

SNAP: Synthetically Negative Augmented Pretraining for Vision-Language Models

Nikolaos Giakoumoglou^{1*,†}, Paschalis Giakoumoglou^{2†}, Andreas Floros^{1*}, Kleonthis Marios Papadopoulos^{1*}, Tania Stathaki¹

¹Imperial College London, ²Centre for Research and Technology Hellas

*Core Team, [†]These authors contributed equally to this work.

Vision-language contrastive pretraining relies on large batches of randomly sampled image-text pairs to provide negative examples. Prior approaches address this by generating hard negatives in the input space—rewriting captions with LLMs or synthesizing images with diffusion models—but incur substantial computational overhead and typically augment only one modality. Synthetic hard negatives generated in the representation space have proven effective for unimodal self-supervised learning, but extending them to vision-language models that align two distinct modalities via an InfoNCE objective is *not* straightforward. We identify two failure modes: cross-modal synthetic negatives fall into the modality gap and are trivially rejected, while intra-modal negatives involving the positive pair suffer from positive leakage that sends contradictory gradients. Both failure modes additionally cause the learnable temperature to diverge. We propose SNAP (**S**ynthetically **N**egative **A**ugmented **P**retraining), 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 9% 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.

Date: March 31, 2026

Correspondence: Nikolaos Giakoumoglou <nikos@imperial.ac.uk>

IMPERIAL

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 (Deng et al., 2009). 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 (van den Oord et al., 2019; Hadsell et al., 2006). Seminal works CLIP (Radford et al., 2021), ALIGN (Jia et al., 2021), and BASIC (Pham et al., 2023) demonstrated remarkable zero-shot transfer capabilities by training on hundreds of millions of image-text pairs. However, the standard formulation treats all in-batch negatives equally: at each optimization step, image-text pairs are *randomly* sampled from the training set, and the model rarely encounters highly similar negative pairs that would force fine-grained discrimination. Contrastive learning benefits substantially from *hard* negatives—samples that are difficult to distinguish from the anchor (Robinson et al., 2020; Kalantidis et al., 2020)—yet the random sampling strategy of CLIP-style training seldom provides such signal.

Prior work has explored hard negatives primarily through augmentation in the **input space**: rule-based text perturbations that swap linguistic elements (Yuksekgonul et al., 2023; Zhang et al., 2023), LLM-based caption rewrites (Fan et al., 2023; Patel et al., 2024), or diffusion-generated negative images (Patel et al., 2024). While effective for compositional understanding, these approaches incur significant computational overhead and typically address only one modality at a time (*cf.* Section 2). In unimodal self-supervised learning, synthetic hard negatives generated directly in the representation space have proven highly effective (Kalantidis et al., 2020; Giakoumoglou and Stathaki, 2024; Giakoumoglou et al., 2025a,b).

A natural question arises:

Can we bring representation-space synthetic negatives to vision-language pretraining?

As we show in this paper, the answer is *yes*, but the extension is **not straightforward**. Unlike a single-modality setting where all representations inhabit one homogeneous space, vision-language models align two distinct modalities—images and text—within a shared embedding space. This dual-domain structure introduces failure modes absent in the unimodal case (Section 4.1). First, *cross-modal* synthetic negatives (*i.e.*, blends of image and text embeddings) produce vectors that fall into the modality gap (Liang et al., 2022), making them trivially distinguishable and uninformative. Second, even when restricting to *intra-modal* negatives, naïvely including the positive pair in the synthesis causes *positive leakage*: the synthetic “negative” retains genuine positive signal, sending contradictory gradients that confuse optimization. We further observe that both failure modes cause the learnable temperature parameter τ to diverge, destabilizing training entirely (Section 4.1.4). We validate these findings empirically in Figure 3, where strategies that avoid both failure modes ($s = 3$, $s = 4$) consistently outperform all alternatives. These observations motivate SNAP (**S**ynthetically **N**egative **A**ugmented **P**retraining), a method that generates intra-modal hard negatives which *never* involve the positive from either modality, thereby sidestepping both cross-modal geometric artifacts and intra-modal positive leakage (Section 4.2). SNAP builds on top of CLIP-like methods with an InfoNCE objective and can be applied to existing frameworks such as CLIP (Radford et al., 2021) and FLIP (Li et al., 2023).

We make the following **contributions**: (i) We show that extending synthetic negatives to vision-language pretraining is not straightforward, identifying two failure modes (Sections 4.1 and 4.1.4). (ii) Guided by these findings, 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 (Section 4.2). (iii) We conduct comprehensive evaluations across multiple architectures (ViT-B/16, ViT-B/32, RN-50), datasets (CC3M, CC12M), and downstream tasks—zero-shot retrieval (Table 5), zero-shot classification (Table 6), linear probe evaluation (Table 7)—demonstrating consistent improvements across all settings.

2 Related Work

2.1 Vision-Language Contrastive Pretraining

CLIP (Radford et al., 2021) and ALIGN (Jia et al., 2021) established the standard approach for vision-language contrastive pretraining: learning aligned image-text representations via a batch-level InfoNCE loss (van den Oord et al., 2019) over large-scale web-crawled data. LiT (Zhai et al., 2022) showed that freezing a pretrained vision encoder and training only the text tower can be surprisingly effective, while SigLIP (Zhai et al., 2023) replaced the softmax-based InfoNCE with a pairwise sigmoid formulation that removes the need for global normalization. BASIC (Pham et al., 2023) 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 for finer-grained alignment (Yao et al., 2021; Yang et al., 2022; Goel et al., 2022; Lavoie et al., 2024), integration of self-supervised objectives alongside contrastive alignment (Mu et al., 2022; Li et al., 2022; Naeem et al., 2023; Zhou et al., 2023; Maninis et al., 2025; Singh et al., 2022), architectural changes such as multi-tower designs or alternative encoder freezing strategies (Cao et al., 2025; Ranasinghe et al., 2023; Yu et al., 2022; Tschannen et al., 2025), training efficiency improvements through input masking (Li et al., 2023; Yang et al., 2023), scaling to larger models and datasets (Sun et al., 2023, 2024; Yuan et al., 2021; Xiao et al., 2023; Ilharco et al., 2021), and improved training signals via longer or synthetically generated captions (Zhang et al., 2024; Zheng et al., 2024; Fan et al., 2023; Wu et al., 2024; Liu et al., 2024a). SNAP is complementary to all these directions: it augments the InfoNCE contrastive loss with synthetic negatives in the representation space 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 (Schroff et al., 2015; Sohn, 2016; Wu et al., 2018; Giakoumoglou et al., 2025c). In **vision**, synthetic hard negatives generated in the representation space have been shown to significantly improve self-supervised representation learning (Kalantidis et al., 2020; Robinson et al., 2020; Giakoumoglou and Stathaki, 2024; Giakoumoglou et al., 2025a,b). In **vision-language**

Table 1 Comparison of vision-language methods on synthetic data generation. We indicate synthetic *negatives/positives*, synthesized *image* and *text* modalities, and *input* vs *representation space* generation.

Method	Synthetic		Modality		Generation Space
	Negatives	Positives	Image	Text	
CLIP (Radford et al., 2021)	–	–	–	–	None
NegCLIP (Yuksekgonul et al., 2023)	✓	–	–	✓	Input
TripletCLIP (Patel et al., 2024)	✓	–	✓	✓	Input
ALIP (Yang et al., 2023)	–	✓	–	✓	Input
CLIPS (Liu et al., 2024a)	–	✓	–	✓	Input
LaCLIP (Fan et al., 2023)	–	✓	–	✓	Input
DreamLIP (Zheng et al., 2024)	–	✓	–	✓	Input
SNAP (ours)	✓	–	✓	✓	Representation

learning, hard negative mining has been explored primarily on the text side, where negative captions are constructed by swapping linguistic elements (Yuksekgonul et al., 2023) or applying structured compositional perturbations (Zhang et al., 2023). However, these methods rely on explicit linguistic rules, operate exclusively on one modality, and do not produce hard negatives for the image side—mainly because semantic perturbations at the pixel level are far more difficult than sentence-level edits. SNAP addresses both limitations by extending representation-space synthetic negatives to the vision-language setting, generating hard negatives for *both* modalities without linguistic heuristics or pixel-level manipulation.

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 foundation models pretrained on billions of samples. On the text side, captions are rewritten or generated using LLMs (Fan et al., 2023; Patel et al., 2024), captioning models (Yang et al., 2023; Liu et al., 2024a), or multimodal LLMs (Zheng et al., 2024; Wu et al., 2024; Lai et al., 2024; Liu et al., 2024b). On the image side, TripletCLIP (Patel et al., 2024) generates hard negative images using text-to-image diffusion models (*e.g.*, SDXL-Turbo (Podell et al., 2023)) conditioned on LLM-generated captions. While effective, these methods introduce external knowledge from models trained on data that may overlap with or exceed the downstream evaluation benchmarks, making it difficult to disentangle genuine representational improvements from implicit data leakage. They also add substantial computational and storage overhead, as the generative models must be run either offline over the entire dataset or during training. SNAP avoids these issues entirely: synthetic negatives are generated in the representation space from in-batch embeddings alone, requiring no external models and no additional data. Table 1 compares SNAP with prior vision-language methods, highlighting our unique use of representation-space synthetic negatives for both modalities, which avoids the computational overhead and data leakage risks of input-space generation.

3 Preliminaries

In this section, we introduce the building blocks for integrating synthetic hard negatives into vision-language contrastive learning: the standard CLIP objective (Section 3.1), how synthetic negatives augment the contrastive loss (Section 3.2), and the generation strategies (Section 3.3). These preliminaries set up the notation for Section 4, where we analyze why naïvely applying these strategies fails and present 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 (Dosovitskiy et al., 2021) or a convnet (He et al., 2016)) and a text encoder g_ϕ with parameters ϕ (*e.g.*, a GPT-style transformer (Radford et al., 2018, 2019)). Each input sample $(\mathbf{x}_i, \mathbf{c}_i)$ consists of a matching 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

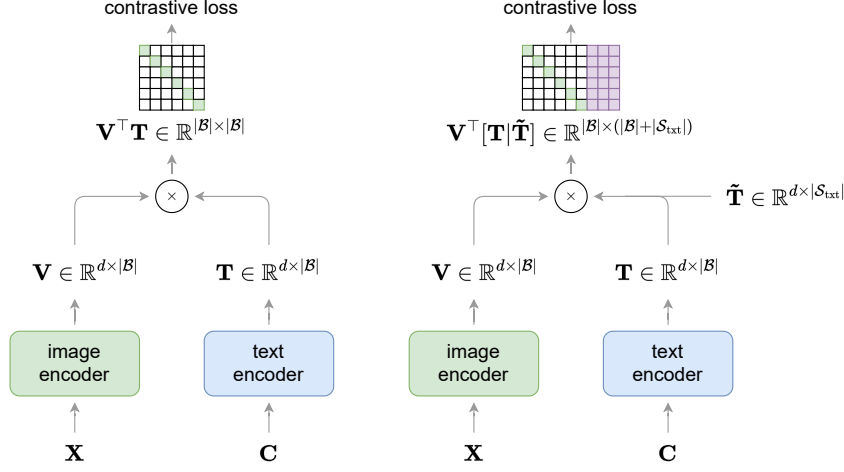


Figure 1 Architecture comparison for the i2t direction. (a) Standard CLIP encodes images \mathbf{X} and captions \mathbf{C} into embeddings $\mathbf{V} \in \mathbb{R}^{d \times |\mathcal{B}|}$ and $\mathbf{T} \in \mathbb{R}^{d \times |\mathcal{B}|}$, computing the similarity matrix $\mathbf{V}^\top \mathbf{T} \in |\mathcal{B}| \times |\mathcal{B}|$ using only batch negatives. (b) Our method generates synthetic text negatives $\tilde{\mathbf{T}} \in \mathbb{R}^{d \times |\mathcal{S}_{\text{txt}}|}$ directly in the representation space and concatenates them with batch embeddings to form an augmented similarity matrix $\mathbf{V}^\top [\mathbf{T}|\tilde{\mathbf{T}}] \in |\mathcal{B}| \times (|\mathcal{B}| + |\mathcal{S}_{\text{txt}}|)$. In the similarity matrices, green cells denote positive pairs, white cells denote batch negatives, and purple cells denote synthetic negatives. The t2i direction follows an analogous procedure with synthetic image negatives (*cf.* Section 3.2).

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:

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

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 (van den Oord et al., 2019). In the standard CLIP formulation (Radford et al., 2021), 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 (Chen et al., 2020).

3.2 Synthetic Hard Negatives in the Representation Space

We augment the denominators of both \mathcal{L}_{i2t} and \mathcal{L}_{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:

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

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

The final loss combines both augmented directional terms:

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

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.* Section 5.5). With synthetic negatives, we produce high-quality, diverse negatives **fast** and **cheaply**, enabling effective training even with moderate batch sizes (*cf.* Section 5.6).

3.3 Synthetic Negative Generation Strategies

We define the “hardness” of negative samples by their similarity to the query (Kalantidis et al., 2020). We present the framework using the i2t direction; the t2i direction follows analogously. Given a query image embedding \mathbf{v}_i , we select the top- N batch text negatives with the highest cosine similarity to \mathbf{v}_i , denoted $\hat{\mathcal{T}}_N^i \subset \mathcal{B} \setminus \{i\}$. For the t2i direction, we select the top- N hard image negatives $\hat{\mathcal{V}}_N^i$ for a text query \mathbf{t}_i .

We implement six transformation strategies to generate synthetic negatives from these hard negatives. For the i2t direction, each strategy produces synthetic text negatives $\tilde{\mathbf{t}}_k$ using the query \mathbf{v}_i and hard negatives $\mathbf{t}_j, \mathbf{t}_l$ sampled uniformly from $\hat{\mathcal{T}}_M^i$:

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

All synthetic negatives are ℓ_2 -normalized after generation. The first three strategies draw from geometric transformations in the representation space: **(i) Interpolated negatives** ($s = 1$) create synthetic examples between the query and hard negatives via linear interpolation (Zhang et al., 2018), with ratio α_k . **(ii) Extrapolated negatives** ($s = 2$) explore directions beyond hard negatives, with extrapolation distance β_k . **(iii) Mixup negatives** ($s = 3$) combine pairs of hard negatives with mixing coefficient γ_k (Zhang et al., 2018; Verma et al., 2019). The fourth strategy introduces controlled stochasticity: **(iv) Noise-injected negatives** ($s = 4$) perturb hard negatives with Gaussian noise of variance σ^2 . The last two strategies are inspired by adversarial attacks (Goodfellow et al., 2015; Madry et al., 2018): **(v) Perturbed negatives** ($s = 5$) apply gradient-ascent perturbations with step size δ , analogous to PGD (Madry et al., 2018), and **(vi) Adversarial negatives** ($s = 6$) use sign-based perturbations with strength η , following FGSM (Goodfellow et al., 2015). For further intuition on each strategy, we refer readers to (Giakoumoglou and Stathaki, 2024; Giakoumoglou et al., 2025a). The complete set of synthetic text negatives for query i is $\mathcal{S}_{\text{txt}}^i = \bigcup_{s=1}^6 \{\tilde{\mathbf{t}}_k^{(s)}\}$, and analogously $\mathcal{S}_{\text{img}}^i = \bigcup_{s=1}^6 \{\tilde{\mathbf{v}}_k^{(s)}\}$ for the t2i direction. Crucially, strategies $s = 1$ 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 Section 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 (Section 3.3) naturally suggests applying the same strategies to CLIP-style training. However, vision-language models align two distinct modalities within a shared embedding space, and this dual-domain structure introduces failure modes that do not exist in the unimodal case. In this section, we systematically analyze these failure modes (Section 4.1), their secondary effect on the learnable temperature (Section 4.1.4), and present the resulting design choices

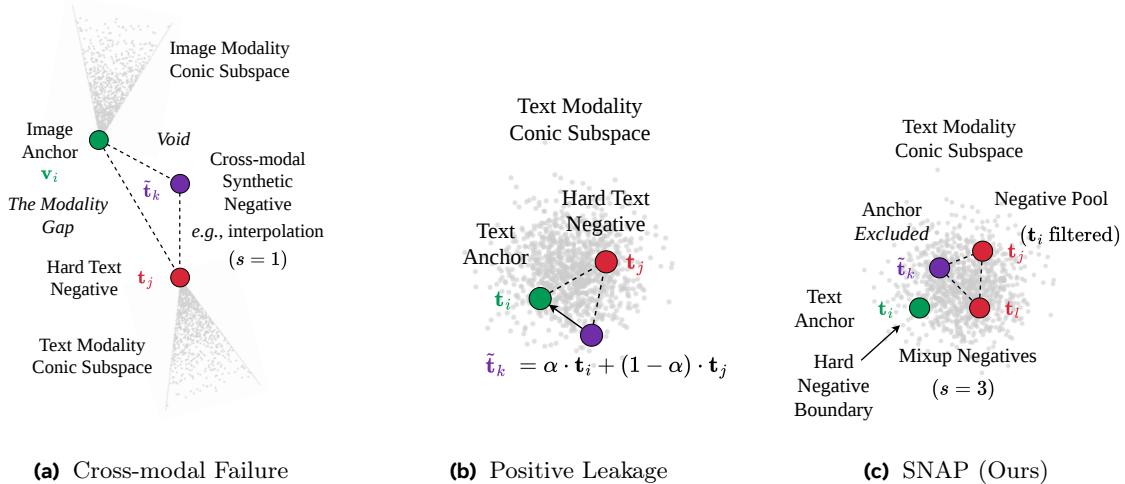


Figure 2 Geometric intuition of failure modes in vision-language pretraining. (a) **Cross-modal failure:** Blending image and text embeddings creates samples in the “modality gap” that are trivially rejected (Section 4.1). (b) **Positive Leakage:** Including the positive pair in synthesis creates negatives that contain the actual signal, causing gradient conflict (Section 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 (Section 4.2).

Table 2 Cross-modal vs. intra-modal strategies. 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	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 (↓)

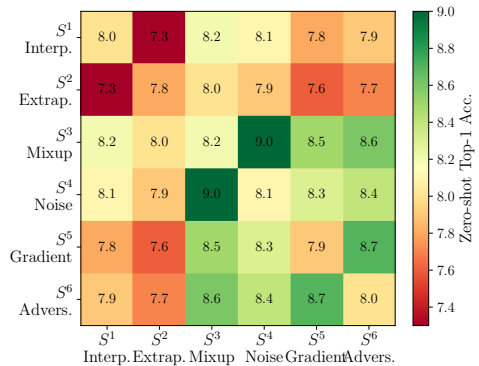


Figure 3 Pairwise strategy complementarity on IN-val. Strategies $s = 3$ and $s = 4$ achieve the highest accuracy.

that define SNAP (Section 4.2). All experiments are conducted on CC3M with ViT-B/16 and report ImageNet validation (IN-val) and ImageNet-V2 (IN-v2) zero-shot accuracy. Unless otherwise stated, we generate 64 synthetic negatives per query from the top-256 hardest negatives, split equally among the active strategies (e.g., 32 per strategy when using $s=3, 4$). The hyperparameters $\sigma = 0.01$, $\delta = 0.01$, and $\eta = 0.01$ are set empirically without extensive tuning, following (Giakoumoglou and Stathaki, 2024; Giakoumoglou et al., 2025a).

4.1 Naïve Extension and Failure Modes

We consider the natural ways to extend representation-space synthetic negatives to the vision-language setting and identify two distinct failure modes.

4.1.1 Cross-modal Synthetic Negatives Fail due to Geometry

The most direct extension from unimodal methods is to apply all six strategies from Equation (5) as defined, including strategies $s = 1$ and $s = 2$ that blend the image query \mathbf{v}_i with text negatives \mathbf{t}_j (or vice versa). However, vision-language models exhibit a well-documented *modality gap* (Liang et al., 2022) (Figure 2a):

Table 3 Temperature ablation. A fixed temperature prevents logit scale saturation caused by hard negatives, enabling SNAP to surpass CLIP by a large margin. We highlight the default hyperparameter.

Method	τ	IN-val	IN-v2
CLIP	learn.	10.4	8.4
SNAP	learn.	10.0 (↓)	9.0 (↑)
SNAP	fixed	11.0 (↑)	9.6 (↑)

Table 4 Directionality ablation. Applying synthetic negatives to both the i2t and t2i directions yields the best performance. We highlight the default hyperparameter.

Method	Dir.	IN-val	IN-v2
SNAP	i2t only	10.7	8.8
SNAP	t2i only	10.6	8.7
SNAP	both	11.0	9.6

image and text embeddings cluster in distinct regions (often described as separate cones) of the shared space. A cross-modal interpolation $\alpha \cdot \mathbf{v}_i + (1 - \alpha) \cdot \mathbf{t}_j$ produces a vector that falls between these cones—a region that real embeddings never occupy. Such synthetic negatives are trivially distinguishable from genuine same-modality embeddings and provide no useful contrastive signal. As shown in Table 2, applying all six strategies in cross-modal mode degrades performance below the CLIP baseline.

4.1.2 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 , keeping generation within a single modality. However, this introduces a subtler failure. Consider the resulting 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 genuine positive signal. As $\alpha \rightarrow 1$, it becomes nearly identical to the true positive, yet the InfoNCE denominator treats it as a negative and penalizes the model for being attracted to it. This *positive leakage* (Figure 2b) sends contradictory gradients: the numerator of Equation (2) encourages similarity with \mathbf{t}_i , while the denominator simultaneously penalizes similarity with $\tilde{\mathbf{t}} \approx \mathbf{t}_i$. Even moderate α values inject enough positive signal to confuse the optimization. Table 2 confirms that intra-modal generation with all six strategies improves over the cross-modal variant but still underperforms the CLIP baseline.

4.1.3 Two Distinct Failure Modes, One Solution

Cross-modal negatives fail because of *geometry*—the synthetic vectors land in no-man’s-land between the modality cones—while intra-modal negatives involving the positive fail because of *semantics*—genuine positive signal leaks into what should be a negative. Strategies $s = 3, 4$ avoid both problems entirely: they generate intra-modal negatives by combining or perturbing only non-matching negatives, never involving the positive from either modality. As shown in Table 2, restricting to $s = 3, 4$ in intra-modal mode recovers and matches the CLIP baseline. Figure 3 further validates this: across all pairwise strategy combinations, $s = 3$ and $s = 4$ consistently achieve the highest performance, while any combination involving cross-modal blending or the positive pair underperforms.

4.1.4 Temperature Divergence

Both failure modes described in Section 4.1 have a secondary destabilizing effect: they cause the learnable temperature τ to diverge. When synthetic negatives are either trivially easy (cross-modal, rejected without effort) or contradictory (positive-leaking, penalizing correct behavior), the gradient signal for τ becomes unreliable. The temperature responds by growing unboundedly, flattening the softmax distribution to reduce the impact of the problematic negatives. This creates a positive feedback loop: a larger τ makes *all* negatives—including real ones—less informative, which further degrades the contrastive signal and encourages τ to grow even further. Table 3 quantifies this effect. With a learnable temperature, SNAP achieves 10.0 on IN-val—matching the baseline but not exceeding it, because the temperature partially compensates for the harder negatives by inflating τ . Fixing the temperature removes this escape valve, forcing the model to actually learn from the harder contrastive signal: SNAP with a fixed temperature reaches 11.0 on IN-val and 9.6 on IN-v2, surpassing the CLIP baseline by a substantial margin. We adopt a fixed temperature as the default setting for all subsequent experiments.

4.2 SNAP

Guided by the analysis in [Sections 4.1](#) and [4.1.4](#), SNAP makes three principled design choices.

Intra-modal, positive-free strategies. SNAP uses only strategies $s = 3$ (mixup of two negatives) and $s = 4$ (noise injection on a single negative) from [Equation \(5\)](#), both operating in intra-modal mode ([Figure 2c](#)). These avoid cross-modal geometric collapse and intra-modal positive leakage ([Table 2](#) and [figure 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 ([Table 3](#)).

Bi-directional application. SNAP generates synthetic negatives for *both* the i2t and t2i directions. As shown in [Table 4](#), applying negatives to only one direction underperforms the bi-directional default, confirming that both modalities benefit from harder contrastive signal.

5 Experiments

We evaluate SNAP on top of two InfoNCE-based contrastive frameworks: CLIP ([Radford et al., 2021](#)) and FLIP ([Li et al., 2023](#)), which masks a large portion of image patches during training for efficiency. We reproduce CLIP ([Radford et al., 2021](#)) and SigLIP ([Zhai et al., 2023](#)) under the same training setup for a fair comparison and report results of LaCLIP ([Fan et al., 2023](#)), TripletCLIP ([Patel et al., 2024](#)), and NegCLIP ([Yuksekgonul et al., 2023](#)) from [Patel et al. \(2024\)](#).

5.1 Implementation Details

We build on the OpenCLIP ([Ilharco et al., 2021](#)) codebase and evaluate using CLIP Benchmark ([Cherti and Beaumont, 2022](#)).

Pretraining datasets. We pretrain on two image-text datasets of varying scale: CC3M ([Sharma et al., 2018](#)) and CC12M ([Changpinyo et al., 2021](#)). 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 ([Dosovitskiy et al., 2021](#)), and ResNet-50 ([He et al., 2016](#)). For the text encoder g_ϕ , we use a 12-layer Transformer ([Radford et al., 2021](#)). Both encoders include projection heads mapping to a shared $d = 512$ dimensional embedding space.

Training setup. We use AdamW ([Loshchilov and Hutter, 2019](#)) 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 $4 \times$ A100 GPUs for 40 epochs, with 224×224 images and text length 77. The temperature τ is initialized to 0.07 (and kept constant for SNAP).

5.2 Zero-Shot Image-Text Retrieval

We evaluate cross-modal retrieval on Flickr30k (Young et al., 2014) and MSCOCO (Lin et al., 2014). We compute pairwise cosine similarities between all image and text embeddings in the test split and report Recall@1, Recall@5, and Recall@10 for both i2t and t2i retrieval in Table 5. We observe that SNAP consistently improves retrieval performance across all datasets and backbones. These gains are most significant on MSCOCO, where the larger retrieval pool benefits from the improved discriminative power provided by our synthetic hard negatives.

Table 5 Zero-shot image-text retrieval on Flickr30k and MSCOCO. We report Recall@K (R@K) metrics for text and image retrieval. **Symbols:** † adapted from (Patel et al., 2024)

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 (Zhai et al., 2023)	7.2	19.8	26.5	3.1	10.0	15.7	5.3	15.2	22.3	2.8	9.0	14.0
	NegCLIP (Yuksekgonul et al., 2023)†	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 (Fan et al., 2023)†	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 (Patel et al., 2024)†	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 (Radford et al., 2021)	9.8	25.5	35.7	5.0	15.0	22.3	6.9	19.8	28.5	4.2	12.7	19.0
	+ SNAP (ours)	10.3	25.9	36.1	5.3	15.3	22.6	7.3	20.2	28.9	4.5	13.0	19.3
	FLIP (Li et al., 2023)	6.3	18.5	26.4	3.3	10.7	16.7	5.8	16.9	24.0	3.1	9.6	14.7
+ SNAP (ours)	6.8	19.0	26.9	3.7	11.1	17.1	6.3	17.4	24.5	3.5	10.0	15.1	
CC12M	SigLIP (Zhai et al., 2023)	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 (Yuksekgonul et al., 2023)†	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 (Fan et al., 2023)†	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 (Patel et al., 2024)†	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 (Radford et al., 2021)	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 (ours)	33.7	60.1	71.8	18.4	40.0	50.9	24.9	51.6	63.7	13.7	31.1	41.8
	FLIP (Li et al., 2023)	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 (ours)	26.8	50.7	62.4	15.0	32.7	44.1	20.9	44.8	56.8	11.3	26.8	37.0	
		ViT-B/16											
CC3M	SigLIP (Zhai et al., 2023)	13.6	31.5	43.6	7.0	19.8	28.1	9.3	24.5	33.4	5.6	15.8	23.1
	CLIP (Radford et al., 2021)	16.2	37.8	49.6	7.7	21.5	31.5	12.0	28.4	38.2	6.6	18.1	26.2
	+ SNAP (ours)	16.7	38.3	50.1	8.2	22.0	32.0	12.5	28.9	38.7	7.1	18.6	26.7
	FLIP (Li et al., 2023)	12.1	31.2	42.1	6.4	17.7	26.4	10.0	24.8	33.6	5.1	15.4	22.7
+ SNAP (ours)	12.6	31.7	42.6	6.9	18.2	26.9	10.5	25.3	34.1	5.6	15.9	23.2	
CC12M	SigLIP (Zhai et al., 2023)	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 (Radford et al., 2021)	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 (ours)	54.2	78.1	86.5	30.0	54.4	66.4	39.1	66.8	77.5	22.4	45.0	56.7
	FLIP (Li et al., 2023)	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 (ours)	48.5	73.8	82.8	26.1	49.7	61.8	34.2	61.1	72.7	19.4	40.0	51.5	
		RN-50											
CC3M	SigLIP (Zhai et al., 2023)	19.4	44.8	56.4	10.5	27.0	37.5	14.0	31.5	42.0	8.0	21.3	29.8
	CLIP (Radford et al., 2021)	20.8	45.8	56.8	11.7	29.6	41.4	16.1	36.5	47.1	9.6	24.3	34.0
	+ SNAP (ours)	21.3	46.3	57.3	12.2	30.1	41.9	16.6	37.0	47.6	10.1	24.8	34.5
	FLIP (Li et al., 2023)	20.6	44.2	56.5	11.3	29.4	40.4	15.8	36.0	47.0	8.8	23.6	33.1
+ SNAP (ours)	21.1	44.7	57.0	11.8	29.9	40.9	16.3	36.5	47.5	9.3	24.1	33.6	

5.3 Zero-Shot Classification

We evaluate zero-shot classification on ImageNet (Deng et al., 2009; Russakovsky et al., 2015) and 10 downstream datasets spanning fine-grained recognition (Stanford Cars (Krause et al., 2013), FGVC Aircraft (Maji et al., 2013)), scene recognition (SUN397 (Xiao et al., 2010)), texture classification (DTD (Cimpoi et al., 2014)), and generic object classification (CIFAR (Krizhevsky, 2009)). Class names are encoded as text prompts and cosine similarity is used for prediction without any task-specific training. Results are shown in Table 6. We observe that SNAP improves zero-shot accuracy across all settings. While general tasks like ImageNet show steady improvements, we observe the most substantial gains on fine-grained benchmarks.

Table 6 Zero-shot classification accuracy across diverse visual domains. We report top-1 accuracy on ImageNet and 10 downstream datasets, including fine-grained, scene, and object recognition benchmarks. **Symbols:** † adapted from (Patel et al., 2024)

Data	Model	Food-101	CIFAR-10	CIFAR-100	SUN397	Cars	Aircraft	DTD	Pets	Caltech-101	Flowers	Average	ImageNet
		ViT-B/32											
CC3M	SigLIP (Zhai et al., 2023)	3.8	23.8	8.1	12.4	0.8	0.9	4.4	4.1	26.6	4.2	8.9	5.6
	NegCLIP (Yuksekgonul et al., 2023)†	–	–	8.2	–	–	1.0	6.8	4.1	29.4	3.7	–	4.6
	LaCLIP (Fan et al., 2023)†	–	–	8.0	–	–	0.8	5.5	3.1	24.6	3.4	–	3.7
	TripletCLIP (Patel et al., 2024)	–	–	11.8	–	–	0.9	8.9	5.7	38.5	8.2	–	7.3
	CLIP (Radford et al., 2021)	5.9	28.7	10.0	14.5	0.8	1.9	8.2	7.0	41.5	5.9	12.4	8.0
	+ SNAP (ours)	6.2	28.8	10.2	14.8	2.0	3.1	8.2	6.9	41.8	7.1	12.9	8.2
	FLIP (Li et al., 2023)	4.7	26.8	8.7	11.4	0.6	1.1	4.5	5.3	37.1	3.9	10.4	6.8
+ SNAP (ours)	4.9	26.9	8.9	11.7	1.8	2.3	4.5	5.2	37.4	5.1	10.9	7.0	
CC12M	SigLIP (Zhai et al., 2023)	11.2	45.4	28.1	31.2	3.4	2.1	15.6	32.5	64.1	14.2	24.8	22.1
	NegCLIP (Yuksekgonul et al., 2023)†	–	–	26.0	–	–	1.5	14.2	30.8	63.4	13.5	–	20.2
	LaCLIP (Fan et al., 2023)†	–	–	23.6	–	–	1.5	13.0	34.7	62.9	13.4	–	19.7
	TripletCLIP (Patel et al., 2024)	–	–	30.3	–	–	1.2	17.4	33.2	65.3	15.4	–	23.3
	CLIP (Radford et al., 2021)	13.1	50.2	31.5	35.1	4.1	2.5	18.2	35.8	67.4	16.8	27.5	24.6
	+ SNAP (ours)	13.4	50.3	31.7	35.4	5.3	3.7	18.2	35.7	67.7	18.0	28.0	24.8
	FLIP (Li et al., 2023)	11.8	46.5	28.9	32.4	3.5	2.2	16.1	33.4	64.8	14.9	25.5	22.8
+ SNAP (ours)	12.1	46.6	29.1	32.7	4.7	3.4	16.0	33.4	65.1	16.1	25.9	23.0	
		ViT-B/16											
CC3M	SigLIP (Zhai et al., 2023)	6.2	28.9	7.0	18.3	1.1	1.2	8.7	6.0	35.5	7.0	12.0	9.9
	CLIP (Radford et al., 2021)	7.4	34.3	9.8	19.0	0.9	1.2	10.1	9.0	46.3	8.4	14.6	10.4
	+ SNAP (ours)	7.7	34.3	10.1	19.3	2.1	2.4	10.1	8.9	46.6	9.6	15.1	11.0
	FLIP (Li et al., 2023)	5.3	24.1	9.3	17.4	1.1	1.4	8.3	6.4	45.0	6.6	12.5	10.3
+ SNAP (ours)	5.5	24.2	9.5	17.7	2.3	2.6	8.3	6.3	45.3	7.8	13.0	10.5	
CC12M	SigLIP (Zhai et al., 2023)	19.8	58.4	36.2	41.5	6.2	4.1	22.1	51.2	78.4	24.1	34.2	31.2
	CLIP (Radford et al., 2021)	22.4	63.1	39.5	44.8	7.5	4.8	24.8	55.4	82.1	27.5	37.2	34.6
	+ SNAP (ours)	22.7	63.2	39.8	45.1	8.7	6.0	24.8	55.3	82.4	28.7	37.7	34.8
	FLIP (Li et al., 2023)	20.2	59.8	37.5	42.6	6.5	4.3	22.8	52.4	79.8	25.3	35.1	32.4
+ SNAP (ours)	20.5	59.9	37.7	42.9	7.7	5.5	22.8	52.3	80.1	26.5	35.6	32.6	
		RN-50											
CC3M	SigLIP (Zhai et al., 2023)	7.5	17.9	5.2	21.1	1.1	1.5	9.4	8.8	38.8	9.5	12.1	13.1
	CLIP (Radford et al., 2021)	7.7	23.7	4.8	23.3	1.4	1.3	9.3	11.7	45.4	12.3	14.1	13.6
	+ SNAP (ours)	8.0	23.8	5.0	23.6	2.6	2.5	9.3	11.6	45.7	13.5	14.6	13.8
	FLIP (Li et al., 2023)	9.0	23.1	6.7	22.4	0.8	1.1	10.9	9.0	41.8	11.7	13.7	13.7
	+ SNAP (ours)	9.2	23.1	6.9	22.7	2.0	2.3	10.9	8.9	42.1	12.9	14.1	13.9

5.4 Linear Probe Classification

We evaluate the quality of frozen image representations by training a linear classifier on top of the frozen image encoder. Following standard protocols (Radford et al., 2021; Kornblith et al., 2019), we freeze all encoder parameters and train only a linear layer using the training split of each dataset, reporting top-1 accuracy on the test split. Results are shown in Table 7. We observe that SNAP-trained encoders produce more linearly separable and robust visual features. The overall average accuracy increases across diverse backbones and data scales, confirming that the benefits of synthetic negatives transfer directly to the visual encoder.

Table 7 Linear probe evaluation of frozen visual representations. We report top-1 accuracy for a linear classifier trained on fixed image encoder features across ImageNet and 10 specialized transfer datasets.

Data	Model	Food-101	CIFAR-10	CIFAR-100	SUN397	Cars	Aircraft	DTD	Pets	Caltech-101	Flowers	Average	ImageNet
		ViT-B/32											
CC3M	SigLIP (Zhai et al., 2023)	35.4	72.5	48.4	29.4	9.5	13.6	31.9	34.0	61.6	50.8	38.7	30.2
	CLIP (Radford et al., 2021)	40.7	77.2	54.6	32.7	13.2	20.1	38.3	40.7	71.1	61.5	45.0	33.9
	+ SNAP (ours)	40.9	77.2	54.8	32.9	14.4	21.3	38.3	40.6	71.3	62.7	45.4	34.1
	FLIP (Li et al., 2023)	40.8	74.8	52.8	34.7	13.9	17.9	39.8	37.6	69.7	61.1	44.3	32.9
	+ SNAP (ours)	41.0	74.8	53.0	34.9	15.1	19.1	39.7	37.6	69.9	62.3	44.7	33.1
CC12M	SigLIP (Zhai et al., 2023)	54.2	84.6	63.8	44.1	24.8	31.2	48.9	56.4	79.2	78.4	56.6	45.8
	CLIP (Radford et al., 2021)	58.6	88.5	68.4	47.9	28.5	36.8	54.2	62.1	85.1	85.9	61.6	48.7
	+ SNAP (ours)	58.8	88.6	68.6	48.1	29.7	38.0	54.1	62.1	85.3	87.1	62.0	48.9
	FLIP (Li et al., 2023)	57.5	87.2	67.1	46.8	27.4	35.1	53.0	60.9	83.9	84.1	60.3	47.5
	+ SNAP (ours)	57.7	87.1	67.3	47.0	28.6	36.3	53.0	60.8	84.1	85.3	60.7	47.7
		ViT-B/16											
CC3M	SigLIP (Zhai et al., 2023)	46.3	73.7	50.3	37.6	14.7	21.7	45.0	47.2	70.8	66.8	47.4	41.3
	CLIP (Radford et al., 2021)	49.2	77.1	55.2	40.3	18.1	24.6	47.8	51.2	73.7	71.6	50.9	43.1
	+ SNAP (ours)	49.4	77.1	55.4	40.5	19.3	25.8	47.7	51.2	73.9	72.8	51.3	43.3
	FLIP (Li et al., 2023)	48.6	75.0	52.9	41.0	16.0	21.3	46.2	47.4	73.2	70.6	49.2	40.9
	+ SNAP (ours)	48.8	75.0	53.1	41.2	17.2	22.5	46.1	47.4	73.4	71.8	49.6	41.1
CC12M	SigLIP (Zhai et al., 2023)	64.1	85.8	66.2	51.4	32.5	38.7	58.1	65.8	84.2	86.4	63.3	55.4
	CLIP (Radford et al., 2021)	68.4	89.2	70.5	55.1	36.8	43.1	62.4	70.5	88.5	90.1	67.5	58.7
	+ SNAP (ours)	68.6	89.2	70.7	55.3	38.0	44.3	62.4	70.4	88.7	91.3	67.9	58.9
	FLIP (Li et al., 2023)	67.1	88.0	69.3	54.0	35.5	41.8	61.2	69.3	87.2	88.9	66.2	57.5
	+ SNAP (ours)	67.3	87.9	69.5	54.2	36.7	43.0	61.2	69.2	87.4	90.1	66.6	57.7
		RN-50											
CC3M	SigLIP (Zhai et al., 2023)	50.9	70.0	46.0	45.6	17.0	20.9	49.5	51.3	73.5	71.5	49.6	47.2
	CLIP (Radford et al., 2021)	56.1	74.0	52.3	50.0	24.6	31.1	53.0	62.4	80.4	80.4	56.4	50.0
	+ SNAP (ours)	56.3	74.0	52.5	50.2	25.8	32.3	52.9	62.3	80.6	81.6	56.8	50.2
	FLIP (Li et al., 2023)	56.7	75.5	53.6	49.0	24.6	31.0	54.4	62.2	80.1	78.8	56.6	50.0
	+ SNAP (ours)	56.9	75.4	53.8	49.2	25.8	32.2	54.4	62.1	80.3	80.0	57.0	50.2

5.5 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 (Radford et al., 2021) requires 32.5 minutes per epoch, while SNAP adds only 3 minutes, resulting in an 8.92% overhead. Similarly, FLIP (Li et al., 2023) 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 fine-grained discriminative gains with a negligible increase in total training time.

5.6 Further Ablations

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

Table 8 Ablation on batch size $|\mathcal{B}|$ for SNAP (strategies $s = 3, 4$; $|\mathcal{S}| = 128$).

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 9 Ablation on number of synthetic negatives $|\mathcal{S}| = |\mathcal{S}_{\text{img}}| + |\mathcal{S}_{\text{txt}}|$ for SNAP (strategies $s = 3, 4$; $|\mathcal{B}| = 4096$).

$ \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

Batch size. We evaluate the sensitivity of SNAP to different batch sizes within our computational budget: 1024, 2048, and 4096. Unlike standard CLIP, which relies heavily on extremely large batches to provide sufficient negative diversity, SNAP maintains more stable performance across these scales (*cf.* Table 8).

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 Table 9, 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.

6 Discussion & Limitations

Discussion. These failure modes are structural consequences of aligning two modalities via contrastive learning, rather than unique to CLIP. Any vision-language method introducing synthetic perturbations will face similar modality gaps and positive leakage. We expect the design principles behind SNAP—restricting synthesis to intra-modal, positive-free operations and fixing the temperature—to transfer to other dual-encoder frameworks and multi-modal settings. More broadly, our findings suggest that unimodal contrastive techniques should not be adopted as black boxes; the shared space geometry imposes constraints requiring deliberate adaptation.

Limitations. Our evaluation is limited to CC3M and CC12M due to computational constraints; validating SNAP at larger pretraining scales (*e.g.*, 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

- Romain Beaumont. img2dataset: Easily turn large sets of image urls to an image dataset. <https://github.com/rom1504/img2dataset>, 2021.
- Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – Mining Discriminative Components with Random Forests. In *Eur. Conf. Comput. Vis. (ECCV)*, pages 446–461, 2014.
- Anjia Cao, Xing Wei, and Zhiheng Ma. FLAME: Frozen Large Language Models Enable Data-Efficient Language-Image Pre-training. In *CVPR*, 2025.
- Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12M: Pushing Web-Scale Image-Text Pre-Training To Recognize Long-Tail Visual Concepts. In *CVPR*, pages 3558–3568, June 2021.
- Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, pages 1597–1607. PMLR, 2020.
- Mehdi Cherti and Romain Beaumont. CLIP benchmark, November 2022. <https://doi.org/10.5281/zenodo.15403103>.
- Mircea Cimpoi, Subhansu Maji, Iasonas Kokkinos, and Andrea Vedaldi. Describing Textures in the Wild. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 3606–3613, 2014.
- Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 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>.
- Lijie Fan, Dilip Krishnan, Phillip Isola, Dina Katabi, and Yonglong Tian. Improving CLIP Training with Language Rewrites. In *NeurIPS*, 2023.
- Li Fei-Fei, Rob Fergus, and Pietro Perona. 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.
- Nikolaos Giakoumoglou and Tania Stathaki. SynCo: Synthetic Hard Negatives for Contrastive Visual Representation Learning, 2024. <https://arxiv.org/abs/2410.02401>.
- Nikolaos Giakoumoglou, Andreas Floros, Kleanthis Marios Papadopoulos, and Tania Stathaki. SynCo-v2: An Empirical Study of Training Self-Supervised Vision Transformers with Synthetic Hard Negatives, 2025a. <https://giakoumoglou.com/src/syncov2/syncov2-main.pdf>.
- Nikolaos Giakoumoglou, Andreas Floros, Kleanthis Marios Papadopoulos, and Tania Stathaki. Unsupervised Training of Vision Transformers with Synthetic Negatives. In *Proceedings of the CVPR 2025 Workshop on Visual Concepts*, 2025b. <https://openreview.net/forum?id=dg8FuaOKnC>.
- Nikolaos Giakoumoglou, Tania Stathaki, and Athanasios Gkelias. A Review on Discriminative Self-supervised Learning Methods in Computer Vision, 2025c. <https://arxiv.org/abs/2405.04969>.
- Shashank Goel, Hritik Bansal, Sumit Bhatia, Ryan A. Rossi, Vishwa Vinay, and Aditya Grover. Cyclip: Cyclic contrastive language-image pretraining, 2022. <https://arxiv.org/abs/2205.14459>.
- Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples, 2015. <https://arxiv.org/abs/1412.6572>.
- Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *CVRP*, volume 2, pages 1735–1742. IEEE, 2006.

- Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *CVPR*, pages 770–778, 2016. <https://api.semanticscholar.org/CorpusID:206594692>.
- Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. OpenCLIP, July 2021. <https://doi.org/10.5281/zenodo.5143773>. If you use this software, please cite it as below.
- 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. <https://arxiv.org/abs/2102.05918>.
- Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. Hard Negative Mixing for Contrastive Learning. In *NeurIPS*, 2020.
- Simon Kornblith, Jonathon Shlens, and Quoc V. Le. Do Better ImageNet Models Transfer Better?, 2019. <https://arxiv.org/abs/1805.08974>.
- Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3D Object Representations for Fine-Grained Categorization. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 554–561, 2013.
- Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. Technical Report 0, University of Toronto, 2009.
- Zhengfeng Lai, Haotian Zhang, Bowen Zhang, Wentao Wu, Haoping Bai, Aleksei Timofeev, Xianzhi Du, Zhe Gan, Jiulong Shan, Chen-Nee Chuah, Yinfei Yang, and Meng Cao. VeCLIP: Improving CLIP Training via Visual-enriched Captions, 2024.
- Samuel Lavoie, Polina Kirichenko, Mark Ibrahim, Mido Assran, Andrew Gordon Wilson, Aaron Courville, and Nicolas Ballas. Modeling caption diversity in contrastive vision-language pretraining. In *ICML*, 2024. <https://openreview.net/forum?id=iaV2fU6Dif>.
- Yanguang Li, Feng Liang, Lichen Zhao, Yufeng Cui, Wanli Ouyang, Jing Shao, Fengwei Yu, and Junjie Yan. Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. In *ICLR*, 2022. <https://openreview.net/forum?id=zq1iJkNk3uN>.
- Yanghao Li, Haoqi Fan, Ronghang Hu, Christoph Feichtenhofer, and Kaiming He. Scaling language-image pre-training via masking, 2023. <https://arxiv.org/abs/2212.00794>.
- Weixin Liang, Yuhui Zhang, Yongchan Kwon, Serena Yeung, and James Zou. Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning, 2022. <https://arxiv.org/abs/2203.02053>.
- Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. *CoRR*, abs/1405.0312, 2014. <http://dblp.uni-trier.de/db/journals/corr/corr1405.html/LinMBHPRDZ14>.
- Yanqing Liu, Xianhang Li, Zeyu Wang, Bingchen Zhao, and Cihang Xie. CLIPS: An Enhanced CLIP Framework for Learning with Synthetic Captions. *arXiv preprint arXiv:2411.16828*, 2024a.
- Yanqing Liu, Kai Wang, Wenqi Shao, Ping Luo, Yu Qiao, Mike Zheng Shou, Kaipeng Zhang, and Yang You. MLLMs-Augmented Visual-Language Representation Learning, 2024b. <https://arxiv.org/abs/2311.18765>.
- Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *ICLR*. OpenReview.net, 2019. <https://openreview.net/forum?id=Bkg6RiCqY7>.
- Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks. In *ICLR*, 2018. <https://openreview.net/forum?id=rJzIBfZAb>.
- Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew B. Blaschko, and Andrea Vedaldi. Fine-Grained Visual Classification of Aircraft. *arXiv*, abs/1306.5151, 2013.
- Kevis-Kokitsi Maninis, Kaifeng Chen, Soham Ghosh, Arjun Karpur, Koert Chen, Ye Xia, Bingyi Cao, Daniel Salz, Guangxing Han, Jan Dlabal, Dan Gnanapragasam, Mojtaba Seyedhosseini, Howard Zhou, and André Araujo. TIPS: Text-Image Pretraining with Spatial Awareness. In *ICLR*, 2025.
- Norman Mu, Alexander Kirillov, David Wagner, and Saining Xie. Slip: Self-supervision meets language-image pre-training. In *European Conference on Computer Vision*, pages 523–540. Springer, 2022.
- Muhammad Ferjad Naeem, Yongqin Xian, Xiaohua Zhai, Lukas Hoyer, Luc Van Gool, and Federico Tombari. Silc: Improving vision language pretraining with self-distillation, 2023. <https://arxiv.org/abs/2310.13355>.

- Maria-Elena Nilsback and Andrew Zisserman. Automated Flower Classification by Use of Color, Shape, and Texture Features. *Machine Vision and Applications*, 18(1):27–39, 2008.
- Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and Dogs. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 3498–3505, 2012.
- 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. *NeurIPS*, 2024.
- Hieu Pham, Zihang Dai, Golnaz Ghiasi, Kenji Kawaguchi, Hanxiao Liu, Adams Wei Yu, Jiahui Yu, Yi-Ting Chen, Minh-Thang Luong, Yonghui Wu, Mingxing Tan, and Quoc V. Le. Combined scaling for zero-shot transfer learning. *Neurocomputing*, 555:126658, 2023. ISSN 0925-2312. doi: <https://doi.org/10.1016/j.neucom.2023.126658>. <https://www.sciencedirect.com/science/article/pii/S0925231223007816>.
- Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis, 2023. <https://arxiv.org/abs/2307.01952>.
- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. Technical report, OpenAI, 2018. https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language Models are Unsupervised Multitask Learners. Technical report, OpenAI, 2019. https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.
- 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. In Marina Meila and Tong Zhang, editors, *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021. <https://proceedings.mlr.press/v139/radford21a.html>.
- Kanchana Ranasinghe, Brandon McKinzie, Sachin Ravi, Yinfei Yang, Alexander Toshev, and Jonathon Shlens. Perceptual grouping in contrastive vision-language models, 2023. <https://arxiv.org/abs/2210.09996>.
- Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. *arXiv preprint arXiv:2010.04592*, 2020.
- Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, April 2015. ISSN 1573-1405. doi: 10.1007/s11263-015-0816-y. <http://dx.doi.org/10.1007/s11263-015-0816-y>.
- Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, page 815–823. IEEE, June 2015. doi: 10.1109/cvpr.2015.7298682. <http://dx.doi.org/10.1109/CVPR.2015.7298682>.
- 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.
- Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. FLAVA: A foundational language and vision alignment model. In *CVPR*, 2022.
- Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. https://proceedings.neurips.cc/paper_files/paper/2016/file/6b180037abbebea991d8b1232f8a8ca9-Paper.pdf.
- Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. Eva-clip: Improved training techniques for clip at scale, 2023. <https://arxiv.org/abs/2303.15389>.
- Quan Sun, Jinsheng Wang, Qiying Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, and Xinlong Wang. Eva-clip-18b: Scaling clip to 18 billion parameters, 2024. <https://arxiv.org/abs/2402.04252>.
- Michael Tschanen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyers, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas

- Steiner, and Xiaohua Zhai. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features, 2025. <https://arxiv.org/abs/2502.14786>.
- Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding, 2019. <https://arxiv.org/abs/1807.03748>.
- Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, Aaron Courville, David Lopez-Paz, and Yoshua Bengio. Manifold Mixup: Better Representations by Interpolating Hidden States, 2019. <https://arxiv.org/abs/1806.05236>.
- Chao-Yuan Wu, R. Manmatha, Alexander J. Smola, and Philipp Krähenbühl. Sampling matters in deep embedding learning, 2018. <https://arxiv.org/abs/1706.07567>.
- Wei Wu, Kecheng Zheng, Shuailei Ma, Fan Lu, Yuxin Guo, Yifei Zhang, Wei Chen, Qingpei Guo, Yujun Shen, and Zheng-Jun Zha. LoTLIP: Improving Language-Image Pre-training for Long Text Understanding. In *NeurIPS*, 2024. <https://openreview.net/forum?id=pc4GSBi1Hx>.
- Bin Xiao, Haiping Wu, Weijian Xu, Xiyang Dai, Houdong Hu, Yumao Lu, Michael Zeng, Ce Liu, and Lu Yuan. Florence-2: Advancing a unified representation for a variety of vision tasks. *arXiv preprint arXiv:2311.06242*, 2023.
- Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. SUN Database: Large-scale Scene Recognition from Abbey to Zoo. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pages 3485–3492, 2010.
- Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Bin Xiao, Ce Liu, Lu Yuan, and Jianfeng Gao. Unified contrastive learning in image-text-label space, 2022.
- Kaicheng Yang, Jiankang Deng, Xiang An, Jiawei Li, Ziyong Feng, Jia Guo, Jing Yang, and Tongliang Liu. ALIP: Adaptive Language-Image Pre-training with Synthetic Caption, 2023.
- Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. Filip: Fine-grained interactive language-image pre-training, 2021. <https://arxiv.org/abs/2111.07783>.
- 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.
- Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. CoCa: Contrastive Captioners are Image-Text Foundation Models. *TMLR*, 2022. ISSN 2835-8856. <https://openreview.net/forum?id=Ee277P3AYC>.
- Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, Ce Liu, Mengchen Liu, Zicheng Liu, Yumao Lu, Yu Shi, Lijuan Wang, Jianfeng Wang, Bin Xiao, Zhen Xiao, Jianwei Yang, Michael Zeng, Luwei Zhou, and Pengchuan Zhang. Florence: A new foundation model for computer vision, 2021. <https://arxiv.org/abs/2111.11432>.
- 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? In *ICLR*, 2023. <https://openreview.net/forum?id=KRLUvvh8uaX>.
- Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. LiT: Zero-Shot Transfer with Locked-image text Tuning. In *CVPR*, pages 18123–18133, June 2022.
- Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid Loss for Language Image Pre-Training, 2023. <https://arxiv.org/abs/2303.15343>.
- Beichen Zhang, Pan Zhang, Xiaoyi Dong, Yuhang Zang, and Jiaqi Wang. Long-clip: Unlocking the long-text capability of clip. *arXiv preprint arXiv:2403.15378*, 2024.
- Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond Empirical Risk Minimization. In *ICLR*, 2018. <https://openreview.net/forum?id=r1Ddp1-Rb>.
- Le Zhang, Rabiul Awal, and Aishwarya Agrawal. Contrasting Intra-Modal and Ranking Cross-Modal Hard Negatives to Enhance Visio-Linguistic Fine-grained Understanding. *arXiv preprint arXiv:2306.08832*, 2023.
- Kecheng Zheng, Yifei Zhang, Wei Wu, Fan Lu, Shuailei Ma, Xin Jin, Wei Chen, and Yujun Shen. DreamLIP: Language-Image Pre-training with Long Captions. In *ECCV*, 2024.

Yufeng Zhou, Lu Qi, Jinyuan Wang, Ming-Hsuan Feng, Wei Zhang, Zhiyong Yang, Weilin Liu, and Yizhou Wang. Non-contrastive learning meets language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8913–8923, June 2023.

Appendix

A Dataset Details

A.1 Pretraining Datasets

We pretrain on two image-text datasets of varying scale: CC3M (Sharma et al., 2018) and CC12M (Changpinyo et al., 2021). Both datasets were downloaded using `img2dataset`¹ (Beaumont, 2021). 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.* Section 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 (Young et al., 2014) and MSCOCO (Lin et al., 2014), and classification on ImageNet (Deng et al., 2009) and 10 downstream datasets. Details are provided in Table 10.

Table 10 Details of downstream classification datasets.

Dataset	Metric	Categories	Train Size	Test Size
Food-101 (Bossard et al., 2014)	Accuracy	101	75,750	25,250
CIFAR-10 (Krizhevsky, 2009)	Accuracy	10	50,000	10,000
CIFAR-100 (Krizhevsky, 2009)	Accuracy	100	50,000	10,000
SUN397 (Xiao et al., 2010)	Accuracy	397	19,850	19,850
Stanford Cars (Krause et al., 2013)	Accuracy	196	8,144	8,041
FGVC Aircraft (Maji et al., 2013)	Mean per class	100	6,667	3,333
DTD (Cimpoi et al., 2014)	Accuracy	47	3,760	1,880
Oxford Pets (Parkhi et al., 2012)	Mean per class	37	3,680	3,669
Caltech-101 (Fei-Fei et al., 2007)	Mean per class	102	3,060	6,085
Oxford Flowers (Nilsback and Zisserman, 2008)	Mean per class	102	2,040	6,149
ImageNet (Deng et al., 2009)	Accuracy	1,000	1,281,167	50,000
Flickr30k (Young et al., 2014)	Recall@ <i>k</i>	–	29,000	1,000
MSCOCO (Lin et al., 2014)	Recall@ <i>k</i>	–	113,287	5,000

B Implementation Details

We build on the OpenCLIP² (Ilharco et al., 2021) codebase and evaluate using CLIP Benchmark³ (Cherti and Beaumont, 2022).

B.1 Encoders

We use ViT-B/16, ViT-B/32 (Dosovitskiy et al., 2021), 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

¹<https://github.com/rom1504/img2dataset>

²https://github.com/mlfoundations/open_clip

³https://github.com/LAION-AI/CLIP_benchmark

CLIP (Radford et al., 2021), 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 Tables 11a and 11b.

Table 11 Hyperparameters. (a) 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) Training 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	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		

C Related Work

We briefly contrast SNAP with the most closely related vision-language methods that incorporate synthetic data during pretraining. NegCLIP (Yuksekgonul et al., 2023) 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 (Fan et al., 2023) augments the text side with synthetic positive captions generated by large language models such as ChatGPT, Bard, and LLaMA. DreamLIP (Zheng et al., 2024) follows a similar philosophy but leverages multimodal LLMs (ShareGPT4V, LLaVA, InstructBLIP) to produce richer positive captions grounded in visual content. TripletCLIP (Patel et al., 2024) 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 9% training time overhead.

D 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.

E 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. Nevertheless, we demonstrated consistent improvements across all evaluated settings, and expect the benefits of synthetic hard negatives to persist or increase at larger scale where batch diversity remains limited relative to dataset size.

F 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 (Radford et al., 2021) and FLIP (Li et al., 2023) frameworks evaluated here, including methods such as SLIP (Mu et al., 2022), DeCLIP (Li et al., 2022), LiT (Zhai et al., 2022), LaCLIP (Fan et al., 2023), DreamLIP (Zheng et al., 2024), CyCLIP (Goel et al., 2022), and TripletCLIP (Patel et al., 2024), all of which build on the same InfoNCE-style contrastive objective.

G Reproducibility

We build on the OpenCLIP (Ilharco et al., 2021) codebase for all training and use CLIP Benchmark (Cherti and Beaumont, 2022) for all downstream evaluations. Pretraining datasets were downloaded using img2dataset (Beaumont, 2021). All experiments use identical hyperparameters across architectures and datasets (*cf.* Table 11). Training was performed on $4\times$ NVIDIA A100 80GB GPUs using PyTorch with mixed-precision (FP16) training. We use the same 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.

H 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., 512)
# s3, s4: number of synthetic negatives per strategy (default: 32 each)

for images, texts in loader:
    # extract embeddings
    v_feat = f_v(images) # Nx C
    t_feat = f_t(texts) # Nx C

    # compute standard CLIP logits (image-to-text)
    logits_base = mm(v_feat, t_feat.t()) # Nx N

    # identify hard negative indices from the current batch
    # excluding the diagonal (positives)
    mask = eye(N).bool()
    logits_neg = logits_base.masked_fill(mask, -1e9)
    _, idxs_hard = topk(logits_neg, k=n_hard, dim=1)

    # S3: Mixup Hard Negatives (between two hard negative anchors)
    s3_feats = generate_s3(t_feat, idxs_hard, s3) # N x s3 x C
    l_s3 = einsum("nc,nsc->ns", [v_feat, s3_feats])

    # S4: Noise-Injected Hard Negatives (perturbation of a hard negative anchor)
    s4_feats = generate_s4(t_feat, idxs_hard, s4) # N x s4 x C
    l_s4 = einsum("nc,nsc->ns", [v_feat, s4_feats])

    # total logits: Nx(N + s3 + s4)
    logits = cat([logits_base, l_s3, l_s4], dim=1)

    # contrastive loss (positives are the diagonal indices 0..N-1)
    labels = arange(N)
    loss = CrossEntropyLoss(logits / t, labels)

    loss.backward()
    update_optimizers()
```

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)
```
