

What Makes Synthetic Hard Negatives Work in Vision-Language Pretraining?

Nikos Giakoumoglou¹, Paschalis Giakoumoglou², Andreas Floros¹,
Kleanthis Marios Papadopoulos¹, and Tania Stathaki¹

¹ Imperial College London, London, UK

² Centre for Research and Technology Hellas, Thessaloniki, Greece
nikos@imperial.ac.uk

Abstract. Synthetic hard negatives generated in the representation space have proven effective for unimodal self-supervised learning, but transferring this idea to vision-language pretraining is not straightforward. We analyze six representation-space synthesis strategies and identify two failure modes in their transfer to vision-language pretraining: cross-modal constructions that produce overly easy negatives or pull them toward the query, and intra-modal constructions that incorporate the matched positive. We also observe logit-scale saturation when training with synthetic hard negatives and a learnable temperature, and find that fixing the temperature improves downstream performance. Using this geometric analysis we propose SNAP, which generates intra-modal hard negatives that never involve the positive from either modality, avoiding both failure modes entirely. SNAP is model-agnostic, requires no external generative models, and adds less than 10% training time overhead. Evaluated on top of CLIP and FLIP across multiple architectures and datasets, SNAP delivers consistent improvements on zero-shot retrieval, zero-shot classification, and linear probe evaluation.

Keywords: CLIP · Contrastive learning · Hard negatives · Synthetic negatives · Vision-language pretraining · Zero-shot transfer

1 Introduction

Contrastive pretraining using weakly-supervised image-text pairs collected from the web has become the dominant paradigm for learning generic visual representations, progressively replacing supervised pretraining on large labeled datasets [8]. The core idea is to jointly learn an aligned embedding space for images and text from paired data, maximizing the similarity of matched (“*positive*”) image-text pairs while pushing apart mismatched (“*negative*”) pairs via a batch-level InfoNCE loss [18, 42]. Seminal works CLIP [48], ALIGN [21], and BASIC [45] demonstrated remarkable zero-shot transfer capabilities by training on hundreds of millions of image-text pairs. However, standard CLIP-style training samples image-text pairs randomly, without explicitly controlling the difficulty or multiplicity of in-batch negatives. Although higher-similarity negatives already receive greater

softmax weight in InfoNCE, their availability depends on batch composition. Hard-negative sampling and synthesis provide explicit control over this training signal [22, 52].

In unimodal self-supervised learning, synthetic hard negatives generated directly in the representation space have proven highly effective [12, 13, 22], offering a fast and lightweight alternative to input-space augmentation. This raises a natural question: *can representation-space synthetic negatives transfer to vision-language pretraining?* Prior work has explored hard negatives for vision-language models primarily in the **input space**: rule-based text perturbations [73, 79], LLM-based caption rewrites [10, 44], or diffusion-generated negative images [44]. These approaches incur significant computational overhead and typically address only one modality at a time (*cf.* Sec. 2). Representation-space generation would sidestep all of these issues. However, as we show in this paper, the transfer is **not straightforward**, and understanding *why* it fails reveals fundamental constraints imposed by multimodal representation geometry.

Unlike the single-modality setting where all representations inhabit one homogeneous space, vision-language models align two distinct modalities within a shared embedding space. This dual-domain structure introduces two geometric failure modes with no unimodal analogue. First, the *cross-modal* synthetic negatives that were tested, produced by blending image and text embeddings, fall into the modality gap [30]: a void between the image and text cones that real embeddings never occupy, making such samples trivially separable and uninformative. Second, restricting synthesis to a single modality but involving the positive pair causes *positive leakage*: the synthetic “negative” inherits genuine positive signal, sending contradictory gradients that confuse optimization. We characterize these geometric effects using representation-level diagnostics. Separately, we observe logit-scale saturation when synthetic hard negatives are introduced with a learnable temperature. Our temperature ablation supports combining positive-free intra-modal synthesis with a fixed temperature (Secs. 4.1 and 4.2).

We make the following **contributions**: **(i)** We identify and characterize two geometric failure modes that prevent naive transfer of representation-space synthetic negatives to vision-language pretraining, *i.e.*, modality gap collapse and positive leakage, and provide quantitative diagnostics for each (Sec. 4.1). **(ii)** Guided by this analysis, we propose SNAP, a model-agnostic method that generates synthetic hard negatives directly in the representation space for both image and text modalities, applicable on top of InfoNCE-based frameworks such as CLIP and FLIP (Sec. 4.3). **(iii)** We conduct comprehensive evaluations across multiple architectures (ViT-B/16, ViT-B/32, RN-50), datasets (CC3M, CC12M), and downstream tasks, *e.g.*, zero-shot retrieval (Tab. 5), zero-shot classification (Tab. 6), linear probe evaluation (Tab. 7), demonstrating broad improvements across the evaluated settings.

2 Related Work

2.1 Vision-Language Contrastive Pretraining

CLIP [48] and ALIGN [21] established the standard approach for vision-language contrastive pretraining: learning aligned image-text representations via a batch-level InfoNCE loss [42] over large-scale web-crawled data. LiT [76] showed that freezing a pretrained vision encoder and training only the text tower can be surprisingly effective, while SigLIP [75] replaced the softmax-based InfoNCE with a pairwise sigmoid formulation that removes the need for global normalization. BASIC [45] demonstrated the benefits of jointly scaling both encoders and data. Since then, a wide range of methods has advanced the field through modifications to the contrastive objective [16, 27, 67, 69], integration of self-supervised objectives [28, 37–39, 56, 81], architectural changes [3, 51, 61, 71], training efficiency improvements [29, 68], scaling [20, 58, 59, 65, 72], and improved training signals via synthetic captions [10, 32, 64, 77, 80]. SNAP is complementary to all these directions, augmenting the InfoNCE loss with representation-space synthetic negatives without modifying the architecture, data pipeline, or base objective.

2.2 Hard Negative Sampling

The quality and difficulty of negative samples are critical for contrastive learning [14, 54, 57, 63]. In **vision**, synthetic hard negatives generated in the representation space have been shown to significantly improve self-supervised representation learning [12, 13, 15, 22, 52]. MoCHi [22] mixes hard-negative features, while SynCo [13] studies six synthesis strategies that underpin our analysis. In **vision-language learning**, linguistic perturbations [73, 79] construct negative captions, and DiHT [47] increases the contribution of hard negatives through importance sampling while NegCLIP [73] and related methods [79] construct negatives through linguistic perturbations. In representation space, m^2 -Mix within m^3 -Mix [41] geodesically mixes another matched image-text pair to form cross-modal negatives. SNAP instead synthesizes query-specific negatives within each modality, using hard-negative mixing and noise during pretraining from scratch.

2.3 Synthetic Augmentation for Vision-Language Models

Several recent methods synthesize additional training data for CLIP-style pretraining, but these approaches operate in the **input space** and rely on large external foundation models. On the text side, captions are rewritten or generated using LLMs [10, 44], captioning models [32, 68], or multimodal LLMs [26, 33, 64, 80]. On the image side, TripletCLIP [44] generates hard negative images using text-to-image diffusion models (*e.g.*, SDXL-Turbo [46]) conditioned on LLM-generated captions. These methods introduce external knowledge from models that may overlap with downstream benchmarks, making it difficult to disentangle representational improvements from data leakage, and add substantial computational overhead. SNAP avoids these issues: synthetic negatives are generated from

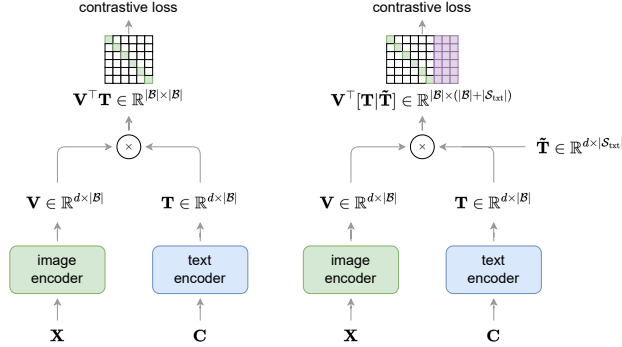


Fig. 1: Architecture comparison for the image-to-text direction. (a) Standard CLIP encodes images $\mathbf{X}$ and captions $\mathbf{C}$ into embeddings $\mathbf{V} \in \mathbb{R}^{d \times |\mathcal{B}|}$ and $\mathbf{T} \in \mathbb{R}^{d \times |\mathcal{B}|}$, computing the similarity matrix $\mathbf{V}^\top \mathbf{T} \in |\mathcal{B}| \times |\mathcal{B}|$ using only batch negatives. (b) Our method generates synthetic text negatives $\hat{\mathbf{T}} \in \mathbb{R}^{d \times |\mathcal{S}_{\text{txt}}|}$ directly in the representation space and concatenates them with batch embeddings to form an augmented similarity matrix $\mathbf{V}^\top [\mathbf{T} | \hat{\mathbf{T}}] \in |\mathcal{B}| \times (|\mathcal{B}| + |\mathcal{S}_{\text{txt}}|)$. In the similarity matrices, **green** cells denote positive pairs, white cells denote batch negatives, and **purple** cells denote synthetic negatives. The text-to-image direction follows an analogous procedure with synthetic image negatives (*cf.* Sec. 3.2).

in-batch embeddings alone, requiring no external models and no additional data. More on related work in Appendix Sec. C.

3 Preliminaries

In this section, we present the building blocks for integrating synthetic hard negatives into vision-language contrastive learning. We first review the standard CLIP objective (Sec. 3.1), then describe how synthetic negatives augment the contrastive loss in the representation space (Sec. 3.2), and finally define the generation strategies (Sec. 3.3). These preliminaries set up the vocabulary for Sec. 4, where we analyze why naively applying these strategies to the vision-language setting fails and present the design choices that lead to SNAP.

3.1 Contrastive Language-Image Pretraining

We assume two parameterized models: an image encoder f_θ with parameters θ (*e.g.*, a vision transformer [9] or a convnet [19]) and a text encoder g_ϕ with parameters ϕ (*e.g.*, a GPT-style transformer [49, 50]). Each input sample $(\mathbf{x}_i, \mathbf{c}_i)$ consists of an image $\mathbf{x}_i$ and text caption $\mathbf{c}_i$. The encoders produce embeddings $\mathbf{v}_i = f_\theta(\mathbf{x}_i)$ and $\mathbf{t}_i = g_\phi(\mathbf{c}_i)$, which are mapped to a common d -dimensional embedding space via projection layers and ℓ_2 -normalized, *i.e.*, $\|\mathbf{v}_i\|_2 = \|\mathbf{t}_i\|_2 = 1$. For a batch $\mathcal{B}$, each sample i is associated with one positive pair and $|\mathcal{B}| - 1$ in-batch negatives. To aid readability, throughout this section we use **red** to denote batch negatives. The bi-directional contrastive loss is computed as:

$$\mathcal{L}_{\text{CLIP}} = -\frac{1}{2|\mathcal{B}|} \left(\underbrace{\sum_{i \in \mathcal{B}} \log \frac{\exp(\mathbf{v}_i^\top \mathbf{t}_i / \tau)}{\sum_{j \in \mathcal{B}} \exp(\mathbf{v}_i^\top \mathbf{t}_j / \tau)}}_{\mathcal{L}_{\text{i2t}}} + \underbrace{\sum_{i \in \mathcal{B}} \log \frac{\exp(\mathbf{t}_i^\top \mathbf{v}_i / \tau)}{\sum_{j \in \mathcal{B}} \exp(\mathbf{t}_i^\top \mathbf{v}_j / \tau)}}_{\mathcal{L}_{\text{t2i}}} \right), \quad (1)$$

where τ is a learnable temperature parameter. We denote the two directional terms as *image-to-text* (i2t) and *text-to-image* (t2i), respectively. The denominators include both the positive term $j = i$ and the $|\mathcal{B}| - 1$ in-batch negatives $j \neq i$, following the standard InfoNCE formulation [42]. In the standard CLIP formulation [48], negatives are drawn exclusively from the batch $\mathcal{B}$, requiring large batch sizes (*e.g.*, 32,768) to provide a sufficiently large and diverse pool of negatives. Training with smaller batches significantly limits discriminative ability, as the lack of diverse negatives leads to weaker representations [5].

3.2 Synthetic Hard Negatives in the Representation Space

We augment the denominators of both $\mathcal{L}_{\text{i2t}}$ and $\mathcal{L}_{\text{t2i}}$ with synthetically generated hard negatives directly in the representation space. To aid readability, we use **purple** to denote synthetic negatives throughout this section. For the i2t term, we generate a set of synthetic text negatives $\mathcal{S}_{\text{txt}}^i = \{\tilde{\mathbf{t}}_k\}$ for each image query $\mathbf{v}_i$. Similarly, for the t2i term, we generate a set of synthetic image negatives $\mathcal{S}_{\text{img}}^i = \{\tilde{\mathbf{v}}_k\}$ for each text query $\mathbf{t}_i$. The augmented losses are:

$$\bar{\mathcal{L}}_{\text{i2t}} = -\frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \log \frac{\exp(\mathbf{v}_i^\top \mathbf{t}_i / \tau)}{\sum_{j \in \mathcal{B}} \exp(\mathbf{v}_i^\top \mathbf{t}_j / \tau) + \sum_{\tilde{\mathbf{t}}_k \in \mathcal{S}_{\text{txt}}^i} \exp(\mathbf{v}_i^\top \tilde{\mathbf{t}}_k / \tau)}, \quad (2)$$

Each query has its own synthetic-negative set, generated from its selected hard-negative pool; synthetic negatives are not shared across queries.

$$\bar{\mathcal{L}}_{\text{t2i}} = -\frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \log \frac{\exp(\mathbf{t}_i^\top \mathbf{v}_i / \tau)}{\sum_{j \in \mathcal{B}} \exp(\mathbf{t}_i^\top \mathbf{v}_j / \tau) + \sum_{\tilde{\mathbf{v}}_k \in \mathcal{S}_{\text{img}}^i} \exp(\mathbf{t}_i^\top \tilde{\mathbf{v}}_k / \tau)}. \quad (3)$$

The final loss combines both augmented directional terms:

$$\bar{\mathcal{L}} = \frac{1}{2} (\bar{\mathcal{L}}_{\text{i2t}} + \bar{\mathcal{L}}_{\text{t2i}}). \quad (4)$$

This formulation enriches the negative set with diverse, hard examples that would be computationally prohibitive to generate in the input space, while requiring minimal overhead as the generation occurs directly in the representation space during the forward pass (*cf.* Sec. 4.3). With synthetic negatives, we produce high-quality, diverse negatives **fast** and **cheaply**, enabling effective training even with moderate batch sizes (*cf.* Appendix Sec. D.1).

3.3 Synthetic Negative Generation Strategies

We define hardness by cosine similarity to the query [22]. Given a query image embedding $\mathbf{v}_i$, we select the top- N batch text negatives $\hat{\mathcal{T}}_N^i \subset \mathcal{B} \setminus i$ with the highest cosine similarity to $\mathbf{v}_i$; analogously, we select $\hat{\mathcal{V}}_N^i$ for text query $\mathbf{t}_i$. We implement six strategies to generate synthetic negatives. For the i2t direction, each strategy produces a synthetic text negative $\tilde{\mathbf{t}}_k$ from the query $\mathbf{v}_i$ and one or two hard negatives $\mathbf{t}_j, \mathbf{t}_l$ sampled uniformly from $\hat{\mathcal{T}}_N^i$

$$\tilde{\mathbf{t}}_k^{(s)} = \begin{cases} \alpha_k \cdot \mathbf{v}_i + (1 - \alpha_k) \cdot \mathbf{t}_j, & s = 1, \quad \alpha_k \sim \mathcal{U}(0, 0.5) \\ \mathbf{t}_j + \beta_k \cdot (\mathbf{t}_j - \mathbf{v}_i), & s = 2, \quad \beta_k \sim \mathcal{U}(1, 1.5) \\ \gamma_k \cdot \mathbf{t}_j + (1 - \gamma_k) \cdot \mathbf{t}_l, & s = 3, \quad \gamma_k \sim \mathcal{U}(0, 1) \\ \mathbf{t}_j + \mathcal{N}(\mathbf{0}, \sigma^2 \cdot \mathbf{I}), & s = 4 \\ \mathbf{t}_j + \delta \cdot \nabla_{\mathbf{t}_j}(\mathbf{v}_i^\top \mathbf{t}_j), & s = 5 \\ \mathbf{t}_j + \eta \cdot \text{sign}(\nabla_{\mathbf{t}_j}(\mathbf{v}_i^\top \mathbf{t}_j)), & s = 6 \end{cases} \quad (5)$$

All synthetic negatives are ℓ_2 -normalized after generation. Strategies $s=1$ (interpolation) and $s=2$ (extrapolation) blend the query with hard negatives [78]; $s=3$ (mixup) combines pairs of hard negatives [62, 78]; $s=4$ perturbs a hard negative with Gaussian noise; and $s=5, 6$ apply gradient-ascent and sign-based adversarial perturbations [17, 35] (see [13] for further detail). Crucially, $s=1$ and $s=2$ involve the query embedding $\mathbf{v}_i$ (or $\mathbf{t}_i$ in the t2i direction), producing *cross-modal* synthetic negatives, while $s=3$ and $s=4$ operate exclusively on same-modality negatives without involving any positive. As we show in Sec. 4.1, this distinction is critical in the vision-language setting.

4 Adapting Synthetic Negatives to Vision-Language Pretraining

The success of representation-space synthetic negatives in unimodal learning (Sec. 3.3) naturally suggests applying the same strategies to CLIP-style training. However, the dual-modality structure of vision-language models introduces failure modes with no unimodal analogue. This raises a central question: *What makes synthetic hard negatives work in vision-language pretraining?* In this section, we systematically analyze these failure modes (Sec. 4.1) along their multimodal geometry (Sec. 4.2), their secondary effect on the learnable temperature (Sec. 4.1), and present the resulting design choices that define SNAP (Sec. 4.3). All ablations use CC3M with ViT-B/16, reporting IN-val and IN-v2 zero-shot accuracy. We generate 64 synthetic negatives per query from the top-256 hardest negatives, split equally among active strategies. The hyperparameters $\sigma = 0.01$, $\delta = 0.01$, and $\eta = 0.01$ are set empirically without extensive tuning, following [13].

4.1 Naive Extension and Failure Modes

Cross-modal synthetic negatives fail due to geometry. The most direct extension is to apply all six strategies from Eq. (5) as defined, where $s=1$ and $s=2$

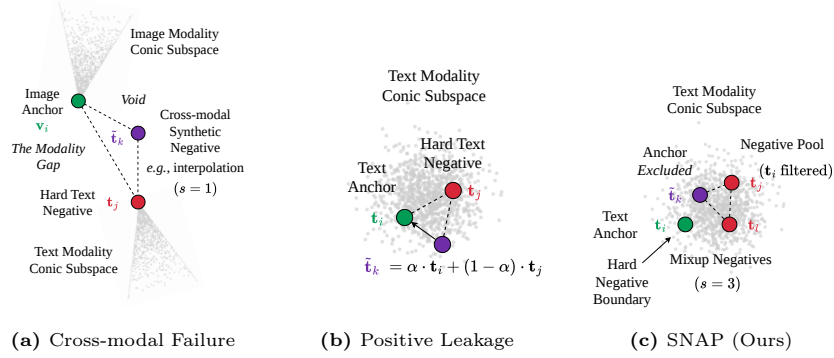


Fig. 2: Geometric intuition of failure modes in vision-language pretraining. (a) **Cross-modal failure:** Blending image and text embeddings creates samples in the “modality gap” that are trivially rejected (Sec. 4.1). (b) **Positive Leakage:** Including the positive pair in synthesis creates negatives that contain the actual signal, causing gradient conflict (Sec. 4.1). (c) **SNAP:** By generating intra-modal negatives and filtering the positive anchor, we create high-quality hard negatives that respect the modality geometry (Sec. 4.3).

blend the image query $\mathbf{v}_i$ with text negatives $\mathbf{t}_j$. However, vision-language models exhibit a well-documented *modality gap* [30] (Fig. 2a): image and text embeddings cluster in distinct cones of the shared space, and cross-modal interpolations such as $\alpha \cdot \mathbf{v}_i + (1 - \alpha) \cdot \mathbf{t}_j$ produce vectors in the void between them that real embeddings never occupy, making them trivially separable. As shown in Table 1, restricting to $s = 3, 4$ in intra-modal mode nearly recovers the CLIP baseline (10.0 vs. 10.4 on IN-val); as we show in Section 4.1, the remaining gap is closed and surpassed once the temperature is fixed.

Intra-modal negatives involving the positive fail due to semantic leakage. A natural fix is to convert all strategies to intra-modal variants by replacing the cross-modal query $\mathbf{v}_i$ in $s = 1, 2$ with the matched text positive $\mathbf{t}_i$. However, this introduces a subtler failure. Consider the intra-modal interpolation: $\tilde{\mathbf{t}} = \alpha \cdot \mathbf{t}_i + (1 - \alpha) \cdot \mathbf{t}_j$, where $\mathbf{t}_i$ is the positive caption for query $\mathbf{v}_i$ and $\mathbf{t}_j$ is a different caption. Because $\mathbf{t}_i$ is the correct match for $\mathbf{v}_i$, the synthetic “negative” $\tilde{\mathbf{t}}$ inherits positive signal. As $\alpha \rightarrow 1$, it becomes nearly identical to the true positive, yet the InfoNCE denominator treats it as a negative. This *positive leakage* (Fig. 2b) sends contradictory gradients: the numerator of Eq. (2) encourages similarity with $\mathbf{t}_i$, while the denominator penalizes similarity with $\tilde{\mathbf{t}} \approx \mathbf{t}_i$. Tab. 1 confirms that intra-modal generation with all six strategies (8.7 on IN-val) improves over the cross-modal variant but still underperforms the CLIP baseline.

Two distinct failure modes, one solution. The tested constructions alter negative geometry through cross-modal operations or introduce positive signal

Table 1: Cross-modal vs. intra-modal strats. Applying all six strategies in cross-modal mode degrades performance below the CLIP baseline. Converting to intra-modal mode helps but still underperforms. Restricting to $s = 3$ and $s = 4$ recovers performance. We highlight the default hyperparameter.

Method	Strategies	Mode	IN-val	IN-v2
CLIP [48]	n/a	n/a	10.4	8.4
SNAP	all 6	cross	7.6 (↓)	6.8 (↓)
SNAP	all 6	intra	8.7 (↓)	7.5 (↓)
SNAP	$s=3, 4$	intra	10.0 (↓)	8.0 (↓)

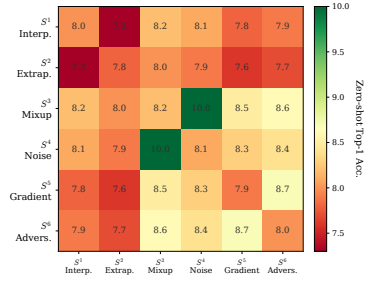


Fig. 3: Pairwise strategy complementarity on IN-val. Strategies $s = 3, 4$ achieve the highest accuracy.

through explicit inclusion of the matched positive. Strategies $s \in \{3, 4\}$ avoid both by generating negatives from non-matching pairs only. Restricting to $s \in \{3, 4\}$ nearly recovers the CLIP baseline (10.0 vs. 10.4 on IN-val, 8.0 vs. 8.4 on IN-v2; Table 1), with the remaining gap closed and surpassed once temperature is fixed (Section 4.1). Figure 3 confirms this: $s = 3$ and $s = 4$ consistently outperform any combination involving cross-modal blending or the positive pair.

Temperature divergence. Adding synthetic hard negatives changes the contrastive denominator and its interaction with the learnable temperature τ . In the evaluated configuration, CLIP+SNAP with learnable τ reaches the OpenCLIP logit-scale clamp of 100 within half an epoch after synthetic negatives are enabled (Fig. 4). This occurs with the positive-free intra-modal strategies $s = 3, 4$. Fixing $\tau = 0.07$ improves IN-val accuracy from 10.0 to 11.0 and IN-v2 accuracy from 8.0 to 9.6 (Tab. 2). Under the same fixed temperature, CLIP without synthetic negatives achieves 10.1 and 8.0, respectively. We therefore adopt fixed $\tau = 0.07$ as part of the SNAP training recipe. The two cross-modal strategies fail through distinct mechanisms: extrapolation ($s=2$) pushes vectors beyond the text cone, while interpolation ($s=1$) pulls toward the image query and thus the positive, making positive leakage the dominant failure rather than trivial separability, as we quantify in Section 4.2.

4.2 Multimodal Geometry of Synthetic Negatives

We characterize the failure modes in Sec. 4.1 using four complementary representation level diagnostics computed on ViT-B/32 trained on CC3M.

Leakage, separability, and hardness diagnostics. We quantify the failure modes with three metrics over 2,000 validation pairs (ViT-B/32, CC3M): *positive leakage* $\cos(\mathbf{t}, \mathbf{t}_i)$, where values above the random baseline (0.19) indicate semantic contamination; *trivially separable rate* (TSR), the fraction of synthetics easier

Table 2: Temperature ablation. With fixed $\tau = 0.07$, CLIP+SNAP outperforms both fixed- and learnable-temperature CLIP. We highlight the default setting.

Method	τ	IN-val	IN-v2
CLIP	learn.	10.4	8.4
CLIP	fixed	10.1	8.0
CLIP+SNAP	learn.	10.0 (↓)	8.0 (↓)
CLIP+SNAP	fixed	11.0 (↑)	9.6 (↑)

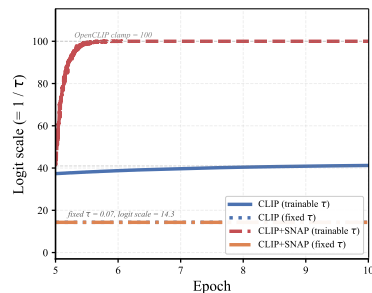


Fig. 4: Logit scale trajectories (ViT-B/32, CC3M, epochs 5 to 10). CLIP+SNAP with a learnable τ saturates the OpenCLIP clamp within half an epoch. Both fixed- τ runs remain stable.

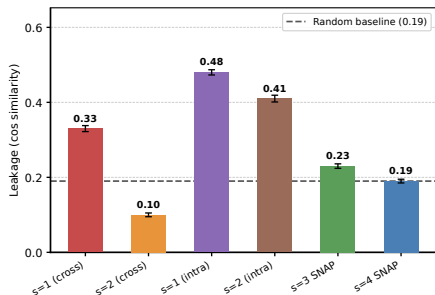


Fig. 5: Positive leakage per strategy (ViT-B/32, CC3M). mean $\cos(\tilde{\mathbf{t}}, \mathbf{t}_i)$ per strategy; dashed line marks the random baseline (0.19).

Table 3: TSR and relative hardness. Lower TSR and higher Δ hardness indicate harder, more informative negatives. We highlight SNAP strategies.

Strategy	TSR (%)↓	Δ hardness↑
$s=1$ (cross)	0.3	+0.281
$s=2$ (cross)	87.8	−0.189
$s=1$ (intra)	0.6	+0.133
$s=2$ (intra)	74.0	−0.096
$s=3$ (SNAP)	2.2	+0.041
$s=4$ (SNAP)	13.2	+0.045

than the 10th percentile of real hard negatives; and *relative hardness* $\Delta = \cos(\mathbf{v}_i, \tilde{\mathbf{t}}) - \cos(\mathbf{v}_i, \mathbf{t}_{\text{rand}})$. As shown in Fig. 5 and Tab. 3, cross-modal strategies fail for opposite reasons: $s=2$ (cross) lands in the void beyond the text cone (TSR 87.8%, $\Delta = -0.189$), while $s=1$ (cross) interpolates toward the positive (leakage 0.33). Intra-modal strategies involving the positive suffer from contamination ($s=1$ intra: 0.48; $s=2$ intra: 0.41) or trivial separability ($s=2$ intra: TSR 74.0%). Only $s=3$ and $s=4$ avoid cross-modal mixing and explicit positive inclusion, exhibiting leakage close to the random baseline (0.23 and 0.19), low TSR (2.2% and 13.2%), and positive relative hardness.

PCA subspace visualization. Fig. 6 projects image embeddings, text embeddings, and synthetic negatives onto the first two principal components. Cross-modal strategies scatter into the void between the image and text cones, a region real embeddings never occupy. Intra-modal strategies involving the positive cluster

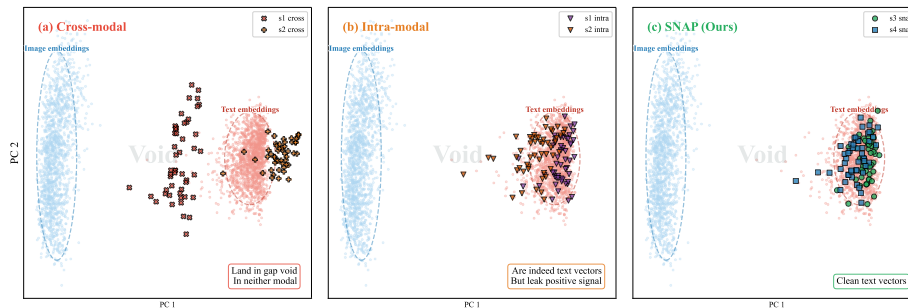


Fig. 6: PCA subspace of synthetic negatives (ViT-B/32, CC3M). (a) Cross-modal negatives land in the gap void between modality cones. (b) Intra-modal negatives involving the positive cluster near the positive anchor, causing leakage. (c) SNAP negatives ($s=3$, $s=4$) are confined within the text cone and away from the positive.

within the text cone but near the positive anchor, explaining the observed leakage. SNAP strategies ($s=3$, $s=4$) are confined within the text cone and distributed away from the positive, occupying the same subspace as real text negatives.

4.3 SNAP

Guided by the analysis in Secs. 4.1 and 4.2, we propose SNAP (Synthetically Negative Augmented Pretraining) that makes three principled design choices.

Intra-modal, positive-free strategies. SNAP uses only strategies $s = 3$ and $s = 4$ from Eq. (5), both operating in intra-modal mode (Fig. 2c). These avoid cross-modal geometric collapse and intra-modal positive leakage (Tab. 1 and Fig. 3).

Fixed temperature. SNAP fixes the temperature τ rather than learning it, preventing the logit scale saturation that arises when hard negatives interact with a learnable temperature (Tab. 2).

Table 4: Synthesis and directionality ablations on CC3M.

Variant	IN-val	IN-v2
Random-neg. dup.	10.0	8.0
Hard-neg. dup.	10.4	8.5
Hard-neg. wt.	10.5	8.7
$s = 3$	10.8	9.2
$s = 4$	10.6	8.8
$s = 3, 4$ (i2t only)	10.7	8.8
$s = 3, 4$ (t2i only)	10.6	8.7
SNAP	11.0	9.6

Bidirectional synthesis. SNAP generates synthetic negatives for both i2t and t2i directions. We compare against repeating randomly sampled or hard negatives in the contrastive denominator, and increasing the weight of hard-negative terms. Both synthesis strategies outperform these controls, with their combination performing best (Tab. 4). Bidirectional synthesis also outperforms either direction alone.

Computational overhead. Since SNAP generates negatives via vector operations—specifically additions, scalar multiplications, and normalization—on pre-computed embeddings, it avoids additional forward or backward passes through the encoders. This design ensures minimal impact on training efficiency, requiring no external generative models or architectural modifications. On a ViT-B/16 backbone, CLIP [48] requires 32.5 minutes per epoch, while SNAP adds only 3 minutes, resulting in an 9.23% overhead. Similarly, FLIP [29] requires 14.9 minutes per epoch, and the addition of SNAP increases this to 16.2 minutes, a comparable overhead of 8.72%. These results demonstrate that SNAP provides significant discriminative gains with a negligible increase in training time. Further ablations on batch size and the number of synthetic negatives are provided in the supplementary material (Appendix Sec. D).

5 Experiments

We train the CLIP and FLIP baselines and their SNAP variants under matched data and training configurations, except for the temperature treatment described above. We also train SigLIP [75] under our training setup. Results for LaCLIP [10], TripletCLIP [44], and NegCLIP [73] are taken from [44] and included for reference only. Detailed implementation details in Appendix Sec. B.

5.1 Implementation Details

Pretraining datasets. We pretrain on two image-text datasets of varying scale: CC3M [55] and CC12M [4]. Due to image link rot, the versions we obtained contain fewer samples than the original releases: $\approx 1.7\text{M}$ out of 3.3M for CC3M and $\approx 7.2\text{M}$ out of 12.4M for CC12M. This may lead to minor performance differences compared to models trained on the full datasets.

Model architectures. For the image encoder f_θ , we use ViT-B/16, ViT-B/32 [9], and ResNet-50 [19]. For the text encoder g_ϕ , we use a 12-layer Transformer [48]. Both encoders include projection heads mapping to a shared $d = 512$ dimensional embedding space.

Training setup. We use AdamW [34] with learning rate 5×10^{-4} , weight decay 0.2, and cosine scheduling with 10k warmup steps. Training uses batch size $|\mathcal{B}| = 4096$ on $4 \times \text{A100 GPUs}$ for 32 epochs, with 224×224 images and text length 77. The temperature τ is initialized to 0.07 (and kept constant for SNAP).

Reproducibility. For CC3M, we pretrain three checkpoints (different seeds) and report the mean, with standard deviations as subscripts in Tables 5 to 7; for CC12M, a single checkpoint is used due to computational constraints.

5.2 Zero-Shot Image-Text Retrieval

We evaluate cross-modal retrieval on Flickr30k [70] and MSCOCO [31], computing pairwise cosine similarities over the full test split and reporting Recall@{1,5,10} for both image-to-text and text-to-image retrieval in Table 5. SNAP improves retrieval in the large majority of settings for both CLIP and FLIP, holding across all three backbones and both pretraining scales, with gains appearing on both retrieval directions rather than favoring one; the few exceptions are isolated cells that are unchanged or lower by at most a few tenths of a point.

Table 5: Zero-shot image-text retrieval on Flickr30k and MSCOCO. We report Recall@K (R@K) metrics for text and image retrieval. **Symbols:** [†] Taken from [44]

Data Method	Text Retrieval						Image Retrieval					
	Flickr30k			MSCOCO			Flickr30k			MSCOCO		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
ViT-B/32												
CC3M	SigLIP [75]	7.2 _{.2}	19.8 _{.3}	26.5 _{.1}	3.1 _{.3}	10.0 _{.1}	15.7 _{.2}	5.3 _{.1}	15.2 _{.2}	22.3 _{.1}	2.8 _{.1}	9.0 _{.1} 14.0 _{.1}
	NegCLIP [73] [†]	4.9	13.8	19.6	1.7	6.3	10.4	4.8	13.0	18.5	2.0	6.6 10.6
	LaCLIP [10] [†]	3.7	10.9	16.0	1.6	5.1	8.9	3.5	10.8	16.0	1.7	6.0 9.5
	TripletCLIP [44] [†]	9.1	22.0	29.8	3.2	10.4	16.2	8.4	22.0	29.6	3.6	11.3 17.4
	CLIP [48]	9.8 _{.2}	25.5 _{.3}	35.7 _{.2}	5.0 _{.2}	15.0 _{.1}	22.3 _{.1}	6.9 _{.1}	19.8 _{.2}	28.5 _{.1}	4.2 _{.2}	12.7 _{.2} 19.0 _{.1}
	+ SNAP	10.4 _{.2}	26.3 _{.1}	36.9 _{.3}	5.5 _{.2}	15.2 _{.2}	23.0 _{.2}	7.9 _{.1}	20.4 _{.2}	28.9 _{.1}	4.1 _{.1}	13.6 _{.1} 20.1 _{.2}
	FLIP [29]	6.3 _{.3}	18.5 _{.1}	26.4 _{.2}	3.3 _{.3}	10.7 _{.3}	16.7 _{.3}	5.8 _{.3}	16.9 _{.1}	24.0 _{.3}	3.1 _{.1}	9.6 _{.1} 14.7 _{.2}
	+ SNAP	6.8 _{.1}	19.0 _{.1}	26.9 _{.3}	3.7 _{.1}	11.1 _{.3}	17.1 _{.1}	6.3 _{.1}	17.4 _{.3}	24.5 _{.1}	3.5 _{.1}	10.0 _{.2} 15.1 _{.2}
CC12M	SigLIP [75]	31.4	57.2	69.5	16.2	36.1	47.2	23.1	48.8	60.8	11.5	28.1 38.4
	NegCLIP [73] [†]	24.7	46.6	58.2	12.3	30.2	41.4	18.7	41.7	53.4	8.6	23.1 32.6
	LaCLIP [10] [†]	21.3	42.7	54.6	10.5	25.9	35.6	15.1	36.3	47.0	7.2	19.8 28.4
	TripletCLIP [44] [†]	28.0	55.9	65.7	14.6	33.0	43.8	25.3	52.4	63.3	11.4	28.5 39.0
	CLIP [48]	33.1	59.5	71.2	17.8	39.4	50.3	24.2	50.9	63.0	13.1	30.5 41.2
	+ SNAP	33.8	60.0	71.9	18.6	39.8	51.3	24.9	51.4	63.5	13.7	31.0 42.6
	FLIP [29]	26.2	50.1	61.8	14.4	32.1	43.5	20.2	44.1	56.1	10.7	26.2 36.4
	+ SNAP	27.2	50.9	63.1	15.1	32.7	44.5	20.9	45.4	56.8	10.4	26.8 37.2
ViT-B/16												
CC3M	SigLIP [75]	13.6 _{.3}	31.5 _{.1}	43.6 _{.1}	7.0 _{.2}	19.8 _{.1}	28.1 _{.2}	9.3 _{.1}	24.5 _{.2}	33.4 _{.1}	5.6 _{.1}	15.8 _{.2} 23.1 _{.3}
	CLIP [48]	16.2 _{.3}	37.8 _{.2}	49.6 _{.2}	7.7 _{.2}	21.5 _{.2}	31.5 _{.2}	12.0 _{.2}	28.4 _{.2}	38.2 _{.1}	6.6 _{.3}	18.1 _{.3} 26.2 _{.1}
	+ SNAP	17.1 _{.3}	38.7 _{.2}	50.5 _{.1}	8.1 _{.1}	22.1 _{.3}	32.7 _{.3}	12.8 _{.1}	28.9 _{.1}	39.2 _{.2}	7.2 _{.2}	18.4 _{.1} 26.8 _{.1}
	FLIP [29]	12.1 _{.2}	31.2 _{.2}	42.1 _{.1}	6.4 _{.2}	17.7 _{.1}	26.4 _{.1}	10.0 _{.2}	24.8 _{.1}	33.6 _{.3}	5.1 _{.1}	15.4 _{.1} 22.7 _{.2}
	+ SNAP	12.3 _{.2}	31.7 _{.3}	43.1 _{.1}	7.1 _{.3}	18.9 _{.2}	27.0 _{.3}	10.9 _{.3}	25.6 _{.3}	34.8 _{.1}	5.9 _{.1}	16.3 _{.3} 24.0 _{.2}
CC12M	SigLIP [75]	49.4	74.2	83.5	26.1	50.4	62.1	34.2	61.9	73.1	19.1	40.5 52.1
	CLIP [48]	53.6	77.5	85.9	29.4	53.8	65.8	38.5	66.2	76.9	21.8	44.4 56.1
	+ SNAP	55.1	78.3	87.1	30.2	54.7	67.1	39.0	67.6	78.2	22.3	45.0 56.6
	FLIP [29]	47.9	73.2	82.2	25.5	49.1	61.2	33.6	60.5	72.1	18.8	39.4 50.9
	+ SNAP	48.5	74.1	82.3	26.4	49.6	62.2	34.3	61.0	72.7	19.5	39.9 52.1
RN-50												
CC3M	SigLIP [75]	19.4 _{.1}	44.8 _{.1}	56.4 _{.2}	10.5 _{.1}	27.0 _{.1}	37.5 _{.3}	14.0 _{.2}	31.5 _{.2}	42.0 _{.1}	8.0 _{.1}	21.3 _{.1} 29.8 _{.1}
	CLIP [48]	20.8 _{.3}	45.8 _{.3}	56.8 _{.1}	11.7 _{.3}	29.6 _{.3}	41.4 _{.1}	16.1 _{.2}	36.5 _{.2}	47.1 _{.1}	9.6 _{.1}	24.3 _{.2} 34.0 _{.2}
	+ SNAP	21.1 _{.2}	47.2 _{.1}	58.1 _{.1}	11.6 _{.1}	29.4 _{.1}	42.1 _{.1}	16.0 _{.1}	37.4 _{.1}	47.8 _{.1}	10.9 _{.1}	25.3 _{.3} 34.9 _{.3}
	FLIP [29]	20.6 _{.1}	44.2 _{.2}	56.5 _{.2}	11.3 _{.1}	29.4 _{.2}	40.4 _{.1}	15.8 _{.2}	36.0 _{.1}	47.0 _{.3}	8.8 _{.2}	23.6 _{.1} 33.1 _{.3}
	+ SNAP	21.2 _{.1}	44.6 _{.2}	57.2 _{.1}	12.2 _{.3}	29.9 _{.3}	40.9 _{.2}	16.2 _{.3}	36.9 _{.1}	46.9 _{.2}	9.2 _{.3}	24.0 _{.2} 34.5 _{.2}

5.3 Zero-Shot Classification

We evaluate zero-shot classification on ImageNet [8, 53] and 10 downstream datasets spanning fine-grained recognition (Stanford Cars [24], FGVC Aircraft [36]), scene recognition (SUN397 [66]), texture classification (DTD [7]), and generic object classification (CIFAR [25]), encoding class names as text prompts and predicting via cosine similarity without any task-specific training (Table 6). SNAP improves accuracy across settings, with the clearest gains on the fine-grained Cars and Aircraft benchmarks, where CC3M ViT-B/32 accuracy roughly doubles (*e.g.*, Cars $0.8 \rightarrow 1.9$, Aircraft $1.9 \rightarrow 3.4$ for CLIP), while general benchmarks such as ImageNet show steady improvements.

Table 6: Zero-shot classification. Top-1 accuracy on ImageNet and 10 downstream datasets spanning fine-grained, scene, and object recognition. [‡]Taken from [44].

Data	Model	Food-101	CIFAR-10	CIFAR-100	SUN397	Cars	Aircraft	DTD	Pets	Caltech-101	Flowers	ImageNet
CC3M	ViT-B/32											
	SigLIP [75]	3.8 _{.3}	23.8 _{.3}	8.1 _{.2}	12.4 _{.2}	0.8 _{.2}	0.9 _{.1}	4.4 _{.2}	4.1 _{.1}	26.6 _{.2}	4.2 _{.2}	5.6 _{.1}
	NegCLIP [73] [‡]	–	–	8.2	–	–	1.0	6.8	4.1	29.4	3.7	4.6
	LaCLIP [10] [‡]	–	–	8.0	–	–	0.8	5.5	3.1	24.6	3.4	3.7
	TripletCLIP [44] [‡]	–	–	11.8	–	–	0.9	8.9	5.7	38.5	8.2	7.3
	CLIP [48]	5.9 _{.2}	28.7 _{.2}	10.0 _{.3}	14.5 _{.1}	0.8 _{.3}	1.9 _{.3}	8.2 _{.2}	7.0 _{.3}	41.5 _{.3}	5.9 _{.1}	8.0 _{.3}
	+ SNAP	6.5 _{.3}	29.3 _{.2}	10.4 _{.2}	14.7 _{.2}	1.9 _{.3}	3.4 _{.2}	9.0 _{.1}	7.3 _{.3}	42.1 _{.2}	7.3 _{.2}	8.2 _{.2}
	FLIP [29]	4.7 _{.1}	26.8 _{.2}	8.7 _{.3}	11.4 _{.2}	0.6 _{.2}	1.1 _{.1}	4.5 _{.1}	5.3 _{.3}	37.1 _{.2}	3.9 _{.1}	6.8 _{.3}
CC12M	+ SNAP	5.2 _{.1}	28.1 _{.2}	9.2 _{.2}	12.4 _{.3}	1.3 _{.1}	2.6 _{.2}	4.6 _{.1}	5.8 _{.2}	38.0 _{.2}	4.5 _{.3}	7.2 _{.1}
	SigLIP [75]	11.2	45.4	28.1	31.2	3.4	2.1	15.6	32.5	64.1	14.2	22.1
	NegCLIP [73] [‡]	–	–	26.0	–	–	1.5	14.2	30.8	63.4	13.5	20.2
	LaCLIP [10] [‡]	–	–	23.6	–	–	1.5	13.0	34.7	62.9	13.4	19.7
	TripletCLIP [44] [‡]	–	–	30.3	–	–	1.2	17.4	33.2	65.3	15.4	23.3
	CLIP [48]	13.1	50.2	31.5	35.1	4.1	2.5	18.2	35.8	67.4	16.8	24.6
	+ SNAP	13.5	50.6	32.7	36.2	5.0	3.0	19.1	36.7	68.1	17.3	24.8
	FLIP [29]	11.8	46.5	28.9	32.4	3.5	2.2	16.1	33.4	64.8	14.9	22.8
CC3M	+ SNAP	13.2	47.7	29.5	33.3	4.0	2.0	17.0	34.3	65.6	15.7	23.9
	ViT-B/16											
	SigLIP [75]	6.2 _{.3}	28.9 _{.3}	7.0 _{.1}	18.3 _{.2}	1.1 _{.3}	1.2 _{.3}	8.7 _{.1}	6.0 _{.2}	35.5 _{.2}	7.0 _{.1}	9.9 _{.2}
	CLIP [48]	7.4 _{.1}	34.3 _{.2}	9.8 _{.2}	19.0 _{.3}	0.9 _{.3}	1.2 _{.1}	10.1 _{.1}	9.0 _{.2}	46.3 _{.1}	8.4 _{.3}	10.4 _{.2}
	+ SNAP	8.4 _{.1}	34.4 _{.1}	10.5 _{.3}	19.6 _{.1}	1.3 _{.2}	2.0 _{.1}	10.8 _{.1}	10.5 _{.1}	47.3 _{.1}	9.0 _{.2}	11.0 _{.2}
CC12M	FLIP [29]	5.3 _{.2}	24.1 _{.3}	9.3 _{.3}	17.4 _{.1}	1.1 _{.2}	1.4 _{.3}	8.3 _{.2}	6.4 _{.2}	45.0 _{.2}	6.6 _{.1}	10.3 _{.3}
	+ SNAP	6.4 _{.2}	25.3 _{.3}	10.5 _{.1}	18.2 _{.1}	1.7 _{.3}	1.9 _{.3}	9.1 _{.3}	7.9 _{.2}	45.7 _{.2}	8.0 _{.1}	11.0 _{.2}
	SigLIP [75]	19.8	58.4	36.2	41.5	6.2	4.1	22.1	51.2	78.4	24.1	31.2
	CLIP [48]	22.4	63.1	39.5	44.8	7.5	4.8	24.8	55.4	82.1	27.5	34.6
	+ SNAP	23.1	63.0	40.1	46.2	7.9	4.6	25.4	56.1	83.2	28.5	35.3
CC3M	FLIP [29]	20.2	59.8	37.5	42.6	6.5	4.3	22.8	52.4	79.8	25.3	32.4
	+ SNAP	20.8	60.1	38.6	43.8	6.7	5.6	23.3	52.3	81.0	26.0	33.8
	RN-50											
	SigLIP [75]	7.5 _{.2}	17.9 _{.2}	5.2 _{.1}	21.1 _{.1}	1.1 _{.1}	1.5 _{.2}	9.4 _{.2}	8.8 _{.1}	38.8 _{.1}	9.5 _{.1}	13.1 _{.2}
	CLIP [48]	7.7 _{.1}	23.7 _{.1}	4.8 _{.3}	23.3 _{.1}	1.4 _{.2}	1.3 _{.2}	9.3 _{.1}	11.7 _{.2}	45.4 _{.2}	12.3 _{.2}	13.6 _{.1}
CC3M	+ SNAP	8.2 _{.1}	24.4 _{.2}	5.8 _{.1}	24.0 _{.1}	1.9 _{.3}	2.1 _{.2}	10.8 _{.1}	12.8 _{.2}	46.1 _{.1}	12.1 _{.2}	14.7 _{.2}
	FLIP [29]	9.0 _{.1}	23.1 _{.1}	6.7 _{.2}	22.4 _{.3}	0.8 _{.1}	1.1 _{.2}	10.9 _{.1}	9.0 _{.2}	41.8 _{.2}	11.7 _{.1}	13.7 _{.1}
	+ SNAP	9.5 _{.3}	22.9 _{.1}	7.4 _{.2}	23.2 _{.1}	1.6 _{.2}	1.9 _{.2}	11.7 _{.1}	10.4 _{.2}	42.2 _{.1}	13.0 _{.2}	14.9 _{.1}

5.4 Linear Probe Classification

Following standard protocols [23, 48], we freeze the image encoder and train a linear classifier, reporting top-1 test accuracy (Tab. 7). SNAP consistently produces more linearly separable visual features across backbones and data scales.

Table 7: Linear probe evaluation. Top-1 accuracy of a linear classifier trained on frozen image encoder features across ImageNet and 10 transfer datasets.

Data	Model	Food-101	CIFAR-10	CIFAR-100	SUN397	Cars	Aircraft	DTD	Pets	Caltech-101	Flowers	ImageNet
ViT-B/32												
CC3M	SigLIP [75]	35.4 _{.1}	72.5 _{.1}	48.4 _{.3}	29.4 _{.1}	9.5 _{.1}	13.6 _{.2}	31.9 _{.2}	34.0 _{.2}	61.6 _{.2}	50.8 _{.2}	30.2 _{.1}
	CLIP [48]	40.7 _{.3}	77.2 _{.3}	54.6 _{.1}	32.7 _{.1}	13.2 _{.2}	20.1 _{.3}	38.3 _{.3}	40.7 _{.1}	71.1 _{.1}	61.5 _{.1}	33.9 _{.1}
	+ SNAP	41.3 _{.2}	78.5 _{.2}	55.2 _{.3}	33.1 _{.3}	13.9 _{.1}	20.7 _{.1}	38.9 _{.3}	41.4 _{.3}	72.0 _{.1}	62.7 _{.2}	34.7 _{.1}
	FLIP [29]	40.8 _{.1}	74.8 _{.2}	52.8 _{.2}	34.7 _{.3}	13.9 _{.2}	17.9 _{.2}	39.8 _{.2}	37.6 _{.2}	69.7 _{.2}	61.1 _{.2}	32.9 _{.2}
	+ SNAP	41.4 _{.1}	75.6 _{.2}	53.6 _{.1}	36.1 _{.3}	14.5 _{.1}	18.7 _{.1}	41.3 _{.1}	37.5 _{.2}	70.8 _{.1}	62.0 _{.2}	33.5 _{.3}
CC12M	SigLIP [75]	54.2	84.6	63.8	44.1	24.8	31.2	48.9	56.4	79.2	78.4	45.8
	CLIP [48]	58.6	88.5	68.4	47.9	28.5	36.8	54.2	62.1	85.1	85.9	48.7
	+ SNAP	59.2	89.1	68.3	49.3	28.9	38.0	54.7	63.4	85.8	86.3	49.8
	FLIP [29]	57.5	87.2	67.1	46.8	27.4	35.1	53.0	60.9	83.9	84.1	47.5
	+ SNAP	58.0	88.1	68.0	47.3	27.2	35.3	53.0	61.7	84.9	84.7	48.6
ViT-B/16												
CC3M	SigLIP [75]	46.3 _{.2}	73.7 _{.2}	50.3 _{.1}	37.6 _{.3}	14.7 _{.1}	21.7 _{.2}	45.0 _{.1}	47.2 _{.1}	70.8 _{.2}	66.8 _{.1}	41.3 _{.2}
	CLIP [48]	49.2 _{.3}	77.1 _{.2}	55.2 _{.2}	40.3 _{.1}	18.1 _{.3}	24.6 _{.2}	47.8 _{.2}	51.2 _{.2}	73.7 _{.1}	71.6 _{.3}	43.1 _{.2}
	+ SNAP	49.7 _{.3}	77.5 _{.1}	56.6 _{.1}	41.0 _{.1}	19.2 _{.1}	25.3 _{.1}	48.1 _{.3}	52.2 _{.2}	74.4 _{.3}	72.3 _{.3}	43.7 _{.2}
	FLIP [29]	48.6 _{.1}	75.0 _{.1}	52.9 _{.3}	41.0 _{.2}	16.0 _{.2}	21.3 _{.3}	46.2 _{.1}	47.4 _{.3}	73.2 _{.2}	70.6 _{.3}	40.9 _{.1}
	+ SNAP	49.3 _{.3}	74.9 _{.1}	53.0 _{.2}	41.1 _{.2}	16.6 _{.2}	22.7 _{.1}	46.0 _{.3}	47.4 _{.2}	74.4 _{.1}	70.9 _{.1}	41.7 _{.1}
CC12M	SigLIP [75]	64.1	85.8	66.2	51.4	32.5	38.7	58.1	65.8	84.2	86.4	55.4
	CLIP [48]	68.4	89.2	70.5	55.1	36.8	43.1	62.4	70.5	88.5	90.1	58.7
	+ SNAP	68.8	90.0	71.3	56.0	38.0	43.8	63.2	71.1	89.6	90.6	59.9
	FLIP [29]	67.1	88.0	69.3	54.0	35.5	41.8	61.2	69.3	87.2	88.9	57.5
	+ SNAP	67.8	87.8	69.7	54.7	36.2	42.4	62.4	70.6	87.8	88.8	58.7
RN-50												
CC3M	SigLIP [75]	50.9 _{.1}	70.0 _{.2}	46.0 _{.2}	45.6 _{.1}	17.0 _{.1}	20.9 _{.2}	49.5 _{.2}	51.3 _{.2}	73.5 _{.3}	71.5 _{.2}	47.2 _{.2}
	CLIP [48]	56.1 _{.1}	74.0 _{.3}	52.3 _{.3}	50.0 _{.3}	24.6 _{.3}	31.1 _{.3}	53.0 _{.3}	62.4 _{.2}	80.4 _{.3}	80.4 _{.3}	50.0 _{.2}
	+ SNAP	56.7 _{.3}	74.1 _{.1}	52.8 _{.1}	50.8 _{.1}	25.3 _{.3}	31.8 _{.1}	54.3 _{.2}	63.6 _{.2}	81.1 _{.1}	80.8 _{.1}	51.4 _{.3}
	FLIP [29]	56.7 _{.2}	75.5 _{.1}	53.6 _{.3}	49.0 _{.1}	24.6 _{.1}	31.0 _{.2}	54.4 _{.2}	62.2 _{.3}	80.1 _{.1}	78.8 _{.1}	50.0 _{.3}
	+ SNAP	56.8 _{.1}	76.1 _{.1}	54.0 _{.1}	49.6 _{.3}	25.4 _{.2}	31.7 _{.2}	55.8 _{.2}	63.1 _{.2}	80.7 _{.2}	80.0 _{.1}	50.7 _{.1}

6 Discussion & Limitations

The failure modes we identify arise from the dual-modality structure of contrastive learning rather than from a CLIP-specific architectural choice. We therefore expect the design principles of SNAP to be relevant to other dual-encoder frameworks using representation-space synthetic negatives. Our evaluation is limited to CC3M and CC12M due to computational constraints; scaling to larger datasets such as LAION-400M remains future work.

Acknowledgements

We acknowledge the computational resources and support provided by the Imperial College Research Computing Service (<http://doi.org/10.14469/hpc/2232>), which enabled our experiments.

References

1. Beaumont, R.: img2dataset: Easily turn large sets of image urls to an image dataset. <https://github.com/rom1504/img2dataset> (2021)
2. Bossard, L., Guillaumin, M., Van Gool, L.: Food-101 – Mining Discriminative Components with Random Forests. In: Eur. Conf. Comput. Vis. (ECCV). pp. 446–461 (2014)
3. Cao, A., Wei, X., Ma, Z.: FLAME: Frozen Large Language Models Enable Data-Efficient Language-Image Pre-training. In: CVPR (2025)
4. Changpinyo, S., Sharma, P., Ding, N., Soricut, R.: Conceptual 12M: Pushing Web-Scale Image-Text Pre-Training To Recognize Long-Tail Visual Concepts. In: CVPR. pp. 3558–3568 (June 2021)
5. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: ICML. pp. 1597–1607. PMLR (2020)
6. Cherti, M., Beaumont, R.: CLIP benchmark (Nov 2022). <https://doi.org/10.5281/zenodo.15403103>, <https://doi.org/10.5281/zenodo.15403103>
7. Cimpoi, M., Maji, S., Kokkinos, I., Vedaldi, A.: Describing Textures in the Wild. In: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR). pp. 3606–3613 (2014)
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: CVPR. pp. 248–255 (2009). <https://doi.org/10.1109/CVPR.2009.5206848>
9. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In: International Conference on Learning Representations (2021), <https://openreview.net/forum?id=YicbFdNTTy>
10. Fan, L., Krishnan, D., Isola, P., Katabi, D., Tian, Y.: Improving CLIP Training with Language Rewrites. In: NeurIPS (2023)
11. Fei-Fei, L., Fergus, R., Perona, P.: Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories. *Computer Vision and Image Understanding* **106**(1), 59–70 (2007)
12. Giakoumoglou, N., Floros, A., Papadopoulos, K.M., Stathaki, T.: Unsupervised Training of Vision Transformers with Synthetic Negatives. In: Second Workshop on Visual Concepts (2025), <https://openreview.net/forum?id=dg8Fua0KnC>
13. Giakoumoglou, N., Stathaki, T.: SynCo: Synthetic Hard Negatives for Contrastive Visual Representation Learning (2024), <https://arxiv.org/abs/2410.02401>
14. Giakoumoglou, N., Stathaki, T., Gkelias, A.: A Review on Discriminative Self-supervised Learning Methods in Computer Vision (2025), <https://arxiv.org/abs/2405.04969>
15. Giakoumoglou, N., Floros, A., Papadopoulos, K.M., Stathaki, T.: ViTAMINS: An Empirical Study of Training Self-Supervised Vision Transformers with Synthetic Hard Negatives (2026), <https://arxiv.org/abs/2609.01041>

16. Goel, S., Bansal, H., Bhatia, S., Rossi, R.A., Vinay, V., Grover, A.: Cyclip: Cyclic contrastive language-image pretraining (2022), <https://arxiv.org/abs/2205.14459>
17. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and Harnessing Adversarial Examples (2015), <https://arxiv.org/abs/1412.6572>
18. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality reduction by learning an invariant mapping. In: CVRP. vol. 2, pp. 1735–1742. IEEE (2006)
19. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. CVPR pp. 770–778 (2016), <https://api.semanticscholar.org/CorpusID:206594692>
20. Ilharco, G., Wortsman, M., Wightman, R., Gordon, C., Carlini, N., Taori, R., Dave, A., Shankar, V., Namkoong, H., Miller, J., Hajishirzi, H., Farhadi, A., Schmidt, L.: OpenCLIP (Jul 2021). <https://doi.org/10.5281/zenodo.5143773>, <https://doi.org/10.5281/zenodo.5143773>, if you use this software, please cite it as below.
21. Jia, C., Yang, Y., Xia, Y., Chen, Y.T., Parekh, Z., Pham, H., Le, Q.V., Sung, Y., Li, Z., Duerig, T.: Scaling up visual and vision-language representation learning with noisy text supervision (2021), <https://arxiv.org/abs/2102.05918>
22. Kalantidis, Y., Sariyildiz, M.B., Pion, N., Weinzaepfel, P., Larlus, D.: Hard Negative Mixing for Contrastive Learning. In: NeurIPS (2020)
23. Kornblith, S., Shlens, J., Le, Q.V.: Do Better ImageNet Models Transfer Better? (2019), <https://arxiv.org/abs/1805.08974>
24. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3D Object Representations for Fine-Grained Categorization. In: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR). pp. 554–561 (2013)
25. Krizhevsky, A.: Learning Multiple Layers of Features from Tiny Images. Tech. Rep. 0, University of Toronto (2009)
26. Lai, Z., Zhang, H., Zhang, B., Wu, W., Bai, H., Timofeev, A., Du, X., Gan, Z., Shan, J., Chuah, C.N., Yang, Y., Cao, M.: VeCLIP: Improving CLIP Training via Visual-enriched Captions (2024)
27. Lavoie, S., Kirichenko, P., Ibrahim, M., Assran, M., Wilson, A.G., Courville, A., Ballas, N.: Modeling caption diversity in contrastive vision-language pretraining. In: ICML (2024), <https://openreview.net/forum?id=iaV2fU6Dif>
28. Li, Y., Liang, F., Zhao, L., Cui, Y., Ouyang, W., Shao, J., Yu, F., Yan, J.: Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. In: ICLR (2022), <https://openreview.net/forum?id=zq1iJkNk3uN>
29. Li, Y., Fan, H., Hu, R., Feichtenhofer, C., He, K.: Scaling language-image pre-training via masking (2023), <https://arxiv.org/abs/2212.00794>
30. Liang, W., Zhang, Y., Kwon, Y., Yeung, S., Zou, J.: Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning (2022), <https://arxiv.org/abs/2203.02053>
31. Lin, T.Y., Maire, M., Belongie, S.J., Bourdev, L.D., Girshick, R.B., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common Objects in Context. CoRR **abs/1405.0312** (2014), <http://dblp.uni-trier.de/db/journals/corr/corr1405.html/LinMBHPRDZ14>
32. Liu, Y., Li, X., Wang, Z., Zhao, B., Xie, C.: CLIPS: An Enhanced CLIP Framework for Learning with Synthetic Captions. arXiv preprint arXiv:2411.16828 (2024)
33. Liu, Y., Wang, K., Shao, W., Luo, P., Qiao, Y., Shou, M.Z., Zhang, K., You, Y.: MLLMs-Augmented Visual-Language Representation Learning (2024), <https://arxiv.org/abs/2311.18765>

34. Loshchilov, I., Hutter, F.: Decoupled Weight Decay Regularization. In: ICLR. OpenReview.net (2019), <https://openreview.net/forum?id=Bkg6RiCqY7>
35. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards Deep Learning Models Resistant to Adversarial Attacks. In: ICLR (2018), <https://openreview.net/forum?id=rJzIBfZAb>
36. Maji, S., Rahtu, E., Kannala, J., Blaschko, M.B., Vedaldi, A.: Fine-Grained Visual Classification of Aircraft. arXiv **abs/1306.5151** (2013)
37. Maninis, K.K., Chen, K., Ghosh, S., Karpur, A., Chen, K., Xia, Y., Cao, B., Salz, D., Han, G., Dlabal, J., Gnanapragasam, D., Seyedhosseini, M., Zhou, H., Araujo, A.: TIPS: Text-Image Pretraining with Spatial Awareness. In: ICLR (2025)
38. Mu, N., Kirillov, A., Wagner, D., Xie, S.: Slip: Self-supervision meets language-image pre-training. In: European Conference on Computer Vision. pp. 523–540. Springer (2022)
39. Naeem, M.F., Xian, Y., Zhai, X., Hoyer, L., Gool, L.V., Tombari, F.: Silc: Improving vision language pretraining with self-distillation (2023), <https://arxiv.org/abs/2310.13355>
40. Nilsback, M., Zisserman, A.: Automated Flower Classification by Use of Color, Shape, and Texture Features. *Machine Vision and Applications* **18**(1), 27–39 (2008)
41. Oh, C., So, J., Byun, H., Lim, Y., Shin, M., Jeon, J.J., Song, K.: Geodesic multi-modal mixup for robust fine-tuning. In: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (eds.) *Advances in Neural Information Processing Systems*. vol. 36, pp. 52326–52341. Curran Associates, Inc. (2023). <https://doi.org/10.52202/075280-2278>, https://proceedings.neurips.cc/paper_files/paper/2023/file/a45296e83b19f656392e0130d9e53cb1-Paper-Conference.pdf
42. van den Oord, A., Li, Y., Vinyals, O.: Representation Learning with Contrastive Predictive Coding (2019), <https://arxiv.org/abs/1807.03748>
43. Parkhi, O.M., Vedaldi, A., Zisserman, A., Jawahar, C.V.: Cats and Dogs. In: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR). pp. 3498–3505 (2012)
44. Patel, M., Kusumba, A., Cheng, S., Kim, C., Gokhale, T., Baral, C., Yang, Y.: TripletCLIP: Improving Compositional Reasoning of CLIP via Synthetic Vision-Language Negatives. *NeurIPS* (2024)
45. Pham, H., Dai, Z., Ghiasi, G., Kawaguchi, K., Liu, H., Yu, A.W., Yu, J., Chen, Y.T., Luong, M.T., Wu, Y., Tan, M., Le, Q.V.: Combined scaling for zero-shot transfer learning. *Neurocomputing* **555**, 126658 (2023). <https://doi.org/https://doi.org/10.1016/j.neucom.2023.126658>, <https://www.sciencedirect.com/science/article/pii/S0925231223007816>
46. Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., Rombach, R.: Sdxl: Improving latent diffusion models for high-resolution image synthesis (2023), <https://arxiv.org/abs/2307.01952>
47. Radenovic, F., Dubey, A., Kadian, A., Mihaylov, T., Vandenhende, S., Patel, Y., Wen, Y., Ramanathan, V., Mahajan, D.: Filtering, distillation, and hard negatives for vision-language pre-training. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 6967–6977 (June 2023)
48. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning Transferable Visual Models From Natural Language Supervision. In: Meila, M., Zhang, T. (eds.) *ICML. Proceedings of Machine Learning Research*, vol. 139, pp. 8748–8763. PMLR (18–24 Jul 2021), <https://proceedings.mlr.press/v139/radford21a.html>
49. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving language understanding by generative pre-training. Tech. rep., OpenAI (2018), <https://arxiv.org/abs/1802.07729>

- [//cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf](https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf)
50. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language Models are Unsupervised Multitask Learners. Tech. rep., OpenAI (2019), https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf
 51. Ranasinghe, K., McKinzie, B., Ravi, S., Yang, Y., Toshev, A., Shlens, J.: Perceptual grouping in contrastive vision-language models (2023), <https://arxiv.org/abs/2210.09996>
 52. Robinson, J., Chuang, C.Y., Sra, S., Jegelka, S.: Contrastive learning with hard negative samples. arXiv preprint arXiv:2010.04592 (2020)
 53. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision* **115**(3), 211–252 (Apr 2015). <https://doi.org/10.1007/s11263-015-0816-y>, <http://dx.doi.org/10.1007/s11263-015-0816-y>
 54. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). p. 815–823. IEEE (Jun 2015). <https://doi.org/10.1109/cvpr.2015.7298682>, <http://dx.doi.org/10.1109/CVPR.2015.7298682>
 55. Sharma, P., Ding, N., Goodman, S., Soricut, R.: Conceptual Captions: A Cleaned, Hypernymed, Image Alt-text Dataset For Automatic Image Captioning. In: *Proceedings of ACL* (2018)
 56. Singh, A., Hu, R., Goswami, V., Couairon, G., Galuba, W., Rohrbach, M., Kiela, D.: FLAVA: A foundational language and vision alignment model. In: *CVPR* (2022)
 57. Sohn, K.: Improved deep metric learning with multi-class n-pair loss objective. In: Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., Garnett, R. (eds.) *Advances in Neural Information Processing Systems*. vol. 29. Curran Associates, Inc. (2016), https://proceedings.neurips.cc/paper_files/paper/2016/file/6b180037abbebea991d8b1232f8a8ca9-Paper.pdf
 58. Sun, Q., Fang, Y., Wu, L., Wang, X., Cao, Y.: Eva-clip: Improved training techniques for clip at scale (2023), <https://arxiv.org/abs/2303.15389>
 59. Sun, Q., Wang, J., Yu, Q., Cui, Y., Zhang, F., Zhang, X., Wang, X.: Eva-clip-18b: Scaling clip to 18 billion parameters (2024), <https://arxiv.org/abs/2402.04252>
 60. Thrush, T., Jiang, R., Bartolo, M., Singh, A., Williams, A., Kiela, D., Ross, C.: Winoground: Probing vision and language models for visio-linguistic compositionality. In: *CVPR* (2022)
 61. Tschannen, M., Gritsenko, A., Wang, X., Naeem, M.F., Alabdulmohsin, I., Parthasarathy, N., Evans, T., Beyer, L., Xia, Y., Mustafa, B., Hénaff, O., Harmen, J., Steiner, A., Zhai, X.: Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features (2025), <https://arxiv.org/abs/2502.14786>
 62. Verma, V., Lamb, A., Beckham, C., Najafi, A., Mitliagkas, I., Courville, A., Lopez-Paz, D., Bengio, Y.: Manifold Mixup: Better Representations by Interpolating Hidden States (2019), <https://arxiv.org/abs/1806.05236>
 63. Wu, C.Y., Manmatha, R., Smola, A.J., Krähenbühl, P.: Sampling matters in deep embedding learning (2018), <https://arxiv.org/abs/1706.07567>
 64. Wu, W., Zheng, K., Ma, S., Lu, F., Guo, Y., Zhang, Y., Chen, W., Guo, Q., Shen, Y., Zha, Z.J.: LoTLIP: Improving Language-Image Pre-training for Long Text Understanding. In: *NeurIPS* (2024), <https://openreview.net/forum?id=pc4GSBi1Hx>

65. Xiao, B., Wu, H., Xu, W., Dai, X., Hu, H., Lu, Y., Zeng, M., Liu, C., Yuan, L.: Florence-2: Advancing a unified representation for a variety of vision tasks. arXiv preprint arXiv:2311.06242 (2023)
66. Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A.: SUN Database: Large-scale Scene Recognition from Abbey to Zoo. In: IEEE Conf. Comput. Vis. Pattern Recog. (CVPR). pp. 3485–3492 (2010)
67. Yang, J., Li, C., Zhang, P., Xiao, B., Liu, C., Yuan, L., Gao, J.: Unified contrastive learning in image-text-label space (2022)
68. Yang, K., Deng, J., An, X., Li, J., Feng, Z., Guo, J., Yang, J., Liu, T.: ALIP: Adaptive Language-Image Pre-training with Synthetic Caption (2023)
69. Yao, L., Huang, R., Hou, L., Lu, G., Niu, M., Xu, H., Liang, X., Li, Z., Jiang, X., Xu, C.: Filip: Fine-grained interactive language-image pre-training (2021), <https://arxiv.org/abs/2111.07783>
70. Young, P., Lai, A., Hodosh, M., Hockenmaier, J.: 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)
71. Yu, J., Wang, Z., Vasudevan, V., Yeung, L., Seyedhosseini, M., Wu, Y.: CoCa: Contrastive Captioners are Image-Text Foundation Models. TMLR (2022), <https://openreview.net/forum?id=Ee277P3AYC>
72. Yuan, L., Chen, D., Chen, Y.L., Codella, N., Dai, X., Gao, J., Hu, H., Huang, X., Li, B., Li, C., Liu, C., Liu, M., Liu, Z., Lu, Y., Shi, Y., Wang, L., Wang, J., Xiao, B., Xiao, Z., Yang, J., Zeng, M., Zhou, L., Zhang, P.: Florence: A new foundation model for computer vision (2021), <https://arxiv.org/abs/2111.11432>
73. Yuksekgonul, M., Bianchi, F., Kalluri, P., Jurafsky, D., Zou, J.: When and why Vision-Language Models behave like Bags-of-Words, and what to do about it? In: ICLR (2023), <https://openreview.net/forum?id=KRLUvvh8uaX>
74. Yuksekgonul, M., Bianchi, F., Kalluri, P., Jurafsky, D., Zou, J.: When and why vision-language models behave like bags-of-words, and what to do about it? In: International Conference on Learning Representations (2023), <https://openreview.net/forum?id=KRLUvvh8uaX>
75. Zhai, X., Mustafa, B., Kolesnikov, A., Beyer, L.: Sigmoid Loss for Language Image Pre-Training (2023), <https://arxiv.org/abs/2303.15343>
76. Zhai, X., Wang, X., Mustafa, B., Steiner, A., Keysers, D., Kolesnikov, A., Beyer, L.: LiT: Zero-Shot Transfer with Locked-image text Tuning. In: CVPR. pp. 18123–18133 (June 2022)
77. Zhang, B., Zhang, P., Dong, X., Zang, Y., Wang, J.: Long-clip: Unlocking the long-text capability of clip. arXiv preprint arXiv:2403.15378 (2024)
78. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond Empirical Risk Minimization. In: ICLR (2018), <https://openreview.net/forum?id=r1Ddp1-Rb>
79. Zhang, L., Awal, R., Agrawal, A.: Contrasting Intra-Modal and Ranking Cross-Modal Hard Negatives to Enhance Visio-Linguistic Fine-grained Understanding. arXiv preprint arXiv:2306.08832 (2023)
80. Zheng, K., Zhang, Y., Wu, W., Lu, F., Ma, S., Jin, X., Chen, W., Shen, Y.: DreamLIP: Language-Image Pre-training with Long Captions. In: ECCV (2024)
81. Zhou, Y., Qi, L., Wang, J., Feng, M.H., Zhang, W., Yang, Z., Liu, W., Wang, Y.: Non-contrastive learning meets language-image pre-training. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8913–8923 (June 2023)

A Dataset Details

A.1 Pretraining Datasets

We pretrain on two image-text datasets of varying scale: CC3M [55] and CC12M [4]. Both datasets were downloaded using `img2dataset` [1]. Due to image link rot, the versions we obtained contain slightly fewer samples than the original releases, which may lead to minor performance differences compared to models trained on the full datasets. The same training configuration is used for both datasets (*cf.* Sec. B.2).

CC3M: This dataset comprises 3.3 million image-text pairs extracted from 5 billion webpages, with image descriptions derived from the HTML `alt-text` attribute. The version we used contains 1,666,040 unique samples.

CC12M: Following a similar collection procedure as CC3M but with more relaxed filters, CC12M consists of 12.4 million image-text pairs covering a wider range of topics and visual concepts. The version we acquired contains 7,277,079 samples.

A.2 Downstream Datasets

We evaluate zero-shot retrieval on Flickr30k [70] and MSCOCO [31], and classification on ImageNet [8] and 10 downstream datasets. Details are provided in Tab. 8.

Table 8: Details of downstream classification datasets.

Dataset	Metric	Categories	Train Size	Test Size
Food-101 [2]	Accuracy	101	75,750	25,250
CIFAR-10 [25]	Accuracy	10	50,000	10,000
CIFAR-100 [25]	Accuracy	100	50,000	10,000
SUN397 [66]	Accuracy	397	19,850	19,850
Stanford Cars [24]	Accuracy	196	8,144	8,041
FGVC Aircraft [36]	Mean per class	100	6,667	3,333
DTD [7]	Accuracy	47	3,760	1,880
Oxford Pets [43]	Mean per class	37	3,680	3,669
Caltech-101 [11]	Mean per class	102	3,060	6,085
Oxford Flowers [40]	Mean per class	102	2,040	6,149
ImageNet [8]	Accuracy	1,000	1,281,167	50,000
Flickr30k [70]	Recall@ k	–	29,000	1,000
MSCOCO [31]	Recall@ k	–	113,287	5,000

B Implementation Details

We build on the OpenCLIP [20] codebase and evaluate using CLIP Benchmark [6].

B.1 Encoders

We use ViT-B/16, ViT-B/32 [9], and ResNet-50 (RN-50) as vision encoders. All models are pretrained on CC3M, while ViT-B/16 and ViT-B/32 are additionally trained on CC12M. Following CLIP [48], the text encoder is a 12-layer Transformer with vocabulary size 49,408 and maximum context length 77.

B.2 Hyperparameters

We use the same training configuration across all architectures and both pretraining datasets. The only difference between architectures is the image encoder itself; all other hyperparameters, including those for synthetic negative generation, are kept identical. Detailed configurations are provided in Tabs. 9a and 9b.

Table 9: Hyperparameters. (a) CLIP/FLIP training configuration, identical for ViT-B/16, ViT-B/32, and RN-50 on both CC3M and CC12M. (b) SNAP synthetic negative generation parameters.

(a) CLIP/FLIP hyperparameters.		(b) SNAP hyperparameters.	
Config	Value	Config	Value
Image resolution	224×224	Top- N hard negatives	256
Max text length	77	Negatives per strategy	32
Embedding dimension d	512	Number of strategies	2
Batch size $ \mathcal{B} $	4,096	Total syn. negatives / query	64
Optimizer	AdamW [34]	Temperature τ	0.07 (constant)
Learning rate	5×10^{-4}	<i>Strategy hyperparameters</i>	
Weight decay	0.2	Interpolation α_k	$\sim \mathcal{U}(0, 0.5)$
Adam (β_1, β_2)	(0.9, 0.98)	Extrapolation β_k	$\sim \mathcal{U}(1, 1.5)$
Adam ϵ	1×10^{-8}	Mixup γ_k	$\sim \mathcal{U}(0, 1)$
LR schedule	Cosine decay	Noise std σ	0.01
Warmup steps	10,000	Gradient step δ	0.01
Total epochs	32	Adversarial η	0.01
Temperature τ (init)	0.07 (learnable)		
GPUs	$4 \times$ NVIDIA A100		

Distributed mining and gradient flow. Each GPU selects hard negatives for its local queries from the globally gathered current batch, excluding the matched positives. Gathering is performed without gradients, and source embeddings are detached during synthesis. Gradients from synthetic-negative terms flow through the query embeddings.

C Related Work

We briefly contrast SNAP with the most closely related vision-language methods that incorporate synthetic data during pretraining (*cf.* Fig. 7). NegCLIP [73]

constructs hard negative captions by swapping linguistic elements (attributes, relations, objects) in the original text and retrieves positive images from the three nearest neighbors in the training set. LaCLIP [10] augments the text side with synthetic positive captions generated by large language models such as ChatGPT, Bard, and LLaMA. DreamLIP [80] follows a similar philosophy but leverages multimodal LLMs (ShareGPT4V, LLaVA, InstructBLIP) to produce richer positive captions grounded in visual content. TripletCLIP [44] operates on both modalities: it generates hard negative captions via in-context learning with Mistral-7B-Instruct-v0.2 and then synthesizes corresponding negative images using SDXL-Turbo conditioned on those captions. All four methods operate in the input space and depend on external foundation models pretrained on large-scale data, introducing both computational overhead and potential data leakage from the generative models into downstream evaluation. In contrast, SNAP generates hard negatives directly in the representation space from in-batch embeddings alone, requires no external models or additional data, and applies to both modalities symmetrically with less than 10% training time overhead.

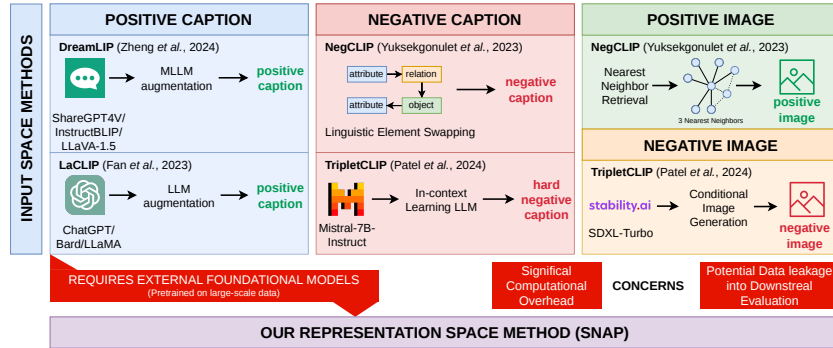


Fig. 7: Comparison of vision-language pretraining methods using synthetic data. Top: Existing **input-space methods** rely on external foundation models (e.g., LLMs for text, Diffusion for images) to generate synthetic data, incurring significant computational overhead and risk of data leakage. Bottom: **SNAP** operates directly in the **representation space**, using in-batch embeddings to symmetrically generate hard negatives for both modalities.

D Further Ablations and Results

We conduct additional ablation studies on CC3M with ViT-B/16, reporting zero-shot classification accuracy on IN-val and IN-v2.

Table 10: Batch size ablation.

Batch Size	Method	IN-val	IN-v2
1024	CLIP	9.2	7.2
	+ SNAP	9.5	8.3
2048	CLIP	9.8	7.8
	+ SNAP	10.1	8.9
4096	CLIP	10.4	8.4
	+ SNAP	11.0	9.6

Table 11: Synthetic negatives ablation.

$ \mathcal{S} $	$ \mathcal{S}_{\text{txt}} $	IN-val	IN-v2
32	16	10.4	8.4
64	32	10.5	8.5
128	64	11.0	9.6
256	128	10.7	8.9
512	256	9.1	7.0
1024	512	4.3	3.2
2048	1024	3.1	2.0

D.1 Batch Size

We evaluate SNAP at batch sizes of 1024, 2048, and 4096 (Tab. 10). SNAP outperforms CLIP at each tested batch size, although its performance decline as batch size decreases is slightly larger. These results demonstrate benefits across the tested batch sizes. The hard-negative pool remains fixed at 256 candidates, so its proportion of the available negatives increases as batch size decreases (*cf.* Tab. 10).

D.2 Number of Synthetic Negatives

We vary the total number of synthetic negatives $|\mathcal{S}| = |\mathcal{S}_{\text{txt}}| + |\mathcal{S}_{\text{img}}|$, split equally between $s = 3, 4$. As shown in Tab. 11, a small number of negatives ($|\mathcal{S}| \leq 64$) yields performance similar to the CLIP baseline, as the contrastive signal is not sufficiently challenging. While $|\mathcal{S}| \geq 1024$ significantly degrades performance. At a batch size of 4096, using 512 or 1024 synthetic negatives—representing 12.5% and 25.0% of the real negative count, respectively—makes the contrastive task too difficult to solve.

Synthesis controls. All synthesis controls use fixed $\tau = 0.07$. The single-strategy variants each generate 64 synthetic negatives per query and direction, matching the total count of full SNAP. The hard-negative weighting control follows the importance-sampling approach of DiHT [47].

D.3 Paired Seed-wise Improvements

Tab. 12 reports paired seed-wise improvements for the principal CC3M ImageNet comparisons. SNAP improves accuracy for every evaluated seed across all three backbones and both baselines, showing that the average gains are consistent across the three runs.

Table 12: Paired seed-wise improvements on CC3M. ImageNet zero-shot accuracy summaries, with standard deviations shown as subscripts. Each paired gain Δ is the accuracy with SNAP minus the baseline accuracy for the same seed, in percentage points. The final column reports the mean and sample standard deviation of the three paired gains.

Backbone	Baseline	Accuracy		Paired gain Δ (pp)			
		Baseline	+SNAP	Seed 1	Seed 2	Seed 3	Mean $\pm$ std.
ViT-B/32	CLIP	8.0 _{.3}	8.2 _{.2}	+0.46	+0.02	+0.06	+0.18 $\pm$ 0.24
	FLIP	6.8 _{.3}	7.2 _{.1}	+0.10	+0.27	+0.74	+0.37 $\pm$ 0.33
ViT-B/16	CLIP	10.4 _{.2}	11.0 _{.2}	+1.07	+0.30	+0.37	+0.58 $\pm$ 0.43
	FLIP	10.3 _{.3}	11.0 _{.2}	+0.88	+1.16	+0.12	+0.72 $\pm$ 0.54
RN-50	CLIP	13.6 _{.1}	14.7 _{.2}	+0.99	+1.05	+1.32	+1.12 $\pm$ 0.18
	FLIP	13.7 _{.1}	14.9 _{.1}	+1.17	+1.25	+1.15	+1.19 $\pm$ 0.05

Table 13: Compositional evaluation. Scores (%) for ViT-B/16 pretrained on CC12M.

Method	ARO		Winoground		
	Relation	Attribute	Text	Image	Group
CLIP	51.2	52.1	24.5	9.8	6.3
+ SNAP	54.8	56.3	25.2	9.6	6.5

D.4 Compositional Evaluation

We evaluate CLIP and SNAP with ViT-B/16 pretrained on CC12M on ARO [74] and Winoground [60] (Tab. 13). SNAP improves ARO relation and attribute accuracy by 3.6 and 4.2 percentage points, respectively, while Winoground scores remain similar.

E Future Work

An immediate direction is to study SNAP at larger scales: training on web-scale datasets, exploring deeper and wider architectures, and integrating with recent advances in text and image encoders. The top-similarity hard-negative pool may also contain semantically compatible but unmatched examples, particularly in web-crawled data. SNAP does not explicitly filter such false negatives; incorporating semantic filtering or auditing their prevalence is left for future work.

F Limitations

Our experiments are conducted on CC3M and CC12M, which are relatively small-scale compared to proprietary datasets used by methods such as CLIP

(400M pairs) and ALIGN (1B pairs). Training on larger datasets was not feasible given our computational budget of $4\times$ A100 GPUs, where a single CC12M run already requires multiple days. Scaling to larger data and model sizes, while remaining environmentally conscious, is an important direction for future work. We observe broad improvements across the evaluated settings; whether these benefits persist at larger data and model scales remains to be tested.

G Discussion

SNAP operates as a drop-in augmentation to any contrastive framework that uses an InfoNCE-style loss with a softmax denominator over negatives. Since synthetic negatives are generated from in-batch embeddings alone, the method is agnostic to the choice of encoders, data, and training procedure. We expect SNAP to be applicable beyond the CLIP [48] and FLIP [29] frameworks evaluated here, including methods such as SLIP [38], DeCLIP [28], FLIP [29], LiT [76], LaCLIP [10], DreamLIP [80], CyCLIP [16], and TripletCLIP [44], all of which build on the same InfoNCE-style contrastive objective.

H Reproducibility

We build on the OpenCLIP [20] codebase for all training and use CLIP Benchmark [6] for all downstream evaluations. Pretraining datasets were downloaded using `img2dataset` [1]. All experiments use identical hyperparameters across architectures and datasets (*cf.* Tab. 9). Training was performed on $4\times$ NVIDIA A100 80GB GPUs using PyTorch with mixed-precision (FP16) training. We use the same set of random seeds across all runs for reproducibility. SNAP requires no external models, no additional data, and no offline preprocessing—the only modification to the standard OpenCLIP training loop is the synthetic negative generation module, which operates entirely in the representation space during the forward pass as detailed in Algorithms 1 and 2. We will release our code and pretrained model weights upon acceptance to facilitate reproduction of all reported results.

I Algorithm

Algorithm 1 provides the pseudo-code of SNAP, followed by the detailed implementation of the two specific generation strategies, $s = 3$ and $s = 4$, used in our approach (Algorithm 2).

Algorithm 1 Pseudocode of SNAP in a PyTorch-like style.

```

# f_v, f_t: visual and text encoders
# t: fixed temperature parameter
# n_hard: pool size for sampling anchors (e.g., 256)
# s3, s4: number of synthetic negatives per strategy (default: 32 each)

# t: fixed temperature
# n_hard: hard-negative pool size
# s3, s4: synthetic negatives per strategy and direction

def synthetic_logits(query, candidates, labels):
    scores = query @ candidates.T
    scores = scores.clone()
    scores[arange(query.shape[0]), labels] = -inf
    idxs = topk(scores, k=n_hard, dim=1).indices

    mix = generate_s3(candidates, idxs, s3)
    noise = generate_s4(candidates, idxs, s4)
    synthetic = cat([mix, noise], dim=1)
    return einsum("nc,nkc->nk", query, synthetic)

for images, texts in loader:
    v = normalize(f_v(images), dim=-1)
    u = normalize(f_t(texts), dim=-1)

    # Global current-batch candidates, detached for synthesis
    all_v = gather_no_grad(v.detach())
    all_u = gather_no_grad(u.detach())

    # Positive indices in the globally gathered batch
    # Assumes equal local batch sizes and rank-ordered gathering
    labels = rank * v.shape[0] + arange(v.shape[0])

    syn_i2t = synthetic_logits(v, all_u, labels)
    syn_t2i = synthetic_logits(u, all_v, labels)

    # Obtain base logits with the existing CLIP gradient handling
    base_i2t, base_t2i = clip_logits(v, u)

    logits_i2t = cat([base_i2t, syn_i2t], dim=1) / t
    logits_t2i = cat([base_t2i, syn_t2i], dim=1) / t

    loss = 0.5 * (
        cross_entropy(logits_i2t, labels)
        + cross_entropy(logits_t2i, labels)
    )

    optimizer.zero_grad()
    loss.backward()
    optimizer.step()

```

Algorithm 2 SNAP Generation Strategies in a PyTorch-like style.

```
def generate_s3(t_feat, idxs_hard, num_neg, gamma=1.0):
    """Mixup_between_two_hard_negatives."""
    # sample two sets of indices from the hard pool
    idx1 = randint(0, n_hard, size=(batch_size, num_neg))
    idx2 = randint(0, n_hard, size=(batch_size, num_neg))

    h1 = t_feat[gather(idxs_hard, 1, idx1)]
    h2 = t_feat[gather(idxs_hard, 1, idx2)]

    lam = rand(batch_size, num_neg, 1) * gamma
    s3 = lam * h1 + (1 - lam) * h2
    return normalize(s3, dim=-1)

def generate_s4(t_feat, idxs_hard, num_neg, sigma=0.01):
    """Noise_injection_into_a_hard_negative."""
    idx = randint(0, n_hard, size=(batch_size, num_neg))
    h = t_feat[gather(idxs_hard, 1, idx)]

    noise = randn_like(h) * sigma
    return normalize(h + noise, dim=-1)
```
