LACE: Latent Alignment via Counterfactual Embeddings

Harsh Udai* **Konda Reddy Mopuri**
Department of Artificial Intelligence,
IIT Hyderabad, India
ai23resch11003@iith.ac.in,
krmopuri@ai.iith.ac.in
Project Page Code

Abstract

Improving compositional understanding in CLIP-style vision-language models remains difficult; models with strong coarse alignment often fail on attribute binding, relations, and multi-object semantics. A common remedy is text-side hard negatives (negations, perturbations, or mined confusions), but concentrating difficulty on the text branch can skew the training signal, exacerbate modality asymmetry, and distort joint-space geometry (e.g., increased hubness and reduced mutual reciprocity), even when Recall@K improves. We propose **LACE** (**L**atent **A**lignment via **C**ounterfactual **E**mbeddings), a lightweight approach that injects structured, semantics-preserving latent edits into the image and text embedding streams without pixel-space editing or external generators. LACE synthesizes counterfactual image and text embeddings by attenuating, swapping, or transplanting a small set of factors tied to target objects, attributes, or relations, producing image-side and text-side hard negatives that rebalance supervision while preserving global alignment. Across compositional and retrieval benchmarks, LACE improves robustness to binding and relational confusions and yields a healthier embedding geometry with improved reciprocity and competitive hubness/local-stability trade-offs.

1 Introduction

Human perception and reasoning are fundamentally *compositional*: we effortlessly distinguish *"a red suitcase next to a black backpack"* from *"a black suitcase next to a red backpack"*, despite identical objects and attributes. Modern vision-language models (VLMs), despite their strong coarse-grained recognition and retrieval capabilities, often collapse such meaningfully different configurations into similar representations, exposing brittle attribute binding, weak role assignment, and poor relational reasoning. This failure directly limits fine-grained retrieval, relation-sensitive VQA, grounded reasoning, and any downstream application that requires knowing not merely *what* objects are present, but *which* attributes belong to *which* objects and how those objects relate spatially or semantically [1, 2]. We argue that this weakness is not simply a matter of insufficient scale, but a consequence of the supervision signal itself. Standard CLIP-style contrastive learning aligns global image and text embeddings using massive weakly supervised web data [3], yet such captions often enumerate entities while leaving attribute-object bindings and relational structure underspecified [4]. As a result, the contrastive objective admits shortcut solutions based on lexical co-occurrence and coarse semantic overlap rather than enforcing genuine compositional constraints. Hard negatives are a natural response, but most existing methods construct them primarily on the text side through swaps, negations, or caption perturbations because they are cheap to generate and easy to insert

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

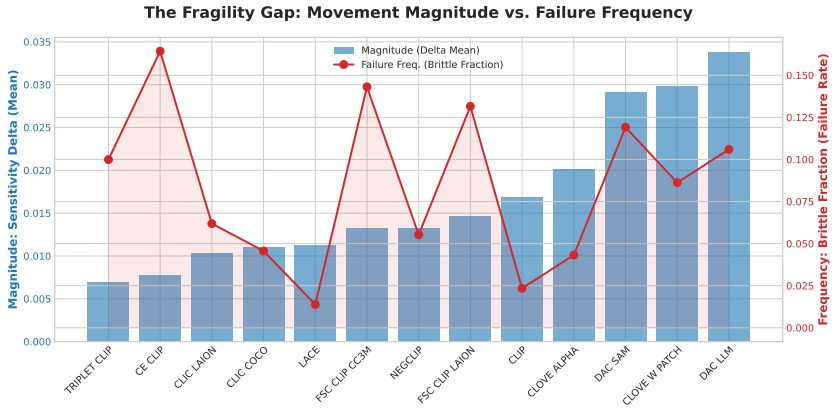

Figure 1: **Fragility gap in local edit geometry.** Bars show average edit displacement (Δ mean), while red markers show average brittle failure rate, both averaged over relation and attribute swaps. Movement magnitude alone does not predict local stability: some methods with modest displacement still fail frequently. Full local-edit statistics are provided in App. C.1.

into contrastive training. This design creates an asymmetric training signal: the text branch is forced to resolve fine-grained alternatives, while the image branch receives no comparable visual near-miss constraint. We show that this imbalance can distort the shared embedding space, increasing hubness, reducing mutual reciprocity, and destabilizing neighborhoods even when some retrieval or compositional scores improve (Table 1). A principled remedy requires image-side counterfactuals–visually similar alternatives that differ only in a targeted object, attribute, or relation–but pixel-space editing is expensive and artifact-prone, while corpus mining requires large-scale retrieval and risks false negatives.

To address this bottleneck, we introduce **LACE** (**L**atent **A**lignment via **C**ounterfactual **E**mbeddings), a lightweight framework that generates structured image-side and text-side counterfactuals directly in embedding space, without pixel-level manipulation, diffusion models, external generators, or retrieval mining. LACE represents each image or text embedding as a sparse mixture of reusable concept atoms corresponding to objects, attributes, and relations, inferred through sparse coding over a fixed concept dictionary. It then synthesizes counterfactual embeddings by targeted attenuation, swaps, or transplants over a small active support while preserving the surrounding semantic context. Because these edits are pure vector operations, LACE enables scalable on-the-fly hard negatives on both modalities within a symmetric contrastive objective, supplying the missing fine-grained visual pressure while preserving global image-text alignment. Importantly, LACE changes only training: at inference, the model remains a standard dual encoder with no dictionary lookup, sparse solver, or counterfactual-generation module. Our evaluation therefore goes beyond headline benchmark accuracy. We report standard compositional performance on COLA, What'sUp, Winoground, SugarCrepe/SC++, and MMVP-VLM, while also measuring whether gains preserve a healthy shared embedding geometry through hubness, mutual reciprocity and local edit stability (Tables 1, 2, and App. C.1). We further validate LACE beyond the InfoNCE setting by adapting it to SigLIP's objective, where counterfactuals are introduced as explicit negative image-text pairs. In summary, our contributions are fourfold:

i. **Geometry diagnosis of text-only hard negatives:** We show that text-centric hard negatives can induce geometric failure modes, amplifying modality asymmetry and degrading neighborhood structure through increased hubness and reduced reciprocity.

ii. **LACE for embedding-space counterfactuals:** We propose **LACE**, a scalable framework for generating structured image-side and text-side counterfactual embeddings via sparse concept decomposition, without pixel-space editing, external generators, or retrieval mining.

iii. **Compositional gains with healthier geometry:** We demonstrate that LACE improves attribute binding, relational sensitivity, and fine-grained visual understanding while producing a healthier shared embedding space, with stronger reciprocity, fewer extreme hubs, and more stable local edits.

iv. **Cross-objective validation in dual encoders:** We evaluate LACE on both CLIP and SigLIP, showing that sparse latent counterfactual supervision is not tied to a single softmax InfoNCE formulation but also transfers to pairwise sigmoid dual-encoder training.

Table 1: **Global embedding-geometry diagnostics on CC3M-50k** ($K = 10$). **Hubness** is measured by Gini and Top-1% mass (lower is better, ↓). **Reciprocity** measures neighborhood symmetry (higher is better, ↑). Top-1% mass and reciprocity are reported in %.

	Hubness (I2T) @10		**Hubness (T2I) @10**		**Reciprocity @10 (%)**	
Method	**Gini ↓**	**Top-1% ↓**	**Gini ↓**	**Top-1% ↓**	**Mutual ↑**	**Edge ↑**
Pretrained backbone						
CLIP ViT-B/32	0.459	7.52	0.493	6.92	50.30	52.18
External checkpoints (contextual)						
FSC-CLIP	0.708	11.76	0.536	9.70	30.36	35.95
CLoVe	0.485	5.82	0.426	6.05	45.23	51.17
DAC	0.834	18.81	0.509	7.34	24.07	26.47
CLIC	0.544	7.63	0.473	6.40	38.77	46.45
CE-CLIP	0.856	22.89	0.626	19.63	17.16	23.01
NegCLIP	0.590	9.07	0.469	6.77	35.74	44.48
TripletCLIP	0.527	5.10	0.403	4.57	46.90	50.48
Controlled fine-tuning comparison						
Matched FT	0.480	7.30	0.417	5.40	46.57	52.46
LACE	0.455	6.96	0.450	6.06	51.41	53.79

2 Related Work

Compositional Gaps in Vision–Language Models. Dual-encoder VLMs such as CLIP [3, 5] achieve strong zero-shot transfer through global image-text alignment, but often miss fine-grained compositional structure, failing on attribute binding, spatial relations, and multi-object semantics [6, 2]. Benchmarks such as Winoground [7], SugarCrepe [8], and What'sUp [1] expose these failures with near-miss distractors. LACE addresses this gap through counterfactual embedding-space supervision without explicit structured annotations.

Structured Supervision and Hard Negatives. Prior work improves compositionality by injecting structure through scene graphs [9, 10], object-centric representations [11], or hard negatives constructed from perturbed captions [12, 13]. While structured approaches can improve binding and relational reasoning, they often depend on external parsers or additional pipelines. Text-side hard negatives are simpler and widely used in methods such as DAC [14] and CLoVe [15], but they concentrate fine-grained pressure on the text branch while leaving the image branch comparatively under-constrained. LACE instead generates hard negatives for both modalities, restoring a more symmetric training signal without requiring structured supervision.

Closest Related Work. Recent methods move beyond text-only perturbations by constructing image-side counterfactual supervision. SPARCL [16], GCS [17], and CounterCurate [18] generate or curate counterfactual image-caption pairs to improve attribute, relation, position, and counting sensitivity. While these works demonstrate the value of image-side counterfactuals, they rely on diffusion models, grounded editing, offline curation, or pixel-level manipulation, introducing cost, alignment noise, and artifact risks. In contrast, LACE synthesizes image-side and text-side counterfactuals directly in the shared embedding space, enabling on-the-fly symmetric supervision without external generators. The closest embedding-space alternative is AHNPL [19] (compared in App. B.2), which maps text-side semantic shifts into the visual embedding space to create image-based hard negatives. LACE instead edits sparse concept supports rather than applying a dense text-difference vector, enabling localized attenuation, swaps, and transplants over objects, attributes, and relations with the same operator for both modalities. Related work shows that VLM embeddings can contain approximately factorized semantic directions [20, 21]; LACE turns this structure into a training-time counterfactual generator rather than using it only for analysis.

Embedding Geometry and Latent Concept Editing. Text-centric hard negatives can reshape the joint embedding space asymmetrically: the text encoder is pushed to resolve fine-grained distinctions, while the visual manifold receives weaker compositional pressure. Such imbalance can worsen hubness, reduce neighborhood reciprocity, and widen modality gaps [22, 23]. LACE targets these geometric failures during training by introducing modality-symmetric counterfactual supervision in embedding space. Our approach is inspired by dictionary learning and sparse autoencoder work

that decomposes model representations into interpretable components [24, 25]; however, rather than using sparse decompositions only for interpretation, we use them for synthesis, generating controlled counterfactual hard negatives without diffusion, GAN-based editing, or other pixel-space generators [26, 27]. Additional related work is discussed in Appendix A.

3 Preliminaries

3.1 CLIP-Style Contrastive Learning

CLIP-style models [3] use a dual-encoder architecture: a vision encoder $f_\theta(\cdot)$ and a text encoder $g_\phi(\cdot)$ map an image I_i and its paired caption T_i to embeddings $\mathbf{v}_i, \mathbf{u}_i \in \mathbb{R}^d$, respectively. Embeddings are ℓ_2-normalized. We use scaled cosine similarity $s(\mathbf{v}, \mathbf{u}) = \mathbf{v}^\top \mathbf{u}/\tau$, where τ is a learnable temperature. Training minimizes the symmetric InfoNCE loss over a batch of N pairs:

$$\mathcal{L}_{\text{I2T}} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp(s(\mathbf{v}_i, \mathbf{u}_i))}{\sum_{j=1}^{N} \exp(s(\mathbf{v}_i, \mathbf{u}_j))}, \; \mathcal{L}_{\text{T2I}} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp(s(\mathbf{u}_i, \mathbf{v}_i))}{\sum_{j=1}^{N} \exp(s(\mathbf{u}_i, \mathbf{v}_j))}. \quad (1)$$

While effective for global alignment, this objective often under-specifies the fine-grained structure needed for compositional reasoning [3].

3.2 Hard Negatives in CLIP

Hard negatives (HNs) are "near-miss" distractors that sharpen contrastive boundaries, typically by enlarging the InfoNCE denominator or adding an auxiliary ranking term. We use the scaled similarity notation from Sec. 3.1.

Text-side hard negatives (T-HN). Text-side hard negatives pair each image I_i with perturbed captions $\{\tilde{T}_{i,k}\}_{k=1}^{K}$–e.g., attribute swaps, relation flips, or negations–and use their embeddings $\tilde{\mathbf{u}}_{i,k}$ as additional negatives for image-to-text contrastive training. While effective for sharpening caption discrimination, many methods leave the text-to-image objective unchanged or only weakly coupled, producing an asymmetric signal: the text branch is forced to resolve fine-grained alternatives, but the image branch receives no comparable visual counterfactual constraint [28, 14, 15, 12, 27, 29, 30].

Image-side hard negatives (I-HN). Analogously, one can add hard-negative images to the T2I denominator. However, generating controlled semantic counterfactual images is typically expensive (editing/synthesis/mining), so most prior work defaults to text-only HNs [28, 27, 31].

Auxiliary ranking-style HN losses. Some methods instead use a ranking objective [29, 13]. Minimizing this loss encourages ($\gamma\,\Delta_{i,k} \ll 0$):

$$\mathcal{L}_{\text{rank}} = \frac{1}{NK}\sum_{i=1}^{N}\sum_{k=1}^{K} \log\left(1 + \exp\left(\gamma\,\Delta_{i,k}\right)\right), \quad \Delta_{i,k} = s(v_i, \tilde{u}_{i,k}) - s(v_i, u_i). \quad (2)$$

3.3 Why asymmetry matters

While the above strategies can improve compositional benchmarks, adding HNs primarily on the text side induces asymmetric gradient pressure, reshaping neighborhood structure unevenly across modalities. Since retrieval uses the shared embedding space as the interface, we argue that evaluation should go beyond Recall@K and include geometry-aware diagnostics of multimodal structure.

4 Understanding the Multimodal Space

CLIP-style models are typically evaluated with Recall@K, but this aggregate metric can obscure how training objectives reshape the geometry of the joint embedding space. High Recall can coexist with a "warped" topology–e.g., unstable, non-reciprocal neighborhoods or heavy mass concentration. To expose these effects, we complement standard retrieval metrics with three diagnostic families: (1) **Neighborhood Reciprocity**, measuring bidirectional alignment; (2) **Hubness and Mass Concentration**, quantifying nearest-neighbor "crowding"; and (3) **Local Edit Geometry**, probing stability under controlled semantic edits.

4.1 Notation

Given N paired samples $\{(I_i, T_i)\}_{i=1}^N$, let $v_i, u_i \in \mathbb{R}^d$ denote ℓ_2-normalized image and text embeddings. Similarity is $s(v, u) = \tau^{-1} v^\top u$. For an image query I_i, let $\mathcal{N}_K^T(i)$ be the indices of the top-K texts ranked by $s(v_i, u_j)$; for a text query T_j, let $\mathcal{N}_K^I(j)$ be the indices of the top-K images ranked by $s(u_j, v_i)$. Let $\mathrm{rank}_{\mathrm{I2T}}(i)$ be the rank of the ground-truth caption T_i among $\{T_j\}_{j=1}^N$ for query I_i (1 is best), and define $\mathrm{rank}_{\mathrm{T2I}}(i)$ analogously.

4.2 Neighborhood Reciprocity and Symmetry

Directional recall only measures if a model can find a target in one direction. Reciprocity diagnostics assess whether the shared space is truly symmetric, ensuring that if an image is "near" a text, that text is also "near" the image.

(1a) Recall@K (directional).

$$\mathrm{R@K}_{\mathrm{I2T}} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}[\mathrm{rank}_{\mathrm{I2T}}(i) \le K], \quad \mathrm{R@K}_{\mathrm{T2I}} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}[\mathrm{rank}_{\mathrm{T2I}}(i) \le K]. \tag{3}$$

(1b) Mutual reciprocity (pair-level). Mutual reciprocity provides a stricter measure of alignment by requiring the ground-truth pair to be among the top-K nearest neighbors of each other simultaneously. A large gap between Recall and Mutual Reciprocity suggests the space is poorly calibrated across modalities.

$$\mathrm{Mutual@K} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\Big[(i \in \mathcal{N}_K^I(i)) \ \wedge \ (i \in \mathcal{N}_K^T(i))\Big]. \tag{4}$$

(1c) Edge reciprocity (graph-level). This generalizes reciprocity to the entire neighborhood. It measures the percentage of all KNN links that are bidirectional, indicating the overall structural consistency of the bipartite image-text graph.

$$\mathrm{EdgeRecip@K} = \frac{\big|\{(i,j) \in E_{\mathrm{I2T}} : (j,i) \in E_{\mathrm{T2I}}\}\big|}{NK}. \tag{5}$$

4.3 Hubness Metrics

Hubness is a phenomenon in high-dimensional spaces where certain points appear as nearest neighbors for a disproportionately large number of queries, effectively "crowding out" correct matches.

(2a) Hubness count (in-degree). We calculate how frequently an item j appears in the top-K lists of the opposite modality. Significant variance in these counts indicates a distorted topology.

$$h_j^T = \big|\{i : j \in \mathcal{N}_K^T(i)\}\big| \quad \text{for } j \in \{1, \dots, N\}. \tag{6}$$

(2b) Gini coefficient (inequality of hub counts). The Gini coefficient provides a single scalar to quantify the "inequality" of these counts. A high Gini coefficient ($G \to 1$) signals the presence of "super-hubs" that dominate the embedding space and degrade retrieval reliability.

$$\mathrm{Gini}(h) = \frac{\sum_{a=1}^{N}\sum_{b=1}^{N} |h_a - h_b|}{2N\sum_{a=1}^{N} h_a}. \tag{7}$$

To further characterize the tail of this distribution, we report the **Top-1% Mass** and **Max Hub Count** in App. C.6, which measure the concentration of neighborhood "votes" in the most dominant points.

4.4 Local Edit Geometry (Smoothness / Stability)

A compositionally robust model should be sensitive to small semantic changes while maintaining local neighborhood stability. We use **counterfactual edits** (e.g., altering object or attribute concepts, such as *"cat"→"dog"* or *"red"→"blue"*) to probe this.

(3a) Text embedding displacement. We measure how much a semantic edit moves the corresponding text embedding. Since all embeddings are ℓ_2-normalized, cosine displacement and Euclidean

displacement are monotonic transformations of each other. We therefore report cosine displacement:

$$\mathrm{Disp}_{\cos}(i) = 1 - (u_i^\top u_i^{edit}). \tag{8}$$

This measures whether the edit induces a nontrivial semantic movement while remaining within the local neighborhood of the original caption.

(3b) Jaccard@K (neighborhood stability). We measure the overlap between the top-K images retrieved by the original caption and its edited variant. Low Jaccard indicates that a small semantic edit induces a large change in the retrieved neighborhood, signaling poor local smoothness in the latent space.

$$\mathrm{Jaccard@K}(i) = \frac{\left|\mathcal{N}_K^I(i) \cap \mathcal{N}_K^{I,\mathrm{edit}}(i)\right|}{\left|\mathcal{N}_K^I(i) \cup \mathcal{N}_K^{I,\mathrm{edit}}(i)\right|},\ \mathrm{Jaccard@K} = \frac{1}{N}\sum_{i=1}^{N} \mathrm{Jaccard@K}(i). \tag{9}$$

(3c) Similarity margin under edits. This metric measures how well the model separates the ground-truth image from its counterfactual distractor. A large positive margin indicates robust discrimination under compositional edits.

$$\Delta_{\mathrm{margin}}(i) = s(v_i, u_i) - s(v_i, u_i^{\mathrm{edit}}). \tag{10}$$

(3d) Brittle fraction (local geometric failures). This metric flags examples with locally inconsistent geometry: either the edited (incorrect) caption is closer to the image than the original ($\Delta_{\mathrm{margin}} \le 0$), or the edited caption still retrieves the target image among its top-K neighbors.

$$\mathrm{Brittle}(i) = \mathbb{I}[\Delta_{\mathrm{margin}}(i) \le 0] \ \vee\ \mathbb{I}\left[I_i \in \mathcal{N}_K^{I,\mathrm{edit}}(i)\right],\ \mathrm{BrittleFrac} = \frac{1}{N}\sum_{i=1}^{N} \mathrm{Brittle}(i). \tag{11}$$

4.5 Key Insights from Geometry Diagnostics

The diagnostics above show that Recall@K alone can hide important changes in the shared image–text geometry. Across Tables 1, 7, 10, and 11, we observe that several text-centric hard-negative methods improve selected retrieval or compositional scores while making the embedding space less balanced. In particular, methods such as NegCLIP, CLoVe, FSC-CLIP, and CE-CLIP often sharpen fine-grained text discrimination, but this can coincide with larger I2T–T2I recall gaps and reduced reciprocity. This suggests that repulsive pressure applied mainly to the text branch can improve one direction of retrieval without equivalently regularizing the visual manifold. Hubness and local edit metrics expose complementary failure modes. Hubness shows that competitive Recall@K can coexist with concentrated nearest-neighbor mass, especially in I2T retrieval, where a small set of text embeddings can become frequent hubs. Local edit stability further reveals brittleness under minimal semantic changes: methods using dense or numerous textual perturbations, such as DAC and CLIC, can show lower neighborhood overlap or higher brittle fractions under relation and attribute swaps, even when average displacement appears reasonable. These diagnostics therefore capture failures that standard retrieval metrics largely miss: unstable neighborhoods, one-sided sharpening, and fragile compositional grounding. The strongest pattern is that image-side constraints matter. TripletCLIP, which introduces both image- and text-side negatives through generative image synthesis, improves reciprocity and reduces hubness relative to purely text-centric approaches, supporting the need for visual near-miss supervision. However, pixel-space counterfactuals are expensive, artifact-prone, and difficult to scale. These findings motivate **LACE**: a lightweight embedding-space approach that supplies image-side and text-side counterfactual supervision, regularizing both modalities while preserving global alignment. We treat geometry metrics as diagnostics rather than causal explanations, but they help distinguish genuine shared-space improvement from one-sided benchmark gains.

Key Insight. Text-centric hard negatives can sharpen benchmark performance while leaving the visual manifold under-constrained. Geometry diagnostics reveal this through reduced reciprocity, increased hubness, and brittle local edits. LACE addresses this by generating lightweight counterfactuals directly in embedding space, supplying fine-grained supervision to both modalities without pixel-level generation.

5 Methodology

We introduce **LACE**, a framework for generating image-side and text-side counterfactuals directly in the shared embedding space of CLIP-style dual encoders. Instead of editing pixels or mining external images, LACE decomposes an image or text embedding into sparse concept atoms, edits a small active support, and uses the resulting counterfactual embeddings as hard negatives during training.

5.1 Sparse Concept Decomposition

Let $\mathbf{v} = f_\theta(I)$ and $\mathbf{u} = g_\phi(T)$ be ℓ_2-normalized image and text embeddings in $\mathbb{R}^d$. LACE assumes that each embedding can be approximated as a sparse mixture of reusable semantic atoms. We construct a fixed concept dictionary $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_K] \in \mathbb{R}^{d \times K}$ from text embeddings of object, attribute, and relation concepts. Given an embedding $\mathbf{z} \in \{\mathbf{v}, \mathbf{u}\}$, we infer sparse coefficients by

$$\alpha^\star(\mathbf{z}; \mathbf{D}) = \arg\min_{\alpha \in \mathbb{R}^K} \|\mathbf{z} - \mathbf{D}\alpha\|_2^2 + \lambda \|\alpha\|_1. \tag{12}$$

The active coefficients identify a small set of candidate concepts that can be locally edited while preserving the surrounding context. To avoid editing spurious atoms, we select editable concepts using both sparse-code magnitude and semantic similarity:

$$\mathcal{K}(\mathbf{z}) = \mathrm{TopP}_k(|\alpha_k^\star(\mathbf{z})|) \cap \mathrm{TopQ}_k(\langle \mathbf{z}, \mathbf{d}_k \rangle). \tag{13}$$

Atoms are optionally bucketed into object, attribute, and relation types to enforce type-consistent edits. Dictionary construction, solver details, and batching are provided in Appendix D.

5.2 Embedding-Space Counterfactual Synthesis

Given a source embedding $\mathbf{z} \in \{\mathbf{v}_i, \mathbf{u}_i\}$, LACE constructs a counterfactual by replacing a source concept $s \in \mathcal{K}(\mathbf{z})$ with a same-type target concept $t \notin \mathcal{K}(\mathbf{z})$. Let $\mathcal{K}_s$ denote the dictionary column(s) associated with the source concept. We form an edited dictionary $\mathbf{D}^{(s \to t)}$ by replacing those columns with the target atom $\mathbf{d}_t$:

$$\mathbf{d}_k^{(s \to t)} = \begin{cases} \mathbf{d}_t, & k \in \mathcal{K}_s, \\ \mathbf{d}_k, & \text{otherwise.} \end{cases} \tag{14}$$

The counterfactual embedding is then reconstructed and renormalized:

$$\mathbf{z}^{cf} = \mathrm{norm}\Big(\mathbf{D}^{(s \to t)} \alpha^\star(\mathbf{z})\Big). \tag{15}$$

For $\mathbf{z} = \mathbf{v}_i$, this yields an image-side counterfactual $\mathbf{v}_{i,k}^{cf}$; for $\mathbf{z} = \mathbf{u}_i$, it yields a text-side counterfactual $\mathbf{u}_{i,k}^{cf}$. Since the edit reuses the original sparse code, its magnitude is controlled by the source activation, producing a targeted near-miss while preserving global context.

5.3 Symmetric Counterfactual Training

For each pair $(\mathbf{v}_i, \mathbf{u}_i)$, LACE generates image-side counterfactuals $\{\mathbf{v}_{i,k}^{cf}\}_{k=1}^{K_{cf}}$ and text-side counterfactuals $\{\mathbf{u}_{i,k}^{cf}\}_{k=1}^{K_{cf}}$ using the same latent editing operator. Text-side counterfactuals act as near-miss negatives for image-to-text retrieval, while image-side counterfactuals act as near-miss negatives for text-to-image retrieval. This yields a symmetric counterfactual objective that pressures both modalities to resolve fine-grained compositional changes.

$$\mathcal{L} = \mathcal{L}_{\mathrm{CLIP}} + \gamma \left(\mathcal{L}_{\mathrm{I2T}}^{cf} + \mathcal{L}_{\mathrm{T2I}}^{cf} \right), \tag{16}$$

where $\mathcal{L}_{\mathrm{CLIP}}$ preserves global image–text alignment, $\mathcal{L}_{\mathrm{I2T}}^{cf}$ contrasts each image against text-side counterfactuals, and $\mathcal{L}_{\mathrm{T2I}}^{cf}$ contrasts each text against image-side counterfactuals. Thus, LACE supplies hard negatives to both retrieval directions. Counterfactual construction is used only during training; inference remains unchanged and requires no dictionary lookup or sparse solver. Full loss expansions, algorithmic details, computational cost, and SigLIP adaptation are provided in App. D.

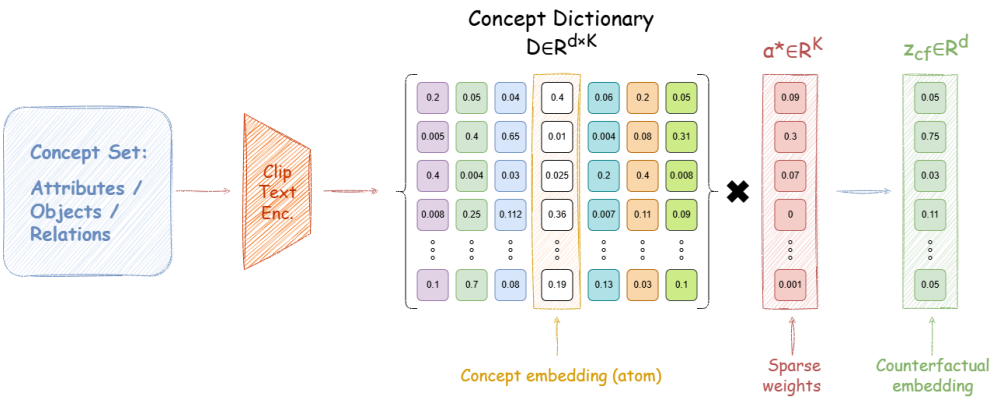

Figure 2: **Modality-agnostic counterfactual generation with sparse concept dictionaries.** LACE builds a fixed dictionary $D \in \mathbb{R}^{d \times K}$ by embedding curated object, attribute, and relation concepts as unit-normalized atoms. Given an image embedding v or text embedding u, it infers sparse concept weights α, edits selected active atoms through attenuation, swaps, or transplants, and reconstructs a counterfactual embedding in the same modality. This shared latent-editing operator enables cheap image-side hard negatives and text-side counterfactuals for symmetric training.

Table 2: **Overall benchmark summary.** We report accuracies (%) on COLA, What'sUp, Winoground, SugarCrepe (Add/Replace/Swap avg), SC++ (Replace/Swap avg), ImageNetV2 Top-1, MMVP-VLM Avg, and **Ret Avg** (mean Recall@1 over I2T/T2I, averaged across MS-COCO and Flickr30k). Best/second-best per column are **bold**/underlined. Baselines are contextual (see Sec. 6.2).

Model	COLA	What'sUp	Wino	SC			SC++		IN-v2	Ret	MMVP
	Avg	Avg	Avg	Add	Replace	Swap	Replace	Swap	Top-1	Avg	Avg
CLIP (ViT-B/32)	22.86	41.77	16.40	72.68	80.02	62.60	68.68	43.07	55.96	54.67	27.42
DAC–LLMs	18.41	42.57	12.50	93.65	**89.33**	74.85	53.63	31.95	45.07	47.35	23.71
CLoVe (w/o patching)	26.51	41.83	12.67	90.35	88.73	**76.15**	64.03	40.35	46.05	56.98	22.97
CE-CLIP	19.68	41.73	12.17	91.90	88.53	75.55	65.73	40.10	43.32	61.35	23.70
FSC-CLIP	17.46	40.97	13.33	90.18	85.06	71.12	64.76	41.31	48.06	56.90	20.01
CLIC	22.22	39.79	**17.92**	86.45	84.73	72.55	76.03	**61.60**	55.30	57.13	19.28
NegCLIP	20.00	42.43	16.33	87.70	82.92	69.98	70.52	56.55	48.92	62.55	25.93
TripletCLIP	17.46	41.03	16.42	88.34	86.97	69.81	73.68	44.04	48.37	53.56	19.99
LACE (Ours)	**39.05**	**43.56**	17.42	**93.96**	86.80	68.17	**76.10**	48.88	**56.66**	**63.08**	**35.56**
Δ *(Ours − CLIP)*	+16.19	+1.79	+1.02	+21.28	+6.78	+5.57	+7.42	+5.81	+0.70	+8.41	+8.14

6 Experiments

We evaluate **LACE** under a controlled CLIP fine-tuning protocol, focusing on three questions:

- **Q1: Compositionality:** Does LACE improve attribute binding and relational reasoning beyond text-side hard-negative baselines?
- **Q2: Geometry:** Do these gains come with healthier shared-space structure, including lower hubness, higher reciprocity, and more stable local edits?
- **Q3: Transfer:** Does LACE preserve or improve standard retrieval and zero-shot transfer without task-specific tuning?

6.1 Experimental setup

We fine-tune OpenCLIP ViT-B/32 with identical initialization and optimization settings across our controlled variants. Training uses MS-COCO image-caption pairs, where LACE injects embedding-space image/text hard negatives into the symmetric contrastive objective (Sec. 5). Geometry diagnostics are computed on a fixed CC3M-50k subset with unit-normalized embeddings, covering reciprocity, hubness, and local edit stability for pretrained and fine-tuned checkpoints. Sec. 6.4 further evaluates SigLIP ViT-B/16 for compatibility with pairwise-sigmoid dual encoders. Details and extended analyses are in Appendix D.

6.2 Baselines and Evaluation Benchmarks

Baselines. We compare CLIP-style variants (ViT-B/32 when available): (i) CLIP zero-shot, (ii) a global-only fine-tune baseline (same setup as ours, without LACE counterfactual supervision), and (iii) NegCLIP (ARO). We also include strong prior baselines when retraining is infeasible (reported numbers): DAC, CLoVe, CE-CLIP, FSC-CLIP, TripletCLIP, and CLIC.

Benchmarks. We evaluate compositionality on COLA [32], What'sUp [1], Winoground [7], and SugarCrepe/SugarCrepe++ [8, 33]. We also report ImageNetV2 top-1, retrieval (mean Recall@1 over MS-COCO and Flickr30k, I2T/T2I), and fine-grained understanding on MMVP-VLM [2].

6.3 Compositional Evaluation

Table 2 summarizes averaged results; per-split breakdowns are deferred to Appendix E.1-E.7.

Q1: Does LACE improve compositional robustness beyond text-side hard negatives? Yes (matched gain: Table 24). LACE yields consistent gains on composition-heavy benchmarks that stress binding and relational sensitivity. In Table 2, LACE improves COLA Avg (decomposed in App. E.5) and What'sUp Avg over the CLIP baseline, and raises Wino Avg (Winoground), indicating fewer failures under near-miss attribute/relation confusions. On SugarCrepe, LACE improves Add/Replace/Swap and remains competitive on SC++ where perturbations are harder and less amenable to template-driven shortcuts. Notably, several text-side HN methods (e.g., NegCLIP, CLoVe, CE-CLIP) can increase specific SugarCrepe sub-scores, but LACE improves multiple compositional benchmarks simultaneously, consistent with a less brittle training signal. Improvements on MMVP-VLM indicate enhanced sensitivity to fine-grained visual patterns, supporting LACE's image-side regularization effect. A matched branch ablation further supports the symmetric design: the average score increases from 45.04 with standard fine-tuning to 45.24 with image-only counterfactuals, 46.37 with text-only counterfactuals, and 47.39 when both branches are used (Appendix F.2).

Q2: Are gains accompanied by healthier multimodal geometry? Yes, with the qualification that geometry metrics diagnose the resulting shared space rather than causally explain individual benchmark gains. LACE improves compositional performance (Table 2) while also producing more balanced neighborhood structure. Compared with text-centric hard-negative baselines, LACE achieves stronger reciprocity, competitive hubness, and fewer brittle failures under controlled semantic edits (Tables 1 and 7), suggesting that the gains are not obtained by sharpening only one retrieval direction or by concentrating nearest-neighbor mass around a small number of hubs. The matched comparison isolates these effects from standard fine-tuning: relative to matched FT, LACE increases Mutual@10 from 46.57% to 51.41% and Edge Reciprocity@10 from 52.46% to 53.79%, while reducing I2T Gini from 0.480 to 0.455 and I2T Top-1% mass from 7.30% to 6.96%. T2I Gini rises slightly (0.417→0.450), but the I2T-T2I Gini gap narrows from 0.063 to 0.005, indicating a more balanced cross-modal structure rather than improvement on every scalar. Locally, LACE reduces relation- and attribute-swap brittle fractions from 3.27%/4.65% to 1.10%/1.60%. Together with the controlled objective ablation in Table 24, these results support the claim that counterfactual supervision improves compositional discrimination, while the geometry diagnostics show that the resulting representation is more reciprocal and less brittle.

Q3: Does LACE preserve general retrieval/transfer? Yes. LACE maintains strong **retrieval** performance and improves the Ret Avg summary metric while also improving IN-v2 Top-1 in Table 2, suggesting the method does not trade off global alignment for compositional gains. This is consistent with our training design, we retain the base CLIP objective as an alignment anchor, while counterfactual terms act as auxiliary regularizers (Sec. 5). We provide per-split results in App. E, analyze Swap-style attribute failures in App. E.3, and report additional core design and causal ablations and diagnostics in App. F. Representative latent-transplant examples are shown in Fig. 3.

6.4 Cross-Objective Validation with SigLIP

To test whether LACE is tied to CLIP ViT-B/32 or softmax InfoNCE, we evaluate it on SigLIP ViT-B/16 with a pairwise-sigmoid objective. Since prior compositional fine-tuning methods are primarily reported on CLIP-style backbones, we use this experiment as a controlled within-backbone validation rather than a direct SOTA comparison. Using the same sparse counterfactual generator and adapting only the loss, LACE improves over both pretrained and fine-tuned SigLIP baselines on every reported metric, including What'sUp, COLA, SugarCrepe++, MMVP-VLM, and the overall average (Tab. 3). In particular, it raises the average score from 45.76 to 49.44, suggesting that LACE

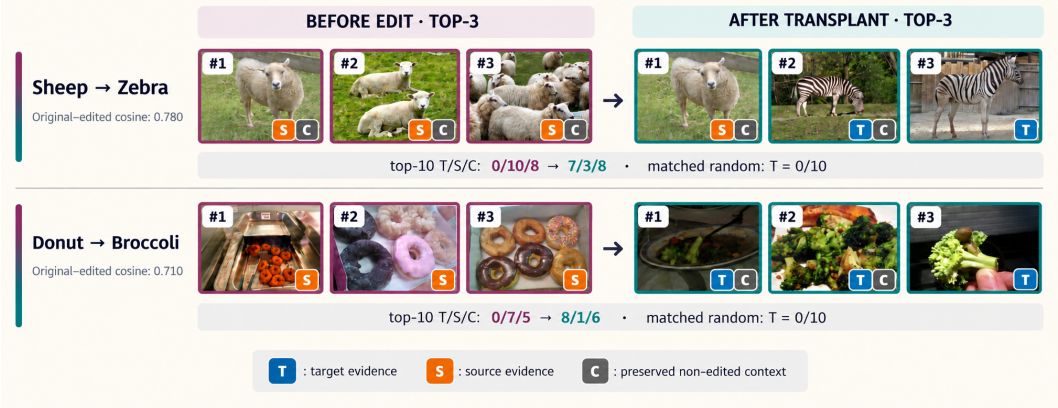

Figure 3: **Nearest-neighbor visualization of successful latent transplants.** For each edit, we show the top-3 COCO validation neighbors before and after transplantation; the statistics below each row are computed over the complete top-10 neighborhood. Target evidence (**T**) replaces source evidence (**S**) while non-edited context (**C**) is largely preserved. A cosine-displacement-matched random perturbation does not retrieve the target concept.

transfers to pairwise-sigmoid dual-encoder training. Further implementation details, design ablations, training-data generalization, and qualitative examples are provided in Appendix D.10, F.2, F.6, and G.

Hyperparameter Sensitivity. Table 4 reports the sensitivity of LACE to the main hyperparameters controlling counterfactual generation and sparse decomposition. The setting $K_{cf} = 0$ or $\gamma = 0$ corresponds to the fine-tuned baseline without counterfactual hard negatives. The default configuration used in the main experiments is $K_{cf} = 4$, $\gamma = 0.5$, $\lambda = 0.10$, and a 14k-atom dictionary. The sensitivity analysis checks whether LACE depends on a narrow hyperparameter choice or remains stable under reasonable perturbations.

Table 3: **LACE with SigLIP.** Average accuracy (%) over compositional benchmarks and MMVP-VLM. ✓ denotes CF hard negatives; bold marks the best result.

Model	CF	What'sUp	COLA	Wino.	SC	SC++	MMVP	Avg.
SigLIP ViT-B/16		44.00	37.46	16.67	78.69	57.35	31.85	44.34
SigLIP FT		44.97	34.89	18.00	79.94	64.92	31.85	45.76
SigLIP FT + CF	✓	**47.00**	**40.90**	**19.00**	**83.98**	**67.95**	**37.78**	**49.44**

Table 4: **Sensitivity of LACE to K_{cf}, loss weight γ, sparse penalty λ, and dictionary size.** We report avg. accuracy (%) over the compositional benchmarks; gray cells denote the default setting.

K_{cf}		γ		λ		Dictionary Size	
Value	**Avg. Acc.**	**Value**	**Avg. Acc.**	**Value**	**Avg. Acc.**	**Atoms**	**Avg. Acc.**
0	45.04	0.00	45.04	0.01	45.99	500	44.73
1	45.30	0.10	45.10	0.05	46.03	1k	44.98
2	46.10	0.25	45.95	0.10	47.39	2k	45.39
4	47.39	0.50	47.39	0.20	47.21	5k	46.00
8	47.36	1.00	47.00	0.50	46.99	14k	47.39

7 Conclusion and Limitations

We introduced **LACE**, a lightweight framework for improving compositional robustness in CLIP-style dual encoders via embedding-space image/text counterfactual supervision. By editing sparse concept decompositions instead of pixels, LACE provides symmetric hard negatives without external generators or inference-time overhead. Across compositional benchmarks, geometry diagnostics, and SigLIP validation, LACE improves binding and relational robustness while preserving retrieval and zero-shot transfer.

Limitations. LACE relies on fixed dictionary coverage and sparse-code quality, which may degrade for rare and abstract concepts. Its latent edits are efficient but approximate visual counterfactuals and do not capture low-level appearance changes. Extending LACE beyond dual encoders to generative multimodal models remains future work.

Acknowledgments and Disclosure of Funding

Harsh Udai acknowledges support from the C-DOT STAR Scholarship Program during this research. The authors declare no competing interests.

References

[1] Amita Kamath, Jack Hessel, and Kai-Wei Chang. What's "up" with vision-language models? investigating their struggle with spatial reasoning, 2023. URL https://arxiv.org/abs/2310.19785.

[2] Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms, 2024. URL https://arxiv.org/abs/2401.06209.

[3] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. URL https://arxiv.org/abs/2103.00020.

[4] Simon Schrodi, David T. Hoffmann, Max Argus, Volker Fischer, and Thomas Brox. Two effects, one trigger: On the modality gap, object bias, and information imbalance in contrastive vision-language models, 2025. URL https://arxiv.org/abs/2404.07983.

[5] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision, 2021. URL https://arxiv.org/abs/2102.05918.

[6] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it?, 2023. URL https://arxiv.org/abs/2210.01936.

[7] Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. Winoground: Probing vision and language models for visio-linguistic compositionality, 2022. URL https://arxiv.org/abs/2204.03162.

[8] Cheng-Yu Hsieh, Jieyu Zhang, Zixian Ma, Aniruddha Kembhavi, and Ranjay Krishna. Sugarcrepe: Fixing hackable benchmarks for vision-language compositionality, 2023. URL https://arxiv.org/abs/2306.14610.

[9] Yufeng Huang, Jiji Tang, Zhuo Chen, Rongsheng Zhang, Xinfeng Zhang, Weijie Chen, Zeng Zhao, Zhou Zhao, Tangjie Lv, Zhipeng Hu, and Wen Zhang. Structure-clip: Towards scene graph knowledge to enhance multi-modal structured representations, 2023. URL https://arxiv.org/abs/2305.06152.

[10] Harman Singh, Pengchuan Zhang, Qifan Wang, Mengjiao Wang, Wenhan Xiong, Jingfei Du, and Yu Chen. Coarse-to-fine contrastive learning in image-text-graph space for improved vision-language compositionality, 2023. URL https://arxiv.org/abs/2305.13812.

[11] Rim Assouel, Pietro Astolfi, Florian Bordes, Michal Drozdzal, and Adriana Romero-Soriano. Object-centric binding in contrastive language-image pretraining, 2025. URL https://arxiv.org/abs/2502.14113.

[12] Jihoon Kwon, Kyle Min, and Jy yong Sohn. Enhancing compositional reasoning in clip via reconstruction and alignment of text descriptions, 2025. URL https://arxiv.org/abs/2510.16540.

[13] Youngtaek Oh, Jae Won Cho, Dong-Jin Kim, In So Kweon, and Junmo Kim. Preserving multi-modal capabilities of pre-trained vlms for improving vision-linguistic compositionality, 2024. URL https://arxiv.org/abs/2410.05210.

[14] Sivan Doveh, Assaf Arbelle, Sivan Harary, Roei Herzig, Donghyun Kim, Paola Cascante-bonilla, Amit Alfassy, Rameswar Panda, Raja Giryes, Rogerio Feris, Shimon Ullman, and Leonid Karlinsky. Dense and aligned captions (dac) promote compositional reasoning in vl models, 2023. URL https://arxiv.org/abs/2305.19595.

[15] Santiago Castro, Amir Ziai, Avneesh Saluja, Zhuoning Yuan, and Rada Mihalcea. Clove: Encoding compositional language in contrastive vision-language models, 2024. URL https://arxiv.org/abs/2402.15021.

[16] Haoxin Li and Boyang Li. Enhancing vision-language compositional understanding with multimodal synthetic data, 2025. URL https://arxiv.org/abs/2503.01167.

[17] Zexi Jia, Chuanwei Huang, Hongyan Fei, Yeshuang Zhu, Zhiqiang Yuan, Ying Deng, Jiapei Zhang, Jinchao Zhang, and Jie Zhou. A visual leap in clip compositionality reasoning through generation of counterfactual sets, 2025. URL https://arxiv.org/abs/2507.04699.

[18] Jianrui Zhang, Mu Cai, Tengyang Xie, and Yong Jae Lee. Countercurate: Enhancing physical and semantic visio-linguistic compositional reasoning via counterfactual examples, 2024. URL https://arxiv.org/abs/2402.13254.

[19] Xin Huang, Ruibin Li, Tong Jia, Wei Zheng, and Ya Wang. Visual perturbation and adaptive hard negative contrastive learning for compositional reasoning in vision-language models, 2025. URL https://arxiv.org/abs/2505.15576.

[20] Matthew Trager, Pramuditha Perera, Luca Zancato, Alessandro Achille, Parminder Bhatia, and Stefano Soatto. Linear spaces of meanings: Compositional structures in vision-language models, 2024. URL https://arxiv.org/abs/2302.14383.

[21] Davide Berasi, Matteo Farina, Massimiliano Mancini, Elisa Ricci, and Nicola Strisciuglio. Not only text: Exploring compositionality of visual representations in vision-language models, 2025. URL https://arxiv.org/abs/2503.17142.

[22] Simion-Vlad Bogolin, Ioana Croitoru, Hailin Jin, Yang Liu, and Samuel Albanie. Cross modal retrieval with querybank normalisation, 2022. URL https://arxiv.org/abs/2112.12777.

[23] Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data, 2018. URL https://arxiv.org/abs/1710.04087.

[24] Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. https://transformer-circuits.pub/2023/monosemantic-features/index.html.

[25] Usha Bhalla, Alex Oesterling, Suraj Srinivas, Flavio P. Calmon, and Himabindu Lakkaraju. Interpreting clip with sparse linear concept embeddings (splice), 2024. URL https://arxiv.org/abs/2402.10376.

[26] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery, 2021. URL https://arxiv.org/abs/2103.17249.

[27] Maitreya Patel, Abhiram Kusumba, Sheng Cheng, Changhoon Kim, Tejas Gokhale, Chitta Baral, and Yezhou Yang. Tripletclip: Improving compositional reasoning of clip via synthetic vision-language negatives, 2024. URL https://arxiv.org/abs/2411.02545.

[28] Ugur Sahin, Hang Li, Qadeer Khan, Daniel Cremers, and Volker Tresp. Enhancing multimodal compositional reasoning of visual language models with generative negative mining, 2023. URL https://arxiv.org/abs/2311.03964.

[29] Le Zhang, Rabiul Awal, and Aishwarya Agrawal. Contrasting intra-modal and ranking cross-modal hard negatives to enhance visio-linguistic compositional understanding, 2024. URL https://arxiv.org/abs/2306.08832.

[30] Amit Peleg, Naman Deep Singh, and Matthias Hein. Advancing compositional awareness in clip with efficient fine-tuning, 2025. URL https://arxiv.org/abs/2505.24424.

[31] Yeyuan Wang, Dehong Gao, Lei Yi, Linbo Jin, Jinxia Zhang, Libin Yang, and Xiaoyan Cai. Enhancing fine-grained vision-language pretraining with negative augmented samples, 2024. URL https://arxiv.org/abs/2412.10029.

[32] Arijit Ray, Filip Radenovic, Abhimanyu Dubey, Bryan A. Plummer, Ranjay Krishna, and Kate Saenko. Cola: A benchmark for compositional text-to-image retrieval, 2023. URL https://arxiv.org/abs/2305.03689.

[33] Sri Harsha Dumpala, Aman Jaiswal, Chandramouli Sastry, Evangelos Milios, Sageev Oore, and Hassan Sajjad. Sugarcrepe++ dataset: Vision-language model sensitivity to semantic and lexical alterations, 2024. URL https://arxiv.org/abs/2406.11171.

[34] Shashank Goel, Hritik Bansal, Sumit Bhatia, Ryan A. Rossi, Vishwa Vinay, and Aditya Grover. Cyclip: Cyclic contrastive language-image pretraining, 2022. URL https://arxiv.org/abs/2205.14459.

[35] Roei Herzig, Alon Mendelson, Leonid Karlinsky, Assaf Arbelle, Rogerio Feris, Trevor Darrell, and Amir Globerson. Incorporating structured representations into pretrained vision & language models using scene graphs, 2023. URL https://arxiv.org/abs/2305.06343.

[36] Tuomas Oikarinen and Tsui-Wei Weng. Clip-dissect: Automatic description of neuron representations in deep vision networks, 2023. URL https://arxiv.org/abs/2204.10965.

[37] Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation, 2022. URL https://arxiv.org/abs/2110.02711.

[38] Amita Kamath, Cheng-Yu Hsieh, Kai-Wei Chang, and Ranjay Krishna. The hard positive truth about vision-language compositionality, 2024. URL https://arxiv.org/abs/2409.17958.

[39] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2818–2829, 2023.

[40] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training, 2023. URL https://arxiv.org/abs/2303.15343.

[41] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015. URL https://arxiv.org/abs/1405.0312.

[42] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of ACL*, 2018.

[43] Yu-Guan Hsieh, Cheng-Yu Hsieh, Shih-Ying Yeh, Louis Béthune, Hadi Pour Ansari, Pavan Kumar Anasosalu Vasu, Chun-Liang Li, Ranjay Krishna, Oncel Tuzel, and Marco Cuturi. Graph-based captioning: Enhancing visual descriptions by interconnecting region captions, 2025. URL https://arxiv.org/abs/2407.06723.

[44] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014. doi: 10.1162/tacl_a_00166. URL https://aclanthology.org/Q14-1006/.

[45] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet?, 2019. URL https://arxiv.org/abs/1902.10811.

[46] Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL `https://arxiv.org/abs/2412.15115`.

[47] Sylvain Gugger, Lysandre Debut, Thomas Wolf, Philipp Schmid, Zachary Mueller, Sourab Mangrulkar, Marc Sun, and Benjamin Bossan. Accelerate: Training and inference at scale made simple, efficient and adaptable. `https://github.com/huggingface/accelerate`, 2022.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction describe LACE as an embedding-space counterfactual supervision method for CLIP-style dual encoders. These claims are supported by the main compositional and retrieval results in Sec. 6 and Table 2, the cross-objective SigLIP validation in Sec. 6.4 and Table 27, and the ablations in Appendix F. The scope is also qualified in Sec. 7, which states that extensions to generative models remain future work.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses limitations in Sec. 7, including reliance on a fixed concept dictionary, dependence on sparse reconstruction quality, the approximate nature of embedding-space counterfactuals, and the current focus on CLIP-style dual encoders rather than generative models. The appendix further reports reconstruction statistics and high-error down-weighting in Appendix D.2, and discusses training-time overhead and inference implications in Appendix D.9.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.

- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not present formal theoretical results, theorems, or proofs. The linear representation hypothesis is used as an operational modeling assumption for constructing embedding-space counterfactuals, not as a theorem introduced or proven in this work. The mathematical expressions in Sec. 5 define the sparse decomposition, counterfactual construction, and training objectives rather than formal theoretical claims.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper specifies the main OpenCLIP ViT-B/32 controlled setup, the MS-COCO fine-tuning data, the CC3M-50k geometry-evaluation split, and the evaluation benchmarks in Sec. 6. The method is fully specified in Sec. 5, including sparse concept decomposition, editable-set selection, counterfactual construction, and the symmetric training objective. The appendix provides dictionary construction details in Appendix D.1, sparse reconstruction statistics in Appendix D.2, sparse solver and batching details in Appendix D.3, training-time compute and LoRA details in Appendix D.9, the SigLIP pairwise-sigmoid adaptation in Appendix D.10, and ablations/hyperparameter sensitivity in Appendix F.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Training and evaluation code, together with instructions for reproducing the main results, is publicly available at `https://github.com/Harsh-Udai/LACE`, including scripts for dictionary construction, sparse-code inference, counterfactual generation, OpenCLIP/SigLIP fine-tuning, and evaluation.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies the main OpenCLIP ViT-B/32 controlled setup, MS-COCO fine-tuning data, CC3M-50k geometry-evaluation split, benchmark suite, and SigLIP compatibility study in Sec. 6. The method and appendix describe dictionary construction, sparse-code inference, editable-set filtering, counterfactual generation, sparse reconstruction statistics, and the SigLIP pairwise-sigmoid adaptation. Training details, including the use of

`Accelerate`, two NVIDIA RTX A6000 GPUs, `bf16` precision, 26 epochs, dataloader per-device batch size 750, gradient accumulation 1, AdamW optimizer, learning rate 1×10^{-5}, weight decay 0.01, 500 warmup steps, cosine schedule, gradient clipping, and LoRA configuration are provided in Appendix D.9. Hyperparameter choices and sensitivity are reported in Appendix F.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: The paper reports single-run results, following the common reporting convention in prior work on these compositional vision-language benchmarks, where full multi-seed fine-tuning is computationally expensive. To improve reproducibility, we use a fixed training seed of 24 and deterministic evaluation under the reported protocols. We support the claims through consistent trends across multiple benchmarks, controlled ablations, robustness analyses, and backbone/loss generalization experiments, rather than through formal confidence intervals or statistical significance tests.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Appendix D.9 reports the compute setup used for the main experiments: distributed data parallel training with Accelerate on two NVIDIA RTX A6000 GPUs, each with 48GB memory, using mixed-precision bf16 training, 26 epochs, dataloader per-device batch size 750, gradient accumulation 1, and effective batch size approximately 1500. The OpenCLIP and SigLIP LACE runs use the same hardware and training schedule as their

corresponding standard fine-tuning baselines, so their total wall-clock time is comparable to standard CLIP/SigLIP fine-tuning under this setup, with an additional counterfactual-construction overhead of approximately 45 seconds to 1 minute per effective batch. LACE adds no inference-time dictionary lookup, sparse coding, or counterfactual construction.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms to the NeurIPS Code of Ethics. The work uses existing vision-language datasets, pretrained models, and evaluation benchmarks for methodological and empirical analysis. It does not involve human-subject experiments, private user data, collection of sensitive personal information, or deployment of systems in high-stakes decision-making settings. We also discuss limitations and scope in Sec. 7 and Appendix D.9.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Our paper proposes LACE, a fine-tuning framework for improving compositional reasoning in CLIP-style vision-language models. Potential positive impacts include more reliable fine-grained visual understanding, better object-attribute binding, and improved relation-aware retrieval. The work does not generate open-ended content, introduce a deployed system, or directly interact with end users. Potential negative impacts are indirect and depend on downstream use, such as stronger vision-language retrieval or recognition systems being applied in sensitive content-analysis, surveillance, or decision-support settings. We discuss scope and limitations in Sec. 7.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to

generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release a high-risk pretrained language model, image generator, scraped dataset, or deployed system. LACE is a fine-tuning framework for CLIP-style dual encoders and does not introduce open-ended generation capabilities. The experiments use existing public datasets, benchmarks, and pretrained checkpoints for methodological evaluation.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper cites the original creators of the datasets, benchmarks, pretrained models, and software packages used in the experiments. Appendix D.11 summarizes the major existing assets, their source or creator, and their license or terms information as reported by the original releases. We do not repackage or redistribute the original datasets, and all assets are used for academic research purposes under their respective release terms.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Our code is publicly available at `https://github.com/Harsh-Udai/LACE` for reproducing the method and experiments. The repository documents scripts for dictionary construction, sparse-code inference, counterfactual generation, OpenCLIP/SigLIP fine-tuning, and evaluation. The paper does not introduce or redistribute a new dataset; the concept dictionary is constructed from existing public data and is documented in Appendix D.1 and Appendix D.12. Existing assets and their licenses are summarized in Appendix D.11.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing experiments, user studies, human-subject research, or collection of new annotations from human participants. All experiments use existing public datasets, benchmarks, and pretrained model checkpoints.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing, user studies, human-subject research, or collection of data from human participants. Therefore, IRB approval or equivalent human-subjects review is not applicable.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: The paper uses Qwen-VL/Qwen2.5-VL during concept extraction for dictionary construction, together with spaCy caption parsing. This usage is part of the LACE method pipeline because the extracted object, attribute, and relation concepts are used to build the fixed concept dictionary. The usage is described in Appendix D.1 and summarized in Appendix D.12.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.

Appendix Roadmap

A. Extended Related Work and Positioning

- **A.1.** Vision-Language Pretraining and Compositional Gaps
- **A.2.** Benchmarks for Compositional Understanding
- **A.3.** Structured Supervision and Object-Centric Representations
- **A.4.** Hard Negatives and Efficient Fine-Tuning
- **A.5.** Geometry of Multimodal Embedding Spaces
- **A.6.** Interpretability, Sparse Decomposition, and Concept Editing
- **A.7.** Positioning LACE

B. Detailed Comparison to Closest Related Work

- **B.1.** Relation to Synthetic Multimodal Counterfactual Data
- **B.2.** Relation to Embedding-Space Visual Perturbation
- **B.3.** Relation to Hard Positives and Oversensitivity
- **B.4.** Relation to Latent Compositional Structure in VLMs
- **B.5.** Summary of Differences

C. Additional Geometry Diagnostics and Results

- **C.1.** Main Local-Edit Stability
- **C.2.** Remaining Global-Geometry Variants
- **C.3.** Remaining Local-Edit Variants
- **C.4.** Retrieval Recall and Modality Gap
- **C.5.** Multi-K Reciprocity Analysis
- **C.6.** Hubness Severity via MaxCount
- **C.7.** Relation-Swap Local Edit Distribution
- **C.8.** Attribute-Swap Local Edit Distribution

D. Additional Method Details (LACE)

- **D.1.** Dictionary Construction and Concept Set
- **D.2.** Sparse Reconstruction Quality and Sample Weighting
- **D.3.** Sparse-Code Inference: Solver and Batching
- **D.4.** Reliable Editable Set via Intersection Filtering
- **D.5.** Choosing Source and Target Concepts
- **D.6.** Neighborhood-Aware Transplant
- **D.7.** Algorithm Summary
- **D.8.** Efficient Dictionary Solving During Training
- **D.9.** Computational Cost and Inference Implications
- **D.10.** LACE with SigLIP-style Pairwise Sigmoid Objectives
- **D.11.** Licenses and Existing Assets
- **D.12.** Model and Data Documentation

E. Extended Results

- **E.1.** What'sUp
- **E.2.** Winoground
- **E.3.** SugarCrepe and Attribute-Swap Analysis
- **E.4.** SugarCrepe++
- **E.5.** COLA
- **E.6.** Zero-shot Classification and Retrieval
- **E.7.** MMVP-VLM

F. Ablations and Diagnostics

- **F.1.** Effect of Counterfactual Hard Negatives
- **F.2.** Design Ablations
- **F.3.** Semantic Specificity of Latent Counterfactuals
- **F.4.** Hyperparameter Sensitivity
- **F.5.** Cross-Objective Validation with SigLIP
- **F.6.** Training-Data Generalization with GBC

G. Qualitative Analysis of Counterfactual Edits

- **G.1.** Qualitative Retrieval Examples
- **G.2.** Nearest-Neighbor Analysis of Latent Edits

H. Broader Impact

A Extended Related Work and Positioning

Vision-Language Pretraining and Compositional Gaps. Contrastive dual-encoder pretraining aligns a global image embedding with a global text embedding using large-scale web supervision, enabling strong zero-shot transfer and retrieval without a task-specific head. This paradigm has become the de facto backbone for open-vocabulary recognition and text-image retrieval, largely because it scales efficiently and admits simple adaptation via prompt engineering or lightweight fine-tuning [3, 5, 34].

However, despite these impressive transfer capabilities, CLIP-style models often struggle with the *structure* of captions: binding attributes to the correct entities, resolving spatial relations (e.g., left/right, on/under), and disambiguating multi-object scenes. A growing body of work demonstrates that these failures are systematic rather than anecdotal, with models relying on coarse semantic matching while remaining brittle to word order, relation flips, and counterfactual distractors [6, 32, 33, 2].

Benchmarks for Compositional Understanding. To make these weaknesses measurable, recent benchmarks isolate compositional phenomena by pairing an image with a correct caption and *near-miss* alternatives. Winoground tests cross-modal consistency under minimal linguistic swaps [7]. SugarCrepe introduces fluent, adversarially refined hard negatives that reduce template biases and benchmark "hackability " [8], while SugarCrepe++ further scales and diversifies such counterfactual evaluation [33]. COLA targets attribute–object binding in retrieval by constructing distractors with identical primitives but incorrect configurations [32]. What'sUp focuses on spatial relations with tightly controlled image sets [1]. Finally, MMVP-VLM probes fine-grained visual pattern sensitivity in CLIP-style matching, highlighting failures that go beyond standard caption distributions [2].

Structured Supervision and Object-Centric Representations. One prominent line of work injects explicit structure into training signals, encouraging models to represent entities and their interactions rather than relying solely on holistic caption embeddings. Scene-graph-based approaches leverage graph-derived substructures to define multi-granular positives, structured hard negatives, or auxiliary alignment objectives, improving attribute binding and relational reasoning [9, 10, 35]. Object-centric formulations similarly introduce inductive biases that align sets of entities or slots across modalities, mitigating information loss from global pooling [11]. While effective, these methods often rely on external parsers or graph extractors, increasing pipeline complexity and sensitivity to annotation noise.

Hard Negatives and Efficient Fine-Tuning. A complementary family of methods preserves the CLIP architecture while strengthening supervision through synthetic or retrieved counterexamples. Early approaches focused on *text-side* perturbations (negations, swaps, attribute flips) due to their low generation cost [6, 29]. Subsequent work introduced auxiliary losses or granular objectives: READ incorporates token-level reconstruction and sentence-level alignment [12], while FSC-CLIP sharpens fine-grained discrimination via local hard negatives and calibrated regularization [13]. Other strategies emphasize data augmentation (e.g., CLIC) [30], LLM-based caption rewriting (e.g., DAC) [14], or explicit compositional encoding in text representations (e.g., CLoVe) [15]. Although effective on specific benchmarks, many of these approaches inject difficulty asymmetrically on the text side to maintain efficiency. Generative approaches such as TripletCLIP have begun to explore synthesized image negatives [28, 27], but high-quality visual counterfactuals remain computationally expensive. This asymmetry motivates our work: when difficulty is concentrated on a single modality, the shared embedding geometry can become skewed.

Geometry of Multimodal Embedding Spaces. Beyond aggregate metrics such as Recall@K, retrieval performance depends critically on the *geometry* of the learned space. Phenomena like hubness and modality gap distort nearest-neighbor structure, degrading cross-modal matching reliability [22]. Post-hoc normalization methods (e.g., QB-Norm) attempt to correct these effects without retraining [22], echoing local scaling techniques such as CSLS from bilingual lexicon induction [23]. Our work directly links these geometric pathologies to the mechanics of hard-negative mining, proposing geometry-aware diagnostics to expose and address training-induced asymmetries.

Interpretability, Sparse Decomposition, and Concept Editing. Parallel research investigates how VLM representations encode semantic concepts and how they can be isolated or manipulated. Interpretability methods range from concept probing and neuron labeling (e.g., CLIP-Dissect) [36] to sparse autoencoders and dictionary learning that decompose dense representations into monosemantic components [24, 25]. On the generative side, CLIP has enabled pixel-space concept manipulation via StyleGAN or diffusion guidance [26, 37]. **LACE** synthesizes these insights but targets a distinct goal: rather than editing pixels or analyzing neurons, it leverages sparse concept decomposition to generate *image-side and text-side counterfactuals directly in embedding space*. These counterfactuals act as efficient, structured hard negatives that serve as an explicit *geometric regularizer*.

Positioning LACE. In contrast to architecturally complex or graph-based approaches, **LACE** requires neither pixel-space generation nor external generative backbones. Unlike text-only hard-negative strategies, it introduces image-side counterfactual supervision while preserving symmetric text-side pressure. By explicitly regularizing the visual manifold during training, **LACE** aims to bridge compositional gaps while restoring a balanced, symmetric geometry in the joint image–text embedding space.

B Detailed Comparison to Closest Related Work

B.1 Relation to Synthetic Multimodal Counterfactual Data

A growing set of methods improves compositionality by augmenting CLIP-style training with counterfactual image–text pairs. These approaches share the central motivation of LACE: standard contrastive supervision is too coarse to teach attribute binding, role assignment, and relational distinctions. They differ primarily in where the counterfactual is produced.

SPARCL. SPARCL [16] constructs synthetic positive and negative image–caption pairs with subtle variations. It identifies three requirements for useful synthetic compositional data: efficient generation, precise text alignment at the edited region, and image fidelity outside the edited region. To address these, SPARCL injects real image features into a fast text-to-image model and applies style transfer, then trains with an adaptive margin loss that downweights likely incorrect generations and emphasizes hard samples. This is closely related to LACE because both methods recognize that uniform treatment of synthetic negatives is brittle and that image-side variation is important. The difference is operational: SPARCL edits or regenerates images, whereas LACE edits normalized embeddings through sparse concept supports. LACE therefore avoids the major bottlenecks SPARCL explicitly identifies: pixel generation cost, imperfect image–caption alignment, and hallucinated or over-modified synthetic samples.

Generation of Counterfactual Sets. GCS [17] builds counterfactual image–text sets using a block-based diffusion strategy. It decomposes an image description into entities, generates local blocks as semantic "puzzle pieces," and recomposes them according to controlled attribute, position, and relation changes. It also introduces a set-level loss that distinguishes intra-set from inter-set samples. GCS is close to LACE in spirit because it treats counterfactuals as structured perturbations of object-level factors rather than arbitrary negative captions. However, GCS performs this control in pixel space using LLM-guided diffusion, while LACE performs it in latent space using sparse concept decomposition. Thus, GCS offers explicit visual counterfactuals, but LACE is cheaper, generator-free, and usable on the fly during CLIP-style training.

CounterCurate. CounterCurate [18] focuses on both semantic and physically grounded compositional reasoning. It shows that even strong multimodal models can fail at left/right, up/down, and counting distinctions, then curates counterfactual data using horizontal flips, grounded inpainting, and strong generative models. This work is important for positioning LACE because it supports the claim that image-side counterfactuals matter: physical reasoning cannot be fixed reliably by text-only perturbations. LACE differs by replacing curated pixel-level counterfactual pipelines with embedding-level counterfactual synthesis. CounterCurate is therefore evidence for the need; LACE is a scalable mechanism for delivering a related supervision signal without image generation.

B.2 Relation to Embedding-Space Visual Perturbation

AHNPL. AHNPL [19] is arguably the closest methodological neighbor. It observes that many hard-negative methods operate only on text, leaving the image encoder undertrained for fine-grained distinctions. AHNPL computes a semantic shift between a positive caption and a text hard negative, then adds that shift to the original image embedding to create a visual hard negative. It further introduces multimodal hard-negative losses and dynamic margins.

LACE shares AHNPL's premise that visual-side negatives can be produced without pixel editing. The key difference is the structure of the edit. AHNPL uses a dense displacement vector derived from text embeddings, effectively translating a text-side semantic difference into image space. LACE instead decomposes both image and text embeddings over a sparse dictionary of concept atoms and edits a small active support. This gives LACE more explicit control over which concept is attenuated, swapped, or transplanted; supports type-aware edits over objects, attributes, and relations; and uses the same sparse editing operator for both modalities. In short, AHNPL demonstrates the promise of embedding-space visual perturbation, while LACE makes the perturbation factorized, concept-local, and modality-symmetric.

Empirical comparison. A fully controlled comparison is difficult because AHNPL and LACE use different CLIP backbones. Nevertheless, AHNPL's reported plain-CLIP SugarCrepe baseline is within 0.2 points of ours, making its published results informative as a contextual comparison. AHNPL obtains $95.9/87.4/76.1$ on the Add/Replace/Swap subsets, compared with $93.96/86.80/68.17$ for LACE. Thus, AHNPL is stronger on SugarCrepe, particularly on Swap. We therefore do not claim that LACE uniformly dominates AHNPL on forced-choice compositional benchmarks. The main distinction is that LACE produces more consistent improvements across compositional discrimination, multi-object binding, and cross-modal retrieval.

Official-code reproduction. Because AHNPL does not provide a trained checkpoint, we retrained it using the authors' official implementation and reported protocol, OpenAI CLIP ViT-B/32 initialization followed by training on MS-COCO. This differs from the OpenCLIP-LAION initialization used by LACE and is therefore not a strictly backbone-matched comparison. Our reproduction closely matches AHNPL published ARO results, obtaining 84.26 versus 83.8 on VG-Relation and 76.10 versus 77.0 on VG-Attribution, but does not fully recover its reported VALSE and SugarCrepe performance. We consequently use AHNPL published values for benchmark comparisons and use the reproduced checkpoint only for analyses requiring access to its embeddings.

Relative to its own zero-shot backbone, the reproduced AHNPL checkpoint improves What'sUp from 43.40 to 46.00, SugarCrepe from 75.50 to 80.63, SugarCrepe++ from 57.20 to 59.55, and MMVP from 15.56 to 17.04. However, COLA decreases from 22.86 to 11.75 and Winoground decreases from 17.00 to 12.95. In contrast, LACE improves all six benchmarks over its own backbone, COLA increases from 22.86 to 39.05, while improvements on the remaining benchmarks range from 1.0 to 10.6 percentage points. These results suggest that gains on forced-choice discrimination do not necessarily transfer to multi-object binding and cross-modal consistency.

Common-protocol geometry. We additionally evaluate the available checkpoints on the same ordered 50,000 CC3M image-caption pairs with unit-normalized embeddings and $K = 10$. AHNPL rows use AHNPL's loading code (OpenCLIP, OpenAI weights), so cross-row values are approximate.

Table 5: **Common-protocol retrieval geometry on CC3M-50k.** Higher recall and mutual retrieval are better, lower Gini coefficients and maximum retrieval counts indicate weaker hub concentration.

Model	I2T R@10 ↑	T2I R@10 ↑	Mutual@10 ↑	I2T Gini ↓	T2I Gini ↓	I2T Max ↓	T2I Max ↓
AHNPL backbone	44.12	44.13	36.09	0.577	0.544	423	284
AHNPL reproduced	17.58	21.98	12.54	0.819	0.627	1737	3960
LACE	**57.79**	**56.90**	**51.41**	**0.455**	**0.450**	**134**	**128**

The reproduced AHNPL checkpoint (Table 5) exhibits lower retrieval and reciprocity together with a substantially stronger hub concentration than its initialization. This behavior is consistent with representation contraction, its mean matched image-text cosine rises from approximately 0.29 to

0.81, but the mean cosine among distinct image embeddings also rises to approximately 0.93, while image effective rank decreases from roughly $200/512$ to $87/512$. LACE shows higher retrieval and lower hub concentration, subject to this caveat. Because the AHNPL checkpoint is reproduced and the backbones differ, this analysis should not be interpreted as a definitive controlled ranking. Rather, it suggests that improvements on selected forced-choice benchmarks can coexist with contraction, weak cross-modal reciprocity, and extreme retrieval hubs.

B.3 Relation to Hard Positives and Oversensitivity

The Hard Positive Truth. Hard Positive Truth [38] shows that hard-negative fine-tuning can overstate compositional gains because existing evaluations often ignore hard positives: semantics-preserving caption variants that should remain aligned with the same image. The paper finds that hard-negative-only training can make CLIP oversensitive, improving separation from negatives while incorrectly suppressing valid positive paraphrases. This directly motivates a caution in LACE: counterfactual supervision must sharpen genuine semantic distinctions without globally warping the embedding space or making every perturbation look negative.

LACE addresses this concern in two ways. First, edits are localized to sparse concept supports rather than applied as broad repulsive pressure on the entire text branch. Second, LACE evaluates geometry through hubness, reciprocity, and local edit stability, making it harder to hide oversensitivity behind benchmark-only improvements. The relation is complementary: Hard Positive Truth identifies a failure mode of hard-negative training; LACE proposes a counterfactual mechanism and diagnostic suite intended to avoid similar brittleness.

B.4 Relation to Latent Compositional Structure in VLMs

Linear Spaces of Meanings. Linear Spaces of Meanings [20] studies whether CLIP text embeddings of composite concepts can be approximated as sums of factor vectors, or "ideal words." It gives both geometric and probabilistic interpretations of decomposable embeddings and shows that simple linear operations can support classification, retrieval, debiasing, and representation manipulation. LACE uses a compatible intuition: semantic factors can be manipulated in embedding space. However, LACE is not merely analyzing whether such structure exists. It uses a sparse concept dictionary to infer active factors for each image or text embedding, then synthesizes counterfactual negatives for training. Thus, Linear Spaces provides theoretical and empirical support for latent compositional structure; LACE operationalizes a related idea as a scalable training signal.

Not Only Text / GDE. GDE [21] extends the study of compositional structure to visual embeddings. It argues that visual representations are noisier and sparser than text representations, and that normalized CLIP embeddings should be treated with manifold-aware, geodesic structure rather than purely Euclidean linearity. This is highly relevant to LACE because LACE edits image embeddings directly. The distinction is again purpose: GDE is primarily an analysis and representation framework for visual compositionality, while LACE uses embedding decomposition to generate counterfactual supervision. GDE also suggests a possible extension of LACE: replacing or augmenting Euclidean sparse edits with geodesic or tangent-space edits on the hypersphere.

B.5 Summary of Differences

Table 6 compares LACE with the most closely related methods.

C Additional Geometry Diagnostics and Results

This appendix reports additional geometry diagnostics on CC3M-50k. These tables complement the main geometry results in Sec. 4.5 by showing omitted method variants, recall asymmetry, reciprocity at multiple neighborhood sizes, maximum hubness counts, and full local-edit distributions.

C.1 Main Local-Edit Stability

Figure 1 summarizes local edit behavior using average displacement and brittle failure rate. Here we provide the full underlying statistics for relation and attribute swaps, including neighborhood

Table 6: **Comparison between LACE and closely related work.** Prior methods either rely on pixel-space counterfactual generation, remain primarily text-side, or analyze latent compositional structure without using it as training-time supervision. LACE moves counterfactual construction into the shared embedding space, enabling scalable image-side and text-side hard negatives without external generators.

Work	Counterfactual domain	Image-side negatives	Pixel generation/editing	Main limitation relative to LACE
SPARCL [16]	Images + captions	Yes	Yes	Costly generation; residual alignment noise
GCS [17]	Diffusion counterfactual sets	Yes	Yes	Requires LLM/diffusion pipeline
CounterCurate [18]	Curated physical + semantic pairs	Yes	Yes / curated	Offline and generator/benchmark specific
AHNPL [19]	Embedding perturbation	Yes	No	Dense text-shift transfer; less factor-local
Hard Positive Truth [38]	Text positives/negatives	Mostly no	No	Diagnoses oversensitivity; not a generator
Linear Spaces [20]	Text embedding decomposition	No	No	Diagnostic/manipulative, not training-time HNs
GDE [21]	Visual embedding decomposition	No	No	Analyzes geometry; does not synthesize HNs
LACE	Sparse image/text embedding edits	Yes	No	Generator-free, symmetric, concept-local training

overlap via Jaccard@10. These results show that low movement magnitude alone is insufficient: several methods with modest Δ mean still exhibit high brittle failure rates, while LACE maintains low brittleness under both edit types.

Table 7: **Local edit stability on CC3M-50k ($k_{\text{loc}} = 10$).** We report (i) mean change in the edit metric (Δ mean; lower is better, $\downarrow$), (ii) neighborhood overlap (Jaccard@10, $\uparrow$), and (iii) failure rate (BrittleFrac, $\downarrow$), for relation and attribute swaps. All checkpoints are evaluated using the same CC3M-50k local-edit protocol. Matched FT and LACE share the same initialization, fine-tuning data, optimizer, and training schedule, only LACE includes counterfactual supervision. Image-only and text-only CF branch variants are reported in App. F.2.

Method	Relation Swap			Attribute Swap		
	Δ mean $\downarrow$	Jaccard@10 $\uparrow$	BrittleFrac $\downarrow$	Δ mean $\downarrow$	Jaccard@10 $\uparrow$	BrittleFrac $\downarrow$
CLIP (base)	0.0023	0.699	0.015	0.0314	0.339	0.032
FSC-CLIP (laion+coco)	0.0100	0.661	0.022	0.0193	0.338	0.242
CLoVe (alpha)	0.0110	0.605	0.033	0.0293	0.326	0.053
DAC (LLM)	0.0343	0.537	0.072	0.0334	0.304	0.140
CLIC (coco)	0.0036	0.676	0.026	0.0184	0.366	0.065
CE-CLIP	0.0055	0.641	0.058	0.0100	0.415	0.270
NegCLIP	0.0049	0.616	0.041	0.0217	0.315	0.070
TripletCLIP	0.0030	0.696	0.028	0.0109	0.486	0.172
Matched FT	0.0040	0.659	0.033	0.0356	0.287	0.047
LACE (Ours)	0.0032	0.667	0.011	0.0192	0.297	0.016

Comparison with matched fine-tuning. The matched comparison shows (Table 7) that the local-geometry improvements are not explained by standard fine-tuning alone. Relative to matched FT, LACE reduces the relation-swap brittle fraction from 3.27% to 1.10% and the attribute-swap brittle fraction from 4.65% to 1.60%, corresponding to relative reductions of approximately 66% in both cases. Mean displacement decreases from 0.0040 to 0.0032 for relation swaps and from 0.0356 to 0.0192 for attribute swaps, while Jaccard@10 increases from 0.659 to 0.667 and from 0.287 to 0.297, respectively. Thus, LACE reduces local failures without destabilizing the surrounding retrieval neighborhood. As these probes use the swap edits LACE trains on, we treat them as supporting evidence; our main geometry conclusions rest on Table 1.

C.2 Remaining Global-Geometry Variants

Table 8 reports global geometry diagnostics for method variants omitted from the main table after selecting the strongest checkpoint per method family. The metrics include I2T/T2I hubness and reciprocity at $K = 10$.

C.3 Remaining Local-Edit Variants

Table 9 reports local edit stability for the remaining method variants under relation and attribute swaps. These results complement the main local-edit table by showing whether omitted checkpoints exhibit similar brittleness patterns.

Table 8: **Remaining method variants (global geometry on CC3M-50k).** These are the variants omitted from the main table after selecting the stronger checkpoint per family. Top-1% mass is reported in %, reciprocity is mutual/edge at K=10 (in %).

Method	I2T		T2I		Reciprocity ($K = 10$)	
	Gini ↓	Top-1% ↓	Gini ↓	Top-1% ↓	Mutual ↑	Edge ↑
FSC-CLIP (cc3m)	0.74	14.26	0.61	10.48	26.03	32.96
CLoVe (w/o patching)	0.72	12.37	0.62	16.41	30.56	33.00
DAC (LLM)	0.83	19.30	0.54	11.99	25.50	26.90
CLIC (redcaps)	0.54	8.65	0.54	8.74	37.98	43.18

Table 9: **Remaining method variants (local edit stability on CC3M-50k).** These are the variants omitted from the main local-edit table after selecting the stronger checkpoint per family. We report relation-swap and attribute-swap stability at k_{loc}=10. **Cat.** and **BF** denote Catapult and Brittle Fraction metrics, respectively.

Method	Relation swap				Attribute swap			
	Δ mean ↓	Jaccard@10 ↑	Cat. ↓	BF ↓	Δ mean ↓	Jaccard@10 ↑	Cat. ↓	BF ↓
FSC-CLIP (cc3m)	0.0084	0.672	0.0132	0.023	0.0180	0.336	0.0554	0.264
CLoVe (w/o patching)	0.0276	0.503	0.0670	0.042	0.0321	0.304	0.1034	0.130
DAC (LLM)	0.0343	0.537	0.0401	0.072	0.0334	0.304	0.0957	0.140
CLIC (redcaps)	0.0040	0.661	0.0183	0.029	0.0167	0.398	0.0793	0.094

C.4 Retrieval Recall and Modality Gap

Table 10 reports directional Recall@K for I2T and T2I retrieval, together with the signed recall gap. This table quantifies whether a method improves retrieval symmetrically or mainly sharpens one direction.

Table 10: **Retrieval recall and modality gap on CC3M-50k.** We report Recall@K for image→text (I2T) and text→image (T2I), and the recall gap (I2T−T2I) in percentage points. All values are in %. **LACE (Ours)** is evaluated using the same checkpoint and CC3M-50k split as Tables 1 and 7.

Method	K=1			K=5			K=10		
	I2T ↑	T2I ↑	Gap ↓	I2T ↑	T2I ↑	Gap ↓	I2T ↑	T2I ↑	Gap ↓
CLIP (base)	29.41	27.72	1.69	49.27	47.34	1.93	57.39	55.72	1.67
FSC-CLIP	13.54	16.70	−3.16	28.29	33.37	−5.07	36.30	41.60	−5.30
CLoVe	23.58	23.63	−0.05	42.63	42.92	−0.29	51.30	51.54	−0.23
DAC	10.39	17.68	−7.29	22.10	34.81	−12.70	28.79	43.51	−14.72
CLIC	19.28	19.47	−0.19	37.31	38.19	−0.88	45.66	47.08	−1.42
READ-CLIP	13.60	16.67	−3.07	28.27	33.79	−5.53	35.77	41.96	−6.19
CE-CLIP	6.78	11.56	−4.79	16.18	24.64	−8.46	21.62	31.64	−10.02
NegCLIP	17.30	17.93	−0.63	33.60	35.39	−1.78	41.54	43.88	−2.35
TripletCLIP	23.43	25.64	−2.22	43.43	48.10	−4.67	52.83	58.03	−5.20
LACE (Ours)	29.34	28.53	0.81	49.41	48.57	0.84	57.79	56.90	0.88

C.5 Multi-K Reciprocity Analysis

Table 11 reports mutual and edge reciprocity at $K \in \{1, 5, 10\}$. This checks whether improvements in neighborhood symmetry persist across different neighborhood sizes rather than only at the main-table setting.

C.6 Hubness Severity via MaxCount

Table 12 reports the maximum retrieval count for each method. Unlike Gini or Top-1% mass, MaxCount directly exposes extreme super-hubs that dominate many nearest-neighbor lists.

Table 11: **Reciprocity on CC3M-50k at multiple neighborhood sizes.** We report *mutual reciprocity* and *edge reciprocity* (both in %) for KNN graphs at K$\in \{1, 5, 10\}$. Higher is better. **LACE (Ours)** is evaluated using the same checkpoint and CC3M-50k split as Tables 1 and 7.

Method	K=1		K=5		K=10	
	Mutual ↑	Edge ↑	Mutual ↑	Edge ↑	Mutual ↑	Edge ↑
CLIP (base)	21.15	38.24	41.73	49.35	50.30	52.18
FSC-CLIP	8.50	20.10	22.51	31.34	30.36	35.95
CLoVe	16.30	32.72	35.96	46.43	45.23	51.17
DAC	6.47	14.46	17.32	22.76	24.07	26.47
CLIC	12.10	27.67	29.88	41.38	38.77	46.45
READ-CLIP	8.31	21.39	22.43	33.40	30.04	37.97
CE-CLIP	3.80	11.25	11.87	19.23	17.16	23.01
NegCLIP	11.25	26.03	27.43	39.43	35.74	44.48
TripletCLIP	15.51	28.75	36.59	43.82	46.90	50.48
LACE (Ours)	21.51	39.21	42.74	50.74	51.41	53.79

Table 12: **Hubness severity via maximum retrieval count (CC3M-50k, K=10). MaxCount** is the maximum number of times a single item appears in the top-10 retrieval lists (higher indicates stronger hubness).

Method	I2T MaxCount ↓	T2I MaxCount ↓
CLIP (base)	710	357
FSC-CLIP	347	671
CLoVe	173	203
DAC	528	198
CLIC	270	251
READ-CLIP	389	195
CE-CLIP	896	5080
NegCLIP	360	169
TripletCLIP	127	121
LACE (Ours)	134	128

C.7 Relation-Swap Local Edit Distribution

Table 13 reports the full local-edit distribution for relation swaps, including mean, median, percentiles, neighborhood overlap, catapult distance, and brittle fraction.

Table 13: **Local edit distribution on CC3M-50k (Relation swap, k_loc=10).** We report summary statistics of the edit metric change Δ (mean/median and 10th/90th percentiles), neighborhood overlap (Jaccard@10 mean/median), catapult distance (mean), and failure rate (BrittleFrac). **Cat.** and **BF** denote Catapult and Brittle Fraction metrics, respectively.

Method	M	Δ mean ↓	Δ median ↓	Δ p10 ↓	Δ p90 ↓	Jac. mean ↑	Jac. median ↑	Cat. mean ↓	BF ↓
CLIP (base)	14092	0.00	0.00	−0.01	0.01	0.70	0.67	0.02	0.02
FSC-CLIP	14092	0.01	0.01	0.00	0.02	0.66	0.67	0.02	0.02
CLoVe	14092	0.01	0.01	0.00	0.03	0.61	0.67	0.04	0.03
DAC	14092	0.03	0.03	0.00	0.06	0.53	0.54	0.04	0.09
CLIC	14092	0.00	0.00	−0.01	0.01	0.68	0.67	0.02	0.03
READ-CLIP	14092	0.01	0.01	0.00	0.02	0.65	0.67	0.01	0.04
CE-CLIP	14092	0.01	0.00	0.00	0.02	0.64	0.67	0.00	0.06
NegCLIP	14092	0.00	0.00	−0.01	0.02	0.62	0.67	0.02	0.04
TripletCLIP	14092	0.00	0.00	0.00	0.01	0.70	0.67	0.01	0.03
LACE (Ours)	14092	0.00	0.00	0.00	0.01	0.67	0.67	0.01	0.01

C.8 Attribute-Swap Local Edit Distribution

Table 14 reports the full local-edit distribution for attribute swaps. This split is important because LACE is comparatively weaker on some attribute-swap settings, so the distributional view helps make the failure mode explicit.

Table 14: **Local edit distribution on CC3M-50k (Attribute swap, $k_{\text{loc}} = 10$).** We report summary statistics of the edit metric change Δ (mean/median and 10th/90th percentiles), neighborhood overlap (Jaccard@10 mean/median), catapult distance (mean), and failure rate (BrittleFrac). **Cat.**, **Jac.** and **BF** denote Catapult, Jaccard and Brittle Fraction metrics, respectively.

Method	M	Δ mean ↓	Δ median ↓	Δ p10 ↓	Δ p90 ↓	Jac. mean ↑	Jac. median ↑	Cat. mean ↓	BF ↓
CLIP (base)	7082	0.03	0.03	-0.00	0.07	0.34	0.33	0.13	0.03
FSC-CLIP	7082	0.02	0.02	-0.00	0.05	0.34	0.33	0.06	0.24
CLoVe	7082	0.03	0.03	-0.00	0.07	0.33	0.33	0.12	0.05
DAC	7082	0.03	0.03	-0.00	0.06	0.30	0.25	0.09	0.15
CLIC	7082	0.02	0.02	-0.00	0.05	0.37	0.33	0.10	0.07
READ-CLIP	7082	0.02	0.02	-0.00	0.04	0.37	0.33	0.02	0.30
CE-CLIP	7082	0.01	0.01	-0.00	0.03	0.42	0.43	0.01	0.27
NegCLIP	7082	0.02	0.02	-0.00	0.05	0.32	0.25	0.11	0.07
TripletCLIP	7082	0.01	0.01	-0.00	0.03	0.49	0.54	0.03	0.17
LACE (Ours)	7082	0.02	0.02	-0.00	0.04	0.30	0.25	0.07	0.02

D Additional Method Details (LACE)

We follow the notation and core formulation from Sec. 5. This appendix provides (i) implementation details for sparse coding and counterfactual construction, (ii) practical heuristics for robust concept selection, and (iii) an end-to-end algorithmic summary.

D.1 Dictionary Construction and Concept Set

LACE uses a single fixed concept dictionary $D = [d_1, \ldots, d_K] \in \mathbb{R}^{d \times K}$ whose columns are unit-normalized CLIP text embeddings of natural-language concept strings. For the OpenCLIP ViT-B/32 experiments, the embedding dimension is $d = 512$ and the dictionary contains $K = 14{,}000$ atoms: 5,200 object concepts, 5,800 attribute concepts, and approximately 3,000 relation concepts.

Concept extraction. We construct the dictionary from the MS-COCO training split used for fine-tuning. For each image–caption pair, we extract candidate object, attribute, and relation phrases from two sources. First, we parse the human caption with spaCy to obtain noun phrases, adjectival modifiers, and dependency relations. Second, we query a Qwen vision-language model on the image to produce additional visual concept descriptions, including objects, visible attributes, and pairwise/spatial relations. The two sources are merged to improve coverage: caption parsing provides text-grounded concepts, while the VLM description recovers visually salient concepts that may be omitted from the caption.

Type assignment. Each candidate concept is assigned to one of three type buckets: object, attribute, or relation. Object atoms correspond to entity nouns or noun phrases (e.g., "dog", "traffic light", "tennis racket"); attribute atoms correspond to visual properties such as color, material, size, texture, or state (e.g., "red", "wooden", "small", "wet"); relation atoms correspond to spatial or interaction predicates (e.g., "on top of", "next to", "holding", "behind"). These type labels are used only to constrain counterfactual edits so that LACE swaps concepts within the same semantic bucket.

Filtering and deduplication. We lowercase candidate strings, remove punctuation-only strings and very rare malformed phrases, and canonicalize simple variants when possible. Each remaining concept string c_k is embedded with the frozen CLIP text encoder:

$$d_k = \mathrm{norm}(g_\phi(c_k)).$$

To reduce redundancy, we remove near-duplicate candidates whose CLIP-text cosine similarity exceeds 0.9, retaining the more frequent concept in the MS-COCO extraction pool. After filtering, we retain 14,000 atoms with the object/attribute/relation split reported above.

Fixed dictionary, sample-specific sparse codes. The dictionary is built once before training and remains fixed for all experiments. It is shared by both modalities when constructing counterfactual embeddings. For a given image or text embedding z, LACE infers a sample-specific sparse code $\alpha^\star(z; D)$ over this fixed vocabulary. Thus, D is a global concept vocabulary, whereas the active support of $\alpha^\star$ is different for each training example. This distinction is important: LACE does not learn a new dictionary during fine-tuning, and it introduces no trainable dictionary parameters or inference-time components.

D.2 Sparse Reconstruction Quality and Sample Weighting

Because LACE edits embeddings through sparse reconstructions, we measure how well the fixed dictionary represents training examples. For each normalized image embedding v_i in the MS-COCO training set, we compute the sparse reconstruction $\hat{v}_i = \mathrm{norm}(D\alpha^\star(v_i; D))$ and report the reconstruction error

$$e_i = \|v_i - \hat{v}_i\|_2.$$

With the default sparsity weight $\lambda = 0.1$, the mean reconstruction error is 0.31 with standard deviation 0.09. Approximately 6.2% of training images have reconstruction error above 0.55. We treat these high-error examples as lower-confidence counterfactual sources, since poor sparse reconstruction can produce less reliable concept edits.

Down-weighting high-error counterfactuals. For examples with $e_i > 0.55$, we down-weight the corresponding counterfactual loss terms rather than discarding the samples entirely. Concretely, we use a confidence weight

$$w_i = \begin{cases} 1, & e_i \le 0.55, \\ \rho, & e_i > 0.55, \end{cases}$$

where $\rho < 1$ is a fixed down-weighting factor. The weighted counterfactual objective is

$$\mathcal{L}^{cf} = \frac{1}{N} \sum_{i=1}^{N} w_i \left(\mathcal{L}^{cf}_{i,\mathrm{I2T}} + \mathcal{L}^{cf}_{i,\mathrm{T2I}} \right).$$

Here $\mathcal{L}^{cf}_{i,\mathrm{I2T}}$ ($\mathcal{L}^{cf}_{i,\mathrm{T2I}}$) is the InfoNCE loss of Eq. 1 for sample i with its K_{cf} text-side (image-side) counterfactuals added to the denominator.

The standard CLIP loss is not down-weighted; the confidence weight applies only to counterfactual terms. This keeps all original image–caption pairs in the global alignment objective while reducing the influence of unreliable latent edits.

Interpretation. LACE does not require perfect reconstruction. The counterfactual edit relies primarily on support stability: the active atoms should identify semantically meaningful factors whose replacement yields a local near-miss embedding. The reconstruction statistics above therefore serve as a sanity check that the fixed dictionary provides sufficient coverage for most training examples, while the high-error down-weighting mitigates failures caused by rare, abstract, or poorly covered concepts.

D.3 Sparse-Code Inference: Solver and Batching

LACE requires solving the LASSO sparse coding problem (Eq. 12) for each embedding $\mathbf{x}$ (for each modality embedding $z \in \{v, u\}$). Since $\mathbf{D}$ is fixed, inference is amortized by caching $\mathbf{D}$ on device and solving per-sample codes in parallel within each minibatch. We use a first-order solver (FISTA or coordinate descent) with a small fixed iteration budget, which is sufficient for stable supports in practice and adds modest overhead relative to the forward pass.

Practical settings. We use (i) a fixed number of solver iterations, (ii) a fixed sparsity weight λ, and (iii) optional top-K preselection by cosine similarity to reduce compute when K is large. The key requirement for LACE is *support stability* rather than perfect reconstruction.

Table 15: **LACE dictionary and sparse reconstruction statistics** for the OpenCLIP ViT-B/32 experiments. The dictionary is fixed throughout training. Reconstruction error is measured on MS-COCO training image embeddings using the default sparsity weight $\lambda = 0.1$.

Statistic	**Value**
Embedding dimension d	512
Total dictionary atoms K	14,000
Object atoms	5,200
Attribute atoms	5,800
Relation atoms	$\sim$3,000
Duplicate-removal threshold	CLIP-text cosine > 0.9
Mean reconstruction error	0.31
Std. reconstruction error	0.09
High-error threshold	0.55
Fraction above threshold	6.2%

D.4 Reliable Editable Set via Intersection Filtering

$$S(z) = \mathrm{TopP}_k(|\alpha_k^\star(z)|), \tag{17}$$

$$C(z) = \mathrm{TopQ}_k(\langle z, d_k \rangle), \tag{18}$$

$$\mathcal{K}(z) = S(z) \cap C(z). \tag{19}$$

Atom selection exhibits a consistent *sparsity–alignment trade-off*: atoms with large $|\alpha_k^\star|$ reconstruct well but can be idiosyncratic due to dictionary redundancy, while high-cosine atoms are semantically plausible but may not be used by the sparse solution. To obtain a robust editable set, we intersect both signals (Eq. 19) and optionally rank the retained indices by a joint score such as

$$score(k) = |\alpha_k^\star(z)| \cdot \langle z, d_k \rangle \tag{20}$$

which favors atoms that are both *structurally active* and *semantically aligned*.

Type-consistent edits. We bucket candidate concepts into {object, attribute, relation} using a lightweight parser (e.g., spaCy). This enables type-consistent swaps (attribute→attribute, relation→relation), reducing degenerate counterfactuals.

D.5 Choosing Source and Target Concepts

Given $\mathcal{K}(\mathbf{z})$, LACE selects a *source* concept s from the editable set and a *target* concept t of the same type bucket. We use the following constraints:

- **Source present:** prefer s that is supported by $\mathcal{K}(\mathbf{z})$ and appears in the caption concept set (when available).
- **Target absent:** choose $t \notin S(\mathbf{z})$ (not in sparse support) and absent from the caption concept set, to form a true counterfactual.
- **Type match:** t must match the source bucket (object/attribute/relation).

Among valid same-type target concepts, we sample uniformly at random at each training step to increase counterfactual diversity. These rules reduce "false counterfactuals" where the target concept is already implicitly active.

D.6 Neighborhood-Aware Transplant

In addition to replacing the source column(s), LACE replaces a small neighborhood of dictionary atoms that are closest to the source concept in dictionary space. Let $\mathcal{K}_s$ denote the index (or indices) for the source concept and let $\mathcal{N}_m(s)$ be the m nearest dictionary atoms to $\mathbf{d}_s$ (by cosine). We then replace $\mathcal{K}_s \cup \mathcal{N}_m(s)$ with the target embedding $\mathbf{d}_t$ when forming $\mathbf{D}^{(s \to t)}$. This produces stronger but still localized edits. We use neighborhood size $m = 2$, so each transplant replaces the source atom and up to two nearest same-type neighbors. Thus, each counterfactual typically edits 1–3

dictionary columns. Attenuation instead reduces the source coefficients $\alpha_k^\star$ ($k \in \mathcal{K}_s$) while keeping D unchanged, and swap moves the source activation onto the target atom's coefficient, also keeping D unchanged. Full LACE uses counterfactuals from all three operators.

D.7 Algorithm Summary

Algorithm 1 LACE: Embedding-space image/text-side counterfactual hard negatives

Require: Pretrained CLIP encoders f_θ, g_ϕ; concept dictionary $\mathbf{D} \in \mathbb{R}^{d\times K}$; batch $\{(I_i, T_i)\}_{i=1}^N$; sparsity λ; TopP/TopQ; K_{cf}.

Compute embeddings: $\mathbf{v}_i \leftarrow \mathrm{norm}(f_\theta(I_i))$, $\mathbf{u}_i \leftarrow \mathrm{norm}(g_\phi(T_i))$
for each sample $i \in \{1, \dots, N\}$ **do**
  for each modality $m \in \{I, T\}$ **do**
    Set source embedding $\mathbf{z}_i^m \leftarrow \begin{cases} \mathbf{v}_i, & m = I, \\ \mathbf{u}_i, & m = T. \end{cases}$
    Sparse-code $\mathbf{z}_i^m$: $\alpha_{i,m}^\star \leftarrow \arg\min_\alpha \ \|\mathbf{z}_i^m - \mathbf{D}\alpha\|_2^2 + \lambda\|\alpha\|_1$ (Eq. 12)
    Support sets: $S_i^m \leftarrow \mathrm{TopP}(|\alpha_{i,m,k}^\star|)$; $C_i^m \leftarrow \mathrm{TopQ}(\langle \mathbf{z}_i^m, \mathbf{d}_k \rangle)$
    Editable set: $\mathcal{K}_i^m \leftarrow S_i^m \cap C_i^m$ (Eq. 19)
    Bucket $\mathcal{K}_i^m$ into {obj,attr,rel} for type-consistent edits
    for $k = 1, \dots, K_{\mathrm{cf}}$ **do**
      Choose source $s_{i,k}^m \in \mathcal{K}_i^m$ and target $t_{i,k}^m$ with same type, $t_{i,k}^m \notin S_i^m$, and $t_{i,k}^m \notin \mathrm{Concepts}(T_i)$
      Form edited dictionary $\mathbf{D}^{(s_{i,k}^m \to t_{i,k}^m)}$ by replacing the source atom and its m nearest same-type neighbors (Eq. 14)
      Construct counterfactual: $\mathbf{z}_{i,k}^{m,\mathrm{cf}} \leftarrow \mathrm{norm}\Big(\mathbf{D}^{(s_{i,k}^m \to t_{i,k}^m)} \alpha_{i,m}^\star\Big)$ (Eq. 15)
    end for
  end for
end for
Use image-side counterfactuals $\{\mathbf{z}_{i,k}^{I,\mathrm{cf}}\}_{k=1}^{K_{\mathrm{cf}}}$ as $\{\mathbf{v}_{i,k}^{\mathrm{cf}}\}_{k=1}^{K_{\mathrm{cf}}}$ in the T2I denominator
Use text-side counterfactuals $\{\mathbf{z}_{i,k}^{T,\mathrm{cf}}\}_{k=1}^{K_{\mathrm{cf}}}$ as $\{\mathbf{u}_{i,k}^{\mathrm{cf}}\}_{k=1}^{K_{\mathrm{cf}}}$ in the I2T denominator
Optimize the final objective in Eq. 16.

D.8 Efficient Dictionary Solving During Training

LACE relies on sparse decomposition over a fixed concept dictionary to construct counterfactual image and text embeddings. While naïvely solving a sparse coding problem per sample would be prohibitive, we design the procedure to be efficient and compatible with large-scale contrastive training.

First, dictionary solving is performed *only during training* and operates entirely in embedding space, incurring no additional cost at inference time. Given a modality embedding $z \in \{v_i, u_i\}$, we restrict optimization to a candidate set of dictionary atoms selected via fast cosine similarity prefiltering, substantially reducing problem dimensionality. Sparse coefficients are then obtained using a small, fixed number of lightweight optimization steps, which we treat as a deterministic, non-differentiable operator. Gradients are not backpropagated through the solver, avoiding instability and overhead.

Second, counterfactual embeddings are computed on-the-fly and reused across the image-side contrastive loss as well as on the text-side, amortizing solver cost within each minibatch. Because the dictionary is fixed and shared across training, the decomposition is highly cache-friendly and parallelizable across GPUs.

Together, these design choices allow LACE to scale dictionary-based counterfactual supervision to hundreds of thousands of training samples with modest overhead relative to standard CLIP fine-tuning, while retaining precise control over the semantics of the induced counterfactual edits.

D.9 Computational Cost and Inference Implications

LACE introduces additional computation only during training, arising from dictionary-based counterfactual construction on both the image and text embedding spaces. This overhead is controlled and does not affect inference: at test time, no dictionary lookup, sparse coding, or counterfactual construction is performed.

Training-time cost. Sparse decomposition is applied during training to both image and text embeddings. The same fixed dictionary and editing rules are used in both cases. The image-side counterfactuals are the main scalability advantage of LACE, since comparable visual hard negatives would otherwise require pixel-space editing, generation, or retrieval mining. Text-side counterfactuals are included to preserve symmetric fine-grained pressure in the contrastive objective. As described in Appendix D.8, dictionary solving operates on a reduced candidate set obtained via fast similarity prefiltering and uses a fixed, small number of optimization steps without backpropagation through the solver. Dictionary solving is performed symmetrically on both modalities and amortized across losses within each minibatch.

In our implementation, training is performed with distributed data parallelism using `Accelerate` on two NVIDIA RTX A6000 GPUs, each with 48GB memory, using mixed-precision `bf16` training. We train for 26 epochs with dataloader per-device batch size 750, gradient accumulation set to 1, and an effective batch size of approximately 1500. Optimization uses AdamW with learning rate 1×10^{-5}, weight decay 0.01, 500 warmup steps, a cosine learning-rate schedule with cosine cycle factor 0.5, and gradient clipping with maximum gradient norm 1.0. We adapt both the vision and text encoders with LoRA, applying LoRA to the query, key, value, and output projection layers in both encoders, using rank 16, LoRA scaling $\alpha = 32$, and dropout 0.1. The SigLIP compatibility experiment uses the same optimizer, LoRA configuration, and training schedule unless otherwise stated.

Under this setup, counterfactual construction introduces an additional *45 seconds to 1 minute per effective batch* relative to standard CLIP fine-tuning, representing a controlled constant-factor increase in training time. This overhead is substantially lower than pixel-space generation or multi-stage hard-negative mining pipelines commonly used to improve compositional robustness.

Inference-time cost. LACE introduces *no counterfactual-generation module* at inference. At test time, the model reduces to the fine-tuned CLIP-style dual encoder, producing a single image and text embedding per input. No dictionary access, sparse coding, or counterfactual construction is required. When LoRA fine-tuning is used, the deployed model contains the corresponding LoRA-adapted weights, or equivalently the merged LoRA weights if adapters are folded into the backbone. Thus, LACE does not add any inference-time computation beyond the selected fine-tuning mechanism.

Parameter and memory footprint. The LACE counterfactual generator itself has no trainable parameters: the concept dictionary is fixed and shared across training, sparse coding is treated as an external optimization operator, and no optimizer state is maintained for the dictionary or solver. In our experiments, the CLIP/SigLIP encoders are adapted with LoRA, so the fine-tuned checkpoints include the corresponding LoRA adapter parameters. These parameters belong to the underlying fine-tuning scheme rather than to the LACE counterfactual construction mechanism. During training, the fixed dictionary is cached on GPU and incurs only transient memory usage for counterfactual construction. In the OpenCLIP ViT-B/32 setting, the dictionary has size $512 \times 14{,}000$, which is small relative to encoder activations and optimizer state.

Overall, LACE trades a controlled increase in training-time computation for improved compositional robustness and embedding geometry, while preserving the simplicity and deployment characteristics of standard CLIP-style inference. LACE is therefore architecture-preserving and inference-free with respect to counterfactual generation: it changes the training objective through latent counterfactuals, but adds no trainable dictionary, solver, or counterfactual module at inference.

D.10 LACE with SigLIP-style Pairwise Sigmoid Objectives

The main LACE objective in Sec. 5.3 is written for CLIP-style softmax InfoNCE training, where counterfactual hard negatives are added to the image-to-text and text-to-image partition functions. SigLIP-style models use a different contrastive formulation: instead of normalizing over a batch with a softmax denominator, they optimize a pairwise sigmoid loss over positive and negative image–text pairs. This requires a small but important adaptation of the LACE training objective.

Base pairwise sigmoid objective. Let v_i and u_j denote normalized image and text embeddings. We write the pairwise logit as

$$a_{ij} = s(v_i, u_j),$$

where $s(\cdot, \cdot)$ denotes the scaled image–text similarity used by the corresponding SigLIP checkpoint. In practice, this may include the model's learned logit scale and bias. Let

$$\ell(a, y) = \log(1 + \exp(-ya))$$

be the binary logistic loss with label $y \in \{+1, -1\}$. The standard SigLIP-style batch objective can be written as

$$\mathcal{L}_{\text{SigLIP}} = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \ell(a_{ij}, y_{ij}), \quad y_{ij} = \begin{cases} +1, & i = j, \\ -1, & i \neq j. \end{cases}$$

Thus, matched image–caption pairs are positives and mismatched in-batch pairs are negatives.

Counterfactuals as explicit negative pairs. For each positive pair (v_i, u_i), LACE generates image-side counterfactuals $\{v_{i,k}^{cf}\}_{k=1}^{K_{cf}}$ and text-side counterfactuals $\{u_{i,k}^{cf}\}_{k=1}^{K_{cf}}$ using the same sparse concept-editing operator defined in Sec. 5. In the SigLIP setting, these counterfactuals are not inserted into a softmax partition function. Instead, they are added as explicit negative pairs with label -1:

$$(v_i, u_{i,k}^{cf}) \quad \text{and} \quad (v_{i,k}^{cf}, u_i).$$

The first pair asks the model to keep the original image closer to its true caption than to a text-side counterfactual caption embedding. The second pair asks the model to keep the original caption closer to the true image than to an image-side counterfactual image embedding. This mirrors the two directional hard-negative roles used in the InfoNCE formulation, but implements them through pairwise logistic supervision.

The LACE counterfactual loss for SigLIP is

$$\mathcal{L}_{\text{SigLIP}}^{cf} = \frac{1}{NK_{cf}} \sum_{i=1}^{N} \sum_{k=1}^{K_{cf}} \left[\ell\Big(s(v_i, u_{i,k}^{cf}), -1\Big) + \ell\Big(s(v_{i,k}^{cf}, u_i), -1\Big) \right].$$

The final objective is

$$\mathcal{L} = \mathcal{L}_{\text{SigLIP}} + \gamma \mathcal{L}_{\text{SigLIP}}^{cf}.$$

Relation to the InfoNCE formulation. The InfoNCE version of LACE and the SigLIP version share the same counterfactual generator; they differ only in how the generated counterfactuals enter the loss. For OpenCLIP-style softmax training, the counterfactual embeddings augment the retrieval denominators. For SigLIP-style sigmoid training, they become additional negative pairs. In both cases, the intended supervision is the same: the original image–caption pair should remain close, while minimally edited counterfactual near-misses should be pushed away.

Dictionary construction for SigLIP. For the SigLIP experiments, we rebuild the fixed concept dictionary using the SigLIP text encoder rather than reusing the OpenCLIP dictionary. This ensures that all concept atoms lie in the same embedding space as the SigLIP image and text representations. The concept strings, type buckets, filtering rules, and object/attribute/relation construction pipeline are kept the same as in Appendix D.1; only the encoder used to embed each concept string is changed.

Implementation details. During SigLIP fine-tuning, we first compute the standard pairwise sigmoid loss over the minibatch. We then generate LACE counterfactual embeddings for each positive pair using the same sparse decomposition, editable-set filtering, and concept transplant steps used in the OpenCLIP experiments. The resulting counterfactual pairs are assigned label -1 and added to the loss with weight γ. Gradients are not backpropagated through the sparse solver, matching the OpenCLIP implementation.

Interpretation. This adaptation tests whether LACE is merely a modification of the CLIP softmax denominator or a more general counterfactual supervision mechanism for dual encoders. The SigLIP results in Table 27 support the latter: counterfactual hard negatives improve over both pretrained SigLIP and standard SigLIP fine-tuning, indicating that sparse latent counterfactual supervision transfers across contrastive objective families.

D.11 Licenses and Existing Assets

We use publicly available datasets, pretrained models, codebases, and evaluation benchmarks. Table 16 summarizes the primary existing assets used in this work, together with their source and license or terms information as reported by the original releases. We use all assets for academic research purposes and follow their original release terms. We do not repackage or redistribute the original datasets; users should obtain each dataset or checkpoint from its official source and follow the corresponding license and terms of use. For datasets whose images are collected from external sources, the annotations or benchmark metadata may have a separate license while the underlying images remain governed by their original source licenses.

Table 16: **Existing assets used in LACE.** We summarize the main datasets, benchmarks, pretrained models, and codebases used in this work, along with their source and license or terms information as reported by the original releases.

Asset	Source / Creator	License / Terms
OpenCLIP codebase	OpenCLIP [39]	MIT license.
OpenCLIP ViT-B/32 pretrained weights	OpenCLIP / LAION [39]	Follow original OpenCLIP model-card and checkpoint release terms.
SigLIP ViT-B/16 pretrained weights	SigLIP / Google [40]	Follow original SigLIP model-card and checkpoint release terms.
MS-COCO	COCO [41]	COCO annotations are CC BY 4.0; images follow original source terms.
CC3M	Conceptual Captions [42]	Follow original dataset release terms; images follow original source terms.
Graph-Based Captioning (GBC)	GBC [43]	Follow original dataset/model release terms.
Flickr30k	Flickr30k [44]	Apache 2.0 license, according to the public release used.
ImageNetV2	ImageNetV2 [45]	MIT license.
What'sUp	What'sUp [1]	MIT license.
COLA	COLA [32]	MIT license.
Winoground	Winoground [7]	Meta Images Research License; non-commercial research/evaluation use only.
SugarCrepe	SugarCrepe [8]	MIT license.
SugarCrepe++	SugarCrepe++ [33]	MIT license.
MMVP-VLM	MMVP-VLM [2]	MIT license.
Qwen-VL / Qwen2.5-VL	Qwen team, Alibaba Cloud [46]	Apache 2.0 license, according to the public release used.
spaCy	spaCy / Explosion AI	MIT license.
HuggingFace Accelerate	HuggingFace [47]	Apache 2.0 license.
PyTorch	PyTorch	BSD-style license.

D.12 Model and Data Documentation

Model card summary.

- **Base models:** Our main controlled experiments use OpenCLIP ViT-B/32. We additionally evaluate SigLIP ViT-B/16 for cross-objective compatibility with pairwise-sigmoid dual-encoder training.
- **Adaptation:** We fine-tune the vision and text encoders using LoRA adapters. LoRA is applied to the query, key, value, and output projection layers in both encoders, with rank 16, LoRA scaling $\alpha = 32$, and dropout 0.1.
- **Training objective:** For OpenCLIP, training uses the standard symmetric CLIP contrastive objective plus LACE image-side and text-side counterfactual hard-negative losses. Counterfactual text embeddings augment the image-to-text denominator, and counterfactual image embeddings augment the text-to-image denominator. For SigLIP, LACE counterfactuals are added as explicit negative image–text pairs with label -1 in the pairwise sigmoid objective.

- **Counterfactual generator:** LACE constructs counterfactual embeddings through sparse decomposition over a fixed concept dictionary of object, attribute, and relation atoms. The dictionary is fixed during training; sparse coding is treated as an external operator, and no gradients are backpropagated through the sparse solver.
- **Inference:** Inference is unchanged relative to the underlying CLIP-style dual encoder. No dictionary access, sparse coding, or counterfactual construction is used at test time. When LoRA is used, the deployed model contains the LoRA-adapted weights or the corresponding merged weights.
- **Intended use:** Research on compositional robustness, fine-grained image–text retrieval, and geometry-aware analysis of CLIP-style vision-language models.
- **Limitations:** Performance depends on the coverage and quality of the fixed concept dictionary and on the reliability of sparse reconstruction. Embedding-space counterfactuals are efficient and controllable but approximate visual counterfactuals rather than pixel-level edits. The method is evaluated on CLIP-style dual encoders; extension to generative or autoregressive multimodal models remains future work.

Data card summary.

- **Training data:** The main OpenCLIP experiments fine-tune on MS-COCO image–caption pairs. The concept dictionary is also constructed from the MS-COCO training split using caption parsing and visual concept extraction.
- **Additional training-data generalization:** Appendix F.6 additionally evaluates LACE using Graph-Based Captioning (GBC) image–text data as an alternative fine-tuning source.
- **Concept extraction:** Candidate object, attribute, and relation concepts are extracted from MS-COCO captions using spaCy parsing and from images using Qwen-VL/Qwen2.5-VL visual descriptions. Candidate strings are filtered, type-bucketed into object/attribute/relation categories, embedded with the corresponding text encoder, unit-normalized, and deduplicated using a cosine similarity threshold.
- **Dictionary statistics:** In the OpenCLIP ViT-B/32 setting, the dictionary has dimension $512 \times 14{,}000$, containing 5,200 object atoms, 5,800 attribute atoms, and approximately 3,000 relation atoms. For the SigLIP experiment, the same concept strings and filtering rules are used, but atoms are re-embedded with the SigLIP text encoder so that the dictionary lies in the SigLIP embedding space.
- **Geometry evaluation data:** Geometry diagnostics are computed on a fixed CC3M-50k image–text subset using unit-normalized embeddings. We report hubness, reciprocity, retrieval asymmetry, and local edit stability on this subset.
- **Evaluation data:** We evaluate on What'sUp, COLA, Winoground, SugarCrepe, SugarCrepe++, ImageNetV2, MS-COCO/Flickr30k retrieval, and MMVP-VLM.
- **Licenses:** The primary datasets, benchmarks, pretrained models, and software packages used in this work are summarized in Table 16. We use all assets for research purposes under their respective licenses and release terms.

E Extended Results

E.1 Scores on What'sUp benchmark

Table 17 reports detailed results on What'sUp, and we include **Avg(All)** as a single-number summary across all six evaluation splits. As expected, *fine-tuning directly on COCO/GQA-Spatial* attains the highest Avg(All), since its training distribution closely matches the evaluation splits. Our goal is different: **LACE** targets *compositional generalization without* training on COCO/GQA-Spatial, by injecting **image-side and text-side counterfactual hard negatives** synthesized in the shared embedding space.

Concretely, LACE expresses an image embedding as a sparse combination of dictionary atoms and applies *localized counterfactual edits* (e.g., swapping/transplanting relation- or attribute-relevant components while preserving the remaining context). This yields minimally-changed but semantically

Table 17: **What'sUp.** Spatial/relational reasoning with controlled caption flips (e.g., above/below). Scores are accuracies (%). We also report **Avg(All)** = mean across all six splits (What'sUp A/B, COCO-spatial one/two, GQA-spatial one/two). Best/second-best per column (excluding random chance) are **bold**/underlined.

Model	What'sUp Subset A	What'sUp Subset B	COCO-spatial One-obj.	COCO-spatial Two-obj.	GQA-spatial One-obj.	GQA-spatial Two-obj.	Avg (All)
CLIP ViT-B/32	30.3	31.6	43.7	51.1	46.5	47.4	41.77
FT on train COCO, GQA-spatial	28.2	25.2	**67.2**	**60.7**	**64.4**	54.6	**50.05**
FT on LAION-4M-prep	31.6	34.6	43.1	48.9	44.3	50.9	42.23
FT on LAION-4M-prep + neg. cap.	32.0	26.5	39.9	48.9	47.3	45.7	40.05
DAC–LLMs	28.40	34.56	42.10	51.14	48.71	50.52	42.57
DAC–SAM	27.18	27.21	47.66	54.09	46.81	51.20	42.36
CLoVe (w/o patching)	**34.71**	24.75	45.39	51.82	46.90	47.42	41.83
CE-CLIP	31.07	30.39	42.01	52.05	47.07	47.77	41.73
FSC-CLIP (CC3M)	29.85	29.41	42.01	51.36	46.81	46.39	40.97
CLIC-RedCaps	26.21	29.66	41.12	48.18	45.78	47.77	39.79
NegCLIP	32.5	**36.3**	47.4	46.4	45.3	46.7	42.43
TripletCLIP	26.9	32.6	46.5	51.5	56.65	32.0	41.03
LACE (Ours)	32.28	27.94	45.93	52.76	46.43	**56.03**	43.56

targeted negatives, avoiding pixel-space artifacts and reducing reliance on template-style caption corruption, while providing a more modality-symmetric supervisory signal.

Consistent with this design, **LACE** achieves a strong **Avg(All)** (43.56) among the non-oracle baselines, and shows its clearest gains on the most interference-heavy setting, **GQA-spatial Two-obj.** (**56.03**). This pattern aligns with LACE's intent: counterfactual supervision is most beneficial when the model must preserve object identity while resolving subtle relational flips in multi-object scenes. Meanwhile, the What'sUp A/B splits remain relatively close across methods, reflecting the tight ceiling of controlled caption-flip evaluation when training lacks broad relational diversity. Overall, these results suggest that **structure-preserving image-side counterfactuals** particularly strengthen robustness in multi-object relational settings, while maintaining competitive performance across the full benchmark suite.

E.2 Scores on Winoground benchmark

Table 18: **Winoground results.** We report text, image, group scores (%) and their mean as **Avg.**. Higher is better. Best/second-best per column are **bold**/underlined.

Model	Text	Image	Group	Avg.
CLIP	30.70	10.50	8.00	16.40
DAC–LLMs	22.50	10.25	4.75	12.50
DAC–SAM	20.75	**12.25**	8.00	13.67
CLoVe (w/o patching)	22.00	9.50	6.50	12.67
CE-CLIP	19.25	12.00	5.25	12.17
FSC-CLIP (CC3M)	25.50	8.75	5.75	13.33
CLIC-RedCaps	31.25	12.00	**10.50**	**17.92**
NegCLIP	30.00	11.00	8.00	16.33
TripletCLIP	30.00	11.00	8.25	16.42
LACE (Ours)	**33.25**	10.25	8.75	17.42

Winoground evaluates challenging compositional disambiguation through paired image–caption setups. Using the same CLIP-like backbone and evaluation protocol, we report accuracy on the text, image, and group subtasks, along with the overall average (Table 18). **LACE** achieves the **best Text score** (33.25%) and the **second-best overall average** (17.42%), with a competitive Group score (8.75%). Notably, these gains are obtained **without mined or pixel-generated hard negatives**,

suggesting that LACE improves fine-grained cross-modal binding and compositional disambiguation under the Winoground setup rather than overfitting to perturbation templates.

E.3 Detailed Scores on SugarCrepe benchmark

Table 19: **SugarCrepe (SC) detailed results.** We report category-wise accuracies (%). Higher is better. Best/second-best per column are **bold**/underlined.

Model	Add Obj	Add Att	Replace Obj	Replace Att	Replace Rel	Swap Obj	Swap Att
CLIP	77.2	68.2	90.9	80.0	69.1	61.2	64.0
DAC–LLMs	89.6	**97.7**	94.4	89.2	**84.4**	**75.5**	74.2
DAC–SAM	87.6	95.2	91.2	85.8	83.9	71.8	74.6
CLoVe (w/o patching)	86.0	94.7	93.1	**91.2**	81.9	74.7	**77.6**
CE-CLIP	92.2	91.6	94.1	90.4	81.1	74.7	76.4
FSC-CLIP (CC3M)	87.1	93.2	92.4	85.8	77.0	70.6	71.6
CLIC-RedCaps	84.2	88.7	93.3	86.9	74.0	71.8	73.3
NegCLIP	86.95	88.44	91.53	83.25	73.97	72.24	67.72
TripletCLIP	90.40	86.27	94.43	85.53	80.94	69.80	69.82
LACE (Ours)	**95.88**	92.05	**95.10**	86.80	78.52	66.53	69.82

Table 19 reports detailed SugarCrepe accuracies under the Add/Replace/Swap perturbation protocol. **LACE** is strongest on *object-centric* robustness: it achieves the **best** performance on **Add-Obj** (95.88%) and **Replace-Obj** (95.10%), indicating improved sensitivity to object identity changes beyond standard CLIP pretraining. On attribute and relation edits, however, the best results are attained by other structured baselines (e.g., **Add-Att**: DAC–LLMs 97.7%; **Replace-Att**: CLoVe 91.2%; **Replace-Rel**: DAC–LLMs 84.4%), and LACE is comparatively lower on **Swap** splits. Overall, these trends suggest LACE's gains are concentrated in correcting *which object* is present/replaced, while more challenging *binding* stress tests (attribute swaps and relation replacements) remain a harder regime for this training recipe.

Attribute-swap failure analysis. LACE improves most consistently on object-centric Add/Replace perturbations, but is weaker on some Swap-style evaluations, especially attribute swaps. We view this as an informative failure mode rather than a contradiction of the method. Object concepts often correspond to relatively stable directions in the shared embedding space: replacing "dog" with "cat" or adding/removing an object usually induces a localized semantic change. Attributes are less cleanly factorized. Some attributes, such as color, can behave like reusable directions, but many others are object- and context-dependent: "wooden table", "wooden floor", and "wooden toy" do not correspond to the same visual transformation, and attributes such as size, material, texture, and state are often entangled with object identity and scene context.

This explains the pattern in the SugarCrepe and SugarCrepe++ splits: LACE is strongest when the counterfactual edit changes which object is present or replaced, but remains less reliable when the benchmark requires preserving object identity while swapping fine-grained attributes across objects. The local-edit diagnostics are consistent with this interpretation. LACE has low brittle failure rates under attribute edits, indicating that the model is not simply becoming globally oversensitive to all attribute perturbations, but the lower swap accuracy suggests that the current dictionary does not always provide sufficiently disentangled attribute directions for binding-sensitive edits. A natural extension is to augment the dictionary with object-conditioned attribute atoms, such as "red suitcase" or "wooden table", rather than relying only on atomic attribute concepts.

E.4 Scores on SugarCrepe++ benchmark

SugarCrepe++ extends SugarCrepe by pairing each image with two semantically correct but lexically diverse captions, along with a strong hard-negative caption. The benchmark tests whether both positives consistently rank above the negative under compositional perturbations (Table 20). **LACE** achieves the **best** performance on **Replace-Obj** (91.52%) and is second-best on the harder **Replace-Att** (74.5%) and **Replace-Rel** (62.30%) splits, indicating improved robustness when positives vary paraphrastically but remain semantically consistent. However, LACE is weaker on the **Swap** splits

(Swap-Obj/Swap-Att), where **CLIC-RedCaps** and **NegCLIP** lead by a clear margin, suggesting that maintaining correct cross-caption alignment under token-level swapping remains a more challenging regime for LACE's training recipe.

Table 20: **SugarCrepe++ (SC++) results.** We report accuracies (%) for Replace (Obj/Att/Rel) and Swap (Obj/Att). Higher is better. Best/second-best per column are **bold**/underlined.

Model	Replace Obj	Replace Att	Replace Rel	Swap Obj	Swap Att
CLIP	87.89	68.02	50.14	37.95	48.19
DAC–LLMs	65.7	47.6	47.6	31.0	32.9
DAC–SAM	64.3	43.9	48.5	27.8	33.2
CLoVe (w/o patching)	73.7	63.3	55.1	35.5	45.2
CE-CLIP	81.8	62.4	53.0	40.0	40.2
FSC-CLIP (CC3M)	85.4	61.9	46.9	41.6	41.0
CLIC-RedCaps	90.1	**75.1**	**62.9**	**60.8**	**62.4**
NegCLIP	89.58	69.54	52.43	54.69	58.40
TripletCLIP	86.98	71.82	62.23	39.59	48.49
LACE (Ours)	**91.52**	74.5	62.30	46.12	51.65

E.5 Detailed COLA Multi-Object Results

In the main paper, we report an *avg-only* summary of Cola to keep the presentation compact. Here we provide the full **Cola Multi-Object** breakdown in Table 21, reporting joint accuracies for **T2I**, **I2T**, and **GROUP (AND)**. **LACE** improves *consistently and substantially* across all three criteria, achieving the **best** scores on **T2I** (32.86%), **I2T** (56.19%), and **GROUP** (28.10%). Notably, the gain is especially large on **I2T** and **GROUP**, indicating stronger multi-object binding and fewer failures under same-part distractors, rather than merely improving one retrieval direction.

Table 21: **COLA multi-object.** We report T2I, I2T, and GROUP accuracies (%, higher is better). Best/second-best per column are **bold**/underlined.

Model	T2I	I2T	GROUP
CLIP	20.95	33.33	14.29
DAC–LLMs	24.29	19.52	11.43
DAC–SAM	16.67	21.90	8.10
CLoVe (w/o patching)	29.05	35.24	15.24
CE-CLIP	23.33	24.76	10.95
FSC-CLIP (CC3M)	16.19	27.14	9.05
CLIC-RedCaps	19.05	36.19	11.43
NegCLIP	16.19	32.38	11.43
TripletCLIP	10.95	32.38	9.05
LACE (Ours)	**32.86**	**56.19**	**28.10**

The pretrained-to-final COLA improvement should not be attributed entirely to LACE. Under matched initialization, training data, LoRA configuration, optimizer, and training schedule, standard MS-COCO fine-tuning increases the COLA score from 22.86 to 35.56, a gain of 12.70 points. Adding LACE further increases the score by 3.49 points to 39.05. A similar trend holds with GBC training, standard fine-tuning increases COLA from 22.86 to 27.90, and adding LACE further improves it to 32.00. The benefit of LACE is also positive across all six benchmarks under both training sources.

We additionally evaluate on a verified image-disjoint COLA subset to control for possible overlap between Visual Genome and MS-COCO. The improvement from LACE remains positive at $+2.60$ points. Because the matched fine-tuned baseline and LACE use identical training data, overlap may influence their absolute scores but cannot by itself explain this incremental gain.

The comparatively strong COLA result is likely related to task alignment. COLA distractors preserve many of the same objects and attributes while altering their bindings, closely matching the localized near-miss counterfactuals generated by LACE. Consistent with this interpretation, the modality

ablation yields 35.56 for standard fine-tuning, 34.70 with image-only counterfactuals, 37.19 with text-only counterfactuals, and 39.05 with symmetric image-text counterfactuals. The full gain therefore emerges when both modalities receive binding-sensitive counterfactual supervision.

E.6 Zero-shot Classification and Retrieval

Table 22 reports standard zero-shot transfer on ImageNetV2 (Top-1/Top-5) and bidirectional retrieval (Recall@1) on MS-COCO and Flickr30k. **LACE** achieves the **best** ImageNetV2 **Top-1** (56.66%) and remains competitive on **Top-5** (83.33%, second-best). For retrieval, LACE delivers the **best** performance on **MS-COCO T2I R@1** (49.62%) and **Flickr30k T2I R@1** (71.22%), while maintaining strong I2T results (second-best on Flickr30k I2T at 79.7%). Overall, these results indicate that LACE strengthens global image–text alignment (notably in the harder T2I direction) while preserving, and in some cases improving, recognition-style transfer.

Table 22: **Zero-shot classification and retrieval.** We report Recall@1 (%) on MS-COCO and Flickr30k for image-to-text (I2T) and text-to-image (T2I), and ImageNetV2 zero-shot Top-1/Top-5 (%). Higher is better. Best/second-best per column are **bold**/underlined.

Model	MS-COCO I2T R@1	MS-COCO T2I R@1	Flickr30k I2T R@1	Flickr30k T2I R@1	ImageNetV2 Top-1	ImageNetV2 Top-5
CLIP	50.02	30.35	78.59	59.72	<u>55.96</u>	**83.41**
DAC–LLMs	33.6	37.7	53.2	64.9	45.07	75.65
DAC–SAM	33.0	34.0	58.4	61.5	46.72	76.13
CLoVe (w/o patching)	48.0	42.5	69.2	68.2	46.05	75.10
CE-CLIP	<u>55.3</u>	<u>46.9</u>	74.9	68.3	43.32	74.28
FSC-CLIP (CC3M)	47.84	38.81	73.50	67.46	48.06	77.79
CLIC-RedCaps	52.5	33.8	79.4	62.8	55.30	82.49
NegCLIP	**56.84**	41.56	**83.03**	<u>68.73</u>	48.92	77.93
TripletCLIP	47.18	33.17	73.70	60.20	48.37	77.98
LACE (Ours)	51.8	**49.62**	<u>79.7</u>	**71.22**	**56.66**	<u>83.33</u>

E.7 MMVP-VLM: Fine-grained Visual Pattern Understanding

Table 23 reports compositional accuracy across the MMVP-VLM visual attribute taxonomy. **LACE** attains the **best overall average** (35.6%), outperforming all compared CLIP variants on this fine-grained pattern understanding benchmark. The gains are broad rather than isolated: LACE is **best** on *Color and Appearance* (80.0%) and *Viewpoint and Perspective* (46.67%), and is also tied for best on *State and Condition* (46.7%) and *Quantity and Count* (20.0%). Notably, it matches the strongest baseline on *Orientation and Direction* (26.7%) while improving substantially on harder appearance- and viewpoint-sensitive categories, suggesting stronger fine-grained binding of visual cues beyond coarse global alignment.

F Ablations and Diagnostics

This appendix collects ablations and diagnostic analyses used to isolate the contribution of LACE's counterfactual supervision. Unless otherwise stated, all ablations use the same OpenCLIP ViT-B/32 initialization, MS-COCO fine-tuning setup, and evaluation protocol as the main experiments.

F.1 Effect of Counterfactual Hard Negatives

To isolate the effect of LACE's counterfactual (**CF**) hard negatives from standard fine-tuning, we compare a plain fine-tuned baseline (*LACE (FT baseline)*) against *LACE + CF Hard Negatives* under an otherwise matched setup. Table 24 reports average accuracies across all compositional benchmarks (What'sUp, COLA, Winoground, SugarCrepe, SugarCrepe++) as well as the MMVP-VLM diagnostic. The comparison uses the same backbone, data, optimizer, and training schedule. Counterfactual hard negatives consistently improve over the fine-tuned baseline, indicating that the gains are not explained by standard fine-tuning alone.

Table 23: **MMVP-VLM:** Compositional accuracy (%) across visual patterns. Symbols denote visual attributes following the MMVP-VLM [2] taxonomy: Orientation and Direction, Presence of Specific Features, State and Condition, Quantity and Count, Positional and Relational Context, Color and Appearance, Structural Characteristics, Texts, and Viewpoint and Perspective. **Bold** indicates the best score and underline indicates the second-best.

Method	Backbone	Res										Avg.
CLIP	ViT-B/32	224	**26.7**	6.7	40.0	6.7	**26.7**	66.7	33.3	**26.7**	13.3	27.4
DAC-LLMs	ViT-B/32	224	13.3	20.0	26.7	**20.0**	6.7	53.3	**40.0**	**26.7**	6.7	23.7
DAC-SAM	ViT-B/32	224	0.0	20.0	26.7	6.7	13.3	66.7	26.7	**26.7**	20.0	23.0
CLoVe (w/o patching)	ViT-B/32	224	20.0	13.3	26.7	**20.0**	6.7	66.7	33.3	20.0	0.0	23.0
CE-CLIP	ViT-B/32	224	6.7	20.0	**46.7**	13.3	20.0	53.3	20.0	20.0	13.3	23.7
FSC-CLIP (CC3M)	ViT-B/32	224	6.7	6.7	13.3	**20.0**	6.7	53.3	26.7	**26.7**	20.0	20.0
CLIC-RedCaps	ViT-B/32	224	6.7	**26.7**	20.0	6.7	0.0	60.0	20.0	6.7	26.7	19.3
NegCLIP	ViT-B/32	224	20.0	13.3	**46.7**	13.3	20.0	46.7	20.0	**26.7**	26.7	25.9
TripletCLIP	ViT-B/32	224	6.7	13.3	40.0	6.7	13.3	53.3	13.3	13.3	20.0	20.0
LACE (Ours)	ViT-B/32	224	**26.7**	20.0	**46.7**	**20.0**	20.0	**80.0**	33.3	**26.7**	**46.67**	**35.6**

CF hard negatives yield consistent improvements over the FT baseline, increasing the overall average from 45.04% to **47.39**%. The largest gains appear on benchmarks that stress *fine-grained* visual grounding and binding: **COLA** improves from 35.56% to **39.05**% and **MMVP-VLM** from 29.62% to **35.56**%, with additional gains on **Winoground** (15.83% $\rightarrow$ **17.42**%) and both SugarCrepe variants. Overall, this controlled comparison supports the interpretation that the observed compositional robustness improvements are not explained by standard fine-tuning alone; they are associated with adding CF hard negatives that explicitly target cross-modal disambiguation under counterfactual perturbations.

Table 24: **Effect of CF hard negatives in LACE.** We report average accuracies (%) over compositional benchmarks (What'sUp, COLA, Winoground, SugarCrepe, SugarCrepe++) and the fine-grained MMVP-VLM diagnostic, under different training configurations. ✓ indicates that CF hard negatives are used; absence corresponds to standard fine-tuning. Best/second-best per column are **bold**/underlined.

Model	CF HN	What'sUp	COLA	Wino-ground	Sugar-Crepe	Sugar-Crepe++	MMVP-VLM	Avg.
CLIP		41.77	22.86	16.40	72.90	54.60	27.42	39.33
LACE (FT baseline)		42.12	35.56	15.83	82.80	64.30	29.62	45.04
LACE + CF Hard Negatives	✓	**43.56**	**39.05**	**17.42**	**83.52**	**65.22**	**35.56**	**47.39**

F.2 Design Ablations

Table 25 separates two design choices in LACE: which modality receives counterfactual supervision, and which latent edit operator is used to synthesize hard negatives. The branch ablation tests whether image-side, text-side, or symmetric counterfactuals are most effective. The edit-operator ablation tests whether performance comes from simply weakening an active concept, moving activation to a same-type target atom, or replacing the source dictionary direction with a target concept while preserving the original activation strength. The default LACE setting uses symmetric image/text-side counterfactuals with the full edit operator.

Interpreting geometry diagnostics under ablations. The ablations in Table 25 isolate the training mechanism responsible for the main gains. The comparison between the fine-tuned baseline and LACE with counterfactual hard negatives shows that standard fine-tuning alone is not sufficient: the improvement is associated with adding targeted near-miss supervision. However, we do not interpret hubness, reciprocity, or local-edit stability as independent causal variables that determine benchmark accuracy. Instead, they serve as diagnostics for whether hard-negative training improves the shared image–text space or merely sharpens one retrieval direction.

Table 25: Design ablations for LACE. We isolate the contribution of counterfactual branch design and latent edit operators. Image CF denotes image-side counterfactuals added to the T2I objective; Text CF denotes text-side counterfactuals added to the I2T objective. All variants use the same OpenCLIP ViT-B/32 initialization, MS-COCO fine-tuning data, optimizer, and evaluation protocol. Gray cells indicate the default LACE configuration.

Ablation	Variant	Image CF	Text CF	What'sUp	COLA	Wino.	SC	SC++	MMVP
Branch	FT baseline	–	–	42.12	35.56	15.83	82.80	64.30	29.62
	Image-side only	✓	–	43.14	34.70	16.09	82.03	63.50	31.99
	Text-side only	–	✓	43.30	37.19	16.98	82.79	64.30	33.65
	Symmetric LACE	✓	✓	43.56	39.05	17.42	83.52	65.22	35.56
Edit operator	Attenuation only	✓	✓	43.07	38.16	16.00	82.70	62.00	33.78
	Swap only	✓	✓	42.71	35.64	15.00	82.94	64.00	30.00
	Transplant only	✓	✓	43.30	37.24	17.00	83.27	64.29	34.20
	Full LACE	✓	✓	43.56	39.05	17.42	83.52	65.22	35.56

Table 26: **Semantic-targeting control under the matched fine-tuning protocol.** The shuffled control preserves the counterfactual training structure while breaking the correspondence between the sample and the applied semantic edit.

Training	What'sUp	COLA	Wino.	SC	SC++	MMVP	Avg.
FT baseline	42.12	35.56	15.83	82.80	64.30	29.62	45.04
Shuffled CF	42.90	36.07	15.11	82.40	63.00	30.10	44.93
LACE	**43.56**	**39.05**	**17.42**	**83.52**	**65.22**	**35.56**	**47.39**

This distinction is important because hard negatives can improve benchmark scores while producing undesirable geometry, such as increased hubness, reduced reciprocity, or unstable local neighborhoods. LACE is designed to avoid this failure mode by applying counterfactual pressure symmetrically to both image and text embeddings. The geometry results therefore support a more modest but useful claim: successful counterfactual supervision should improve compositional discrimination without warping the shared space into a one-sided or brittle retrieval geometry. LACE satisfies this criterion better than text-centric hard-negative baselines in our diagnostics. Across the matched FT, image-only, text-only, and symmetric configurations, the benchmark average increases from 45.04 to 45.24, 46.37, and 47.39, while Mutual@10 increases from 46.57 to 48.01, 49.80, and 51.41, Edge@10 from 52.46 to 52.80, 53.10, and 53.79, and I2T Gini decreases from 0.4797 to 0.4710, 0.4630, and 0.4552. Locally, relation-swap brittle fraction falls from 3.27% (FT) to 2.10%, 1.10%, and 1.10%, and attribute-swap brittle fraction from 4.65% to 4.40%, 2.10%, and 1.60% for image-only, text-only, and symmetric CF, respectively. The geometry metrics thus capture complementary aspects of the shared space rather than a single predictor of accuracy; for example, text-only CF attains the lowest T2I Gini (0.431 vs. 0.450 for symmetric LACE), while symmetric LACE gives the best downstream performance. This co-variation connects the branch ablation to representation geometry, although it does not establish that any individual geometry statistic causally mediates benchmark accuracy.

F.3 Semantic Specificity of Latent Counterfactuals

We examine whether the gains from LACE depend on the intended sample-concept correspondence or arise from generic embedding perturbations. As a control, we train a residual-shuffled variant that preserves the reconstruction procedure, modality, inferred concept type, number of counterfactuals, symmetric training branches, and hard-negative objective, but replaces the sample-specific edit with a same-type residual drawn from an unrelated example. The shuffled and LACE counterfactuals have closely matched edit magnitudes and negative hardness. A 5,000-pair audit confirms the control is not easier: edit distances match (image 0.0018/0.0018, text 0.0119/0.0120) and shuffled negatives have slightly higher hard-negative LogSumExp (32.0 vs. 31.2). Table 26 compares their downstream performance under the same fine-tuning protocol.

As shown in Table 26, the shuffled control remains close to standard fine-tuning, with an average score of 44.93 compared with 45.04 for the fine-tuned baseline. In contrast, LACE improves all six benchmarks and raises the average to 47.39. Since the shuffled control preserves the counterfactual training structure and approximate edit difficulty while removing the sample-specific semantic

correspondence, this comparison indicates that the gains depend on targeted semantic edits rather than perturbation magnitude alone.

We also measure whether individual edits move embeddings toward their intended semantics. Across 606 eligible image edits, the preference for the target description increases by 0.0122 on average and is positive in 99.67% of cases. Displacement-matched random and shuffled controls produce average changes of only 0.0002 and 0.0001, respectively, and LACE exceeds each control on 99.34% of paired examples. On a 50k CC3M bank, target-concept prevalence among retrieved real images rises from 1.056% to 1.254% after editing, with no enrichment for the controls. On the text side, 965 eligible edits show a mean directional alignment of 0.5299 with the corresponding caption edit, with a positive target-similarity change in 95.85% of cases. The real-image nearest-neighbor analysis in Appendix G.2 provides complementary qualitative evidence that these edits introduce the target concept while largely preserving non-edited context.

This validation is object-dominated: only 10.58% of image-side and 17.08% of text-side benchmark edits are eligible. These results support interpreting the generated vectors as semantically directed latent counterfactuals. They do not imply that every edited embedding corresponds to an exactly realizable pixel-space example, particularly for rare, abstract, or weakly represented relational concepts.

F.4 Hyperparameter Sensitivity

Table 4 summarizes the sensitivity of LACE to the main hyperparameters controlling counterfactual generation and sparse decomposition. The setting $K_{cf} = 0$ or $\gamma = 0$ corresponds to the fine-tuned baseline without counterfactual hard negatives. The default configuration used in the main experiments is $K_{cf} = 4$, $\gamma = 0.5$, $\lambda = 0.10$, and a 14k-atom dictionary.

TopP/TopQ editable-set sensitivity. The TopP/TopQ filter determines which sparse atoms are eligible for counterfactual editing. We use an intersection rule, $\mathcal{K}(z) = \mathrm{TopP}(|\alpha_k^\star(z)|) \cap \mathrm{TopQ}(\langle z, d_k \rangle)$, rather than coefficient-only or cosine-only selection. Coefficient-only selection often retains atoms that help reconstruct the embedding but are semantically brittle due to dictionary redundancy. Cosine-only selection gives semantically plausible concepts, but these concepts may not actually participate in the sparse reconstruction, leading to edits that are weak or disconnected from the sample-specific support. The intersection rule is more conservative: it reduces the editable-set size, but improves edit reliability by requiring a source atom to be both structurally active and semantically aligned. In practice, very small TopP/TopQ values reduce counterfactual diversity because many examples have few valid same-type targets, whereas very large values increase coverage at the cost of admitting unstable atoms. We use the default setting as a middle ground that preserves sufficient editable-set coverage while avoiding noisy source concepts.

Concept coverage. The relevant constraint is usable edit coverage rather than dictionary size alone: a concept must be represented in the dictionary, recovered in the sample-specific sparse support, and admit an absent same-type target. In a stratified audit, object concepts have broad vocabulary coverage, although exact benchmark-edit recovery decreases from 39.8% for frequent objects to 26.9% for rare objects. Attributes are less isolated (15.6% active, 10.8% exact-edit recovery), interaction relations reach 9.2% active recovery, and spatial relations have near-zero exact-edit recovery, incomplete coverage generally yields fewer or weaker counterfactuals rather than arbitrary ones. Validation is therefore strongest for object-centric edits.

F.5 Cross-Objective Validation with SigLIP

We further test whether LACE is specific to CLIP ViT-B/32 and softmax InfoNCE by evaluating it on SigLIP ViT-B/16, a pairwise-sigmoid dual encoder. In this setting, LACE counterfactuals are added as explicit negative image–text pairs with label -1, using the same sparse concept-editing mechanism; details are in Appendix D.10.

As shown in Table 27, LACE improves over both pretrained and fine-tuned SigLIP baselines across all reported compositional metrics. It also recovers the COLA drop caused by standard fine-tuning and raises the average score from 45.76 to 49.44, indicating that LACE transfers beyond softmax InfoNCE to pairwise sigmoid dual-encoder training.

The SigLIP results show that LACE-style counterfactual supervision is not limited to the OpenCLIP ViT-B/32 setting. Standard SigLIP fine-tuning improves several metrics but slightly reduces COLA relative to the pretrained SigLIP baseline. Adding CF hard negatives recovers this drop and improves all reported SigLIP metrics, raising the average score from 45.76 to 49.44.

Table 27: **Effect of CF hard negatives across CLIP and SigLIP.** Avg. accuracy (%) over compositional benchmarks and MMVP-VLM under pretrained, fine-tuned, and CF-hard-negative training settings. ✓ denotes use of CF hard negatives; bold marks the best result within each backbone block.

Model	CF HN	What'sUp	COLA	Wino-ground	SC	SC++	MMVP-VLM	Avg.
CLIP ViT-B/32		41.77	22.86	16.40	72.90	54.60	27.42	39.33
LACE FT baseline		42.12	35.56	15.83	82.80	64.30	29.62	45.04
LACE + CF Hard Neg.	✓	**43.56**	**39.05**	**17.42**	**83.52**	**65.22**	**35.56**	**47.39**
SigLIP ViT-B/16		44.00	37.46	16.67	78.69	57.35	31.85	44.34
SigLIP FT		44.97	34.89	18.00	79.94	64.92	31.85	45.76
SigLIP FT + CF Hard Neg.	✓	**47.00**	**40.90**	**19.00**	**83.98**	**67.95**	**37.78**	**49.44**

F.6 Training-Data Generalization with GBC

To test whether LACE depends on MS-COCO fine-tuning, we additionally train using Graph-Based Captioning (GBC) [43] image–text data. We use GBC as an alternative image–text training source and apply the same embedding-space counterfactual construction. Table 28 shows that standard GBC fine-tuning improves over pretrained CLIP, while adding LACE counterfactual hard negatives further improves all six benchmarks. The average score increases from 41.83 to 45.76, suggesting that LACE is not specific to MS-COCO supervision.

Table 28: **Training-data generalization with GBC.** Average accuracies (%) on compositional benchmarks and MMVP-VLM when fine-tuning with Graph-Based Captioning (GBC) image–text data. CF denotes LACE counterfactual hard negatives.

Model	SC	SC++	Wino.	COLA	What'sUp	MMVP	Avg.
CLIP	72.90	54.60	16.40	22.86	41.77	27.42	39.33
CLIP + GBC FT	77.60	57.60	17.10	27.90	41.90	28.89	41.83
CLIP + GBC FT + CF	**81.92**	**64.32**	**19.00**	**32.00**	**44.00**	**33.33**	**45.76**

G Qualitative Analysis of Counterfactual Edits

G.1 Qualitative Retrieval Examples

Figure 4 shows representative image–caption pairs where LACE assigns higher cosine similarity to the matched pair than several baseline CLIP variants. These examples include object-centric captions, long web-style captions, product-style descriptions, and visually grounded scene descriptions. While qualitative examples are not a substitute for benchmark evaluation, they illustrate the same trend observed quantitatively: LACE improves alignment on semantically specific image–text pairs without changing the inference architecture.

G.2 Nearest-Neighbor Analysis of Latent Edits

To examine how individual counterfactual edits alter the embedding space, we perform a nearest-neighbor analysis using real COCO validation images. For each source image, we infer its sparse code and select the strongest grounded object atom from the reliable editable set. We then choose a same-type target concept that is absent from both the source annotations and its pre-edit sparse support, apply the localized transplant described in Appendix D.6, and retrieve the ten nearest images before and after editing. Each retrieved neighbor is annotated for evidence of the target concept, the original source concept, and preserved non-edited context. As a control, we apply a random

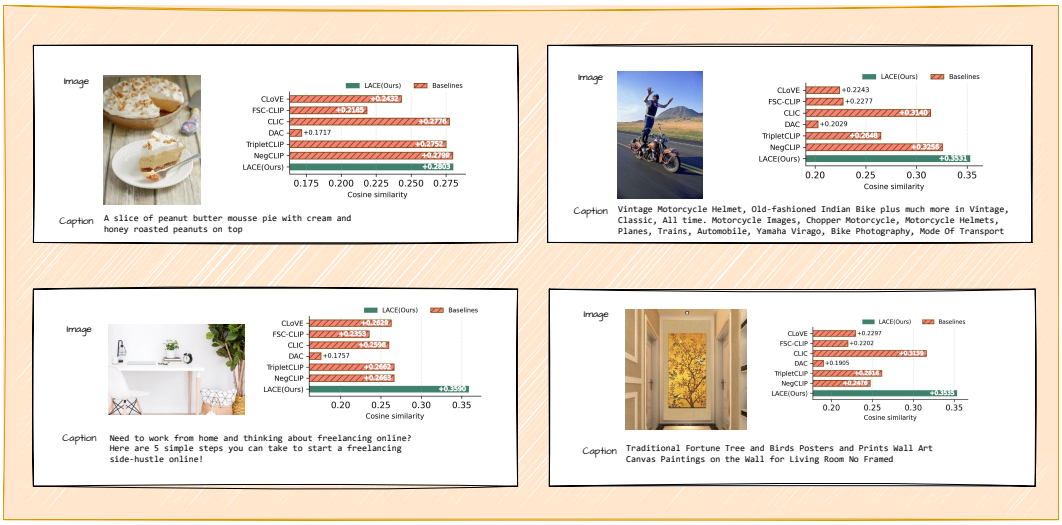

Figure 4: **Qualitative image–caption alignment examples.** For each image–caption pair, bars report cosine similarity assigned by LACE and baseline CLIP variants; higher is better. LACE consistently assigns stronger similarity to the matched caption across diverse web-style examples, including food, vehicle, indoor-scene, and product-style captions.

Table 29: Mean concept occurrences among the top-10 retrieved COCO images across six counterfactual transplants. Target and source hits measure the intended semantic change, while context hits measure preservation of non-edited content.

Query embedding	Target hits	Source hits	Context hits
Before editing	0.0	8.8	7.8
After LACE transplant	**7.7**	**2.3**	7.7
Displacement-matched random control	0.2	7.5	**8.2**

tangent-space perturbation matched to the cosine displacement of the corresponding LACE edit. The reference image is used only to select the target concept, no feature from that image is transferred into the source embedding.

Across the six successful edits, the target concept is absent from the original top-10 neighbors but appears within ranks 1-3 after editing. Target occurrences increase from 0.0 to 7.7, while occurrences of the source concept decrease from 8.8 to 2.3. At the same time, non-edited context remains nearly unchanged (7.8 versus 7.7), indicating that the edit changes the selected concept without replacing the entire neighborhood. The original-edited cosine similarities range from 0.71 to 0.80, showing that these are substantial but localized movements. Excluding the target reference image itself, the edited queries still retrieve an average of 6.8 additional target-containing images. In contrast, displacement-matched random perturbations rarely introduce the target concept and largely preserve the original source neighborhood.

Figure 3 shows representative sheep→zebra and donut→broccoli edits. Before editing, their neighborhoods are dominated by the respective source concepts. After editing, zebra and broccoli images become dominant while scene context such as grass, fields, plates, tables, and food remains visible. The same target-directed behavior occurs for couch→refrigerator, refrigerator→toilet, broccoli→pizza, and frisbee→fire-hydrant transplants.

Failure cases. Not every valid edit produces a complete semantic transition. For ski→pizza, the edited embedding is effectively unchanged (original-edited cosine 1.000), and the retrieved neighborhood remains identical. For motorcycle→keyboard, the edit is nonzero but too small to overcome the dominant source neighborhood; the cosine similarity is 0.961, with no target images appearing in the top ten. These failures have substantially smaller cosine displacements, approximately 0 and 0.039, than the successful edits (0.199-0.290). Edit magnitude therefore provides

a useful diagnostic for ineffective transplants, although these examples alone are insufficient to define a universal threshold.

H Broader Impact

LACE is intended as a research method for improving compositional robustness in CLIP-style vision-language models. By strengthening attribute binding, relation sensitivity, and fine-grained image–text alignment, it may support more reliable retrieval and grounding in settings where models must distinguish which attributes belong to which objects and how objects relate to one another.

As with any improvement to vision-language representation learning, downstream impact depends on the application context. LACE does not introduce open-ended generation, collect new user data, or deploy an autonomous decision-making system. However, fine-grained retrieval and recognition models can be used in a wide range of applications, some of which may require additional fairness, privacy, or safety evaluation. The method also inherits the limitations of the underlying pretrained models and training data, including possible dataset biases and incomplete coverage of rare or abstract concepts.

We view LACE as a contribution toward better understanding and improving compositional alignment, rather than as a standalone deployment solution. Practical deployments should follow the validation practices appropriate to their domain, including robustness checks and application-specific evaluation.