon, we compare three conditions on 1K pairs (Fig. 6): negation, paraphrase (synonym rewording), and random (unrelated captions). The paraphrase curve stays near zero (meaning changes do not alter the `[EOS]` trajectory), the random curve rises monotonically, and only the negation curve exhibits the characteristic rise-and-fall, isolating Representational Collapse as unique to negation.

Attention analysis. Quantmeyer *et al.* [34] found that only 8% of CLIP's attention heads are negator-selective. Figure 7 measures per-head attention on

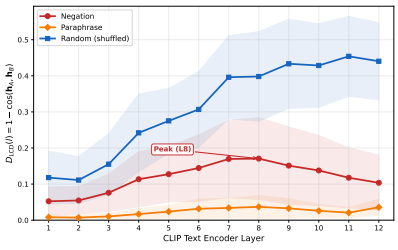

Fig. 6: LCD control experiment. LCD is measured on 1K pairs under three conditions: negation, paraphrase, and random. Only negation exhibits the rise-and-fall pattern, confirming the signal is negation-specific.

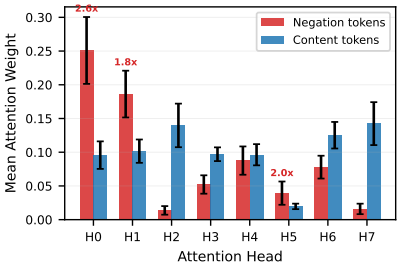

Fig. 7: ECN cross-attention by token type. Mean attention on negation vs. content tokens. Heads H0, H1, and H5 attend preferentially to negation tokens (>1.5×), compared to 8% of CLIP heads [34].

negation *vs*. content tokens across 12 negated prompts. Three of eight ECN heads (H0, H1, H5) allocate 1.8–2.6× more attention to negation tokens (37.5% negation-selective); the remaining heads focus on content words, providing semantic context for *what* is being negated.

5 Conclusion

We identified *Representational Collapse* in CLIP's text encoder: intermediate layers build compositional structure separating negated from affirmative descriptions, but later layers overwrite it as they specialize for visual alignment. Guided by this finding, we proposed PeakPatch, a post-hoc system that extracts the negation signal at the compositional peak through an Embedding Correction Network (ECN) and applies bounded score offsets through a Score Correction Network (SCN), trained jointly with all CLIP parameters frozen—adding 5.2M parameters (3.5% of the backbone). On NegBench, PeakPatch reaches 74.3% COCO MCQ (+35.1 over CLIP, +17.8 over the best fine-tuning method), outperforms all fine-tuning baselines on fully out-of-distribution retrieval, transfers to text-to-image generation (+18.4 negation score), and generalizes across ViT-B/32, ViT-L/14, and SigLIP. We discuss limitations in the supplementary material.

6 Acknowledgements

This material is based in part upon work supported by the National Science Foundation under Grant Numbers CNS-2333487 (NSF Frontier) and CNS-2146449 (NSF CAREER). Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors. We thank the reviewers and area chairs for their constructive feedback, which helped improve this work.

References

1. Aggarwal, B., More, A., Soni, M., Bhat, S.D.: Seeing what's not there: Negation understanding needs more than training. In: ICLR (2026)
2. Alayrac, J.B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., Ring, R., Rutherford, E., Cabi, S., Han, T., Gong, Z., Samangooei, S., Monteiro, M., Menick, J., Borgeaud, S., Brock, A., Nematzadeh, A., Sharifzadeh, S., Bińkowski, M., Barreira, R., Vinyals, O., Zisserman, A., Simonyan, K.: Flamingo: a visual language model for few-shot learning. In: NeurIPS. pp. 23716–23736 (2022)
3. Alhamoud, K., Alshammari, S., Tian, Y., Li, G., Torr, P.H., Kim, Y., Ghassemi, M.: Vision-language models do not understand negation. In: CVPR. pp. 29612–29622 (2025)
4. Changpinyo, S., Sharma, P., Ding, N., Soricut, R.: Conceptual 12M: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In: CVPR. pp. 3558–3568 (2021)
5. Cherti, M., Beaumont, R., Wightman, R., Wortsman, M., Ilharco, G., Gordon, C., Schuhmann, C., Schmidt, L., Jitsev, J.: Reproducible scaling laws for contrastive language-image learning. In: CVPR. pp. 2818–2829 (2023)
6. Dai, W., Li, J., Li, D., Tiong, A.M.H., Zhao, J., Wang, W., Li, B., Fung, P., Hoi, S.C.H.: InstructBLIP: Towards general-purpose vision-language models with instruction tuning. In: NeurIPS. pp. 49250–49267 (2023)
7. Doveh, S., Arbelle, A., Harary, S., Herzig, R., Kim, D., Cascante-Bonilla, P., Alfassy, A., Panda, R., Giryes, R., Feris, R., Ullman, S., Karlinsky, L.: Dense and aligned captions (DAC) promote compositional reasoning in VL models. In: NeurIPS. vol. 36, pp. 76137–76150 (2023)
8. Gandelsman, Y., Efros, A.A., Steinhardt, J.: Interpreting CLIP's image representation via text-based decomposition. In: ICLR (2024)
9. Gao, P., Geng, S., Zhang, R., Ma, T., Fang, R., Zhang, Y., Li, H., Qiao, Y.: CLIP-Adapter: Better vision-language models with feature adapters. IJCV **132**, 581–595 (2024)
10. Gemma Team: Gemma 3 technical report. arXiv preprint arXiv:2503.19786 (2025)
11. Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al.: The llama 3 herd of models. arXiv preprint arXiv:2407.21783 (2024)
12. Hossain, M.M., Chinnappa, D., Blanco, E.: An analysis of negation in natural language understanding corpora. In: Proc. ACL (Short Papers). pp. 716–723 (2022)
13. Hsieh, C.Y., Zhang, J., Ma, Z., Kembhavi, A., Krishna, R.: SugarCrepe: Fixing hackable benchmarks for vision-language compositionality. In: NeurIPS. vol. 36, pp. 31096–31116 (2023)
14. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: LoRA: Low-rank adaptation of large language models. In: ICLR (2022)
15. Huang, Y., Tang, J., Chen, Z., Zhang, R., Zhang, X., Chen, W., Zhao, Z., Zhao, Z., Lv, T., Hu, Z., Zhang, W.: Structure-CLIP: Towards scene graph knowledge to enhance multi-modal structured representations. In: AAAI. vol. 38, pp. 2417–2425 (2024)
16. Huang, Y., Shakeri, F., Dolz, J., Boudiaf, M., Bahig, H., Ben Ayed, I.: LP++: A surprisingly strong linear probe for few-shot CLIP. In: CVPR. pp. 23773–23782 (2024)

17. Jia, C., Yang, Y., Xia, Y., Chen, Y.T., Parekh, Z., Pham, H., Le, Q.V., Sung, Y., Li, Z., Duerig, T.: Scaling up visual and vision-language representation learning with noisy text supervision. In: ICML. pp. 4904–4916 (2021)
18. Jia, M., Tang, L., Chen, B.C., Cardie, C., Belongie, S., Hariharan, B., Lim, S.N.: Visual prompt tuning. In: ECCV. pp. 709–727 (2022)
19. Kamath, A., Singh, M., LeCun, Y., Synnaeve, G., Misra, I., Carion, N.: MDETR – modulated detection for end-to-end multi-modal understanding. In: ICCV. pp. 1780–1790 (2021)
20. Kang, R., Song, Y., Gkioxari, G., Perona, P.: Is CLIP ideal? No. can we fix it? Yes! In: ICCV. pp. 22436–22446 (2025)
21. Khattak, M.U., Rasheed, H., Maaz, M., Khan, S., Khan, F.S.: MaPLe: Multi-modal prompt learning. In: CVPR. pp. 19113–19122 (2023)
22. Li, J., Li, D., Savarese, S., Hoi, S.: BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In: ICML. pp. 19730–19742 (2023)
23. Li, J., Selvaraju, R.R., Gotmare, A.D., Joty, S., Xiong, C., Hoi, S.: Align before fuse: Vision and language representation learning with momentum distillation. In: NeurIPS. pp. 9694–9705 (2021)
24. Li, L.H., Zhang, P., Zhang, H., Yang, J., Li, C., Zhong, Y., Wang, L., Yuan, L., Zhang, L., Hwang, J.N., Chang, K.W., Gao, J.: Grounded language-image pre-training. In: CVPR. pp. 10965–10975 (2022)
25. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common objects in context. In: ECCV. pp. 740–755 (2014)
26. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. In: NeurIPS. pp. 34892–34916 (2023)
27. Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Jiang, Q., Li, C., Yang, J., Su, H., Zhu, J., Zhang, L.: Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. In: ECCV (2024)
28. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: ICLR (2019)
29. Lüddecke, T., Ecker, A.: Image segmentation using text and image prompts. In: CVPR. pp. 7086–7096 (2022)
30. Luo, H., Ji, L., Zhong, M., Chen, Y., Lei, W., Duan, N., Li, T.: CLIP4Clip: An empirical study of CLIP for end to end video clip retrieval and captioning. Neurocomputing **508**, 293–304 (2022)
31. Ma, Z., Hong, J., Gul, M.O., Gandhi, M., Gao, I., Krishna, R.: CREPE: Can vision-language foundation models reason compositionally? In: CVPR. pp. 10910–10921 (2023)
32. van den Oord, A., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)
33. Park, J., Lee, J., Song, J., Yu, S., Jung, D., Yoon, S.: Know "no" better: A data-driven approach for enhancing negation awareness in CLIP. In: ICCV. pp. 2825–2835 (2025)
34. Quantmeyer, V., Mosteiro, P., Gatt, A.: How and where does CLIP process negation? In: Proceedings of the 3rd Workshop on Advances in Language and Vision Research (ALVR). pp. 59–72 (2024)
35. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: ICML. pp. 8748–8763 (2021)

36. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with CLIP latents. arXiv preprint arXiv:2204.06125 (2022)
37. Ranjbar, S.K., Alhamoud, K., Ghassemi, M.: SpaceVLM: Sub-space modeling of negation in vision-language models. arXiv preprint arXiv:2511.12331 (2025)
38. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR. pp. 10684–10695 (2022)
39. Singh, A., Hu, R., Goswami, V., Couairon, G., Galuba, W., Rohrbach, M., Kiela, D.: FLAVA: A foundational language and vision alignment model. In: CVPR. pp. 15638–15650 (2022)
40. Singh, J., Shrivastava, I., Vatsa, M., Singh, R., Bharati, A.: Learning the power of "no": Foundation models with negations. In: WACV. pp. 7991–8001 (2025)
41. Sun, Q., Fang, Y., Wu, L., Wang, X., Cao, Y.: EVA-CLIP: Improved training techniques for CLIP at scale. arXiv preprint arXiv:2303.15389 (2023)
42. Tang, Y., Lin, Z., Wang, Q., Zhu, P., Hu, Q.: AMU-Tuning: Effective logit bias for CLIP-based few-shot learning. In: CVPR. pp. 23323–23333 (2024)
43. Tao, M., Bao, B.K., Tang, H., Xu, C.: GALIP: Generative adversarial CLIPs for text-to-image synthesis. In: CVPR. pp. 14214–14223 (2023)
44. Thrush, T., Jiang, R., Bartolo, M., Singh, A., Williams, A., Kiela, D., Ross, C.: Winoground: Probing vision and language models for visio-linguistic compositionality. In: CVPR. pp. 5238–5248 (2022)
45. Udandarao, V., Gupta, A., Albanie, S.: SuS-X: Training-free name-only transfer of vision-language models. In: ICCV. pp. 2725–2736 (2023)
46. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: NeurIPS. pp. 5998–6008 (2017)
47. Wortsman, M., Ilharco, G., Kim, J.W., Li, M., Kornblith, S., Roelofs, R., Lopes, R.G., Hajishirzi, H., Farhadi, A., Namkoong, H., Schmidt, L.: Robust fine-tuning of zero-shot models. In: CVPR. pp. 7959–7971 (2022)
48. Xu, J., De Mello, S., Liu, S., Byeon, W., Breuel, T., Kautz, J., Wang, X.: GroupViT: Semantic segmentation emerges from text supervision. In: CVPR. pp. 18134–18144 (2022)
49. Xu, J., Mei, T., Yao, T., Rui, Y.: MSR-VTT: A large video description dataset for bridging video and language. In: CVPR. pp. 5288–5296 (2016)
50. Yao, L., Han, J., Wen, Y., Liang, X., Xu, D., Zhang, W., Li, Z., Xu, C., Xu, H.: DetCLIP: Dictionary-enriched visual-concept paralleled pre-training for open-world detection. In: NeurIPS. pp. 9125–9138 (2022)
51. Yu, T., Lu, Z., Jin, X., Chen, Z., Wang, X.: Task residual for tuning vision-language models. In: CVPR. pp. 10899–10909 (2023)
52. Yuksekgonul, M., Bianchi, F., Kalluri, P., Jurafsky, D., Zou, J.: When and why vision-language models behave like bags-of-words, and what to do about it? In: ICLR (2023)
53. Zhai, X., Mustafa, B., Kolesnikov, A., Beyer, L.: Sigmoid loss for language image pre-training. In: ICCV. pp. 11975–11986 (2023)
54. Zhang, R., Fang, R., Zhang, W., Gao, P., Li, K., Dai, J., Qiao, Y., Li, H.: Tip-adapter: Training-free adaption of CLIP for few-shot classification. In: ECCV. pp. 491–507 (2022)
55. Zhao, T., Zhang, T., Zhu, M., Shen, H., Lee, K., Lu, X., Yin, J.: An explainable toolbox for evaluating pre-trained vision-language models. In: Proc. EMNLP (System Demonstrations). pp. 30–37 (2022)
56. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Conditional prompt learning for vision-language models. In: CVPR. pp. 16816–16825 (2022)

57. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. IJCV **130**, 2337–2348 (2022)
58. Zhu, X., Zhang, R., He, B., Zhou, A., Wang, D., Zhao, B., Gao, P.: Not all features matter: Enhancing few-shot CLIP with adaptive prior refinement. In: ICCV. pp. 2605–2615 (2023)