Tune-Up Open-Weight CLIP: Optimization Framework for Self-Supervised Fine-tuning of CLIP

Anant Mehta **Xingyu Chen** **Xiyuan Wei** **Tianbao Yang***
Department of Computer Science & Engineering
Texas A&M University, TX, USA
{anant_mehta,tamucxy2023,xwei,tianbao-yang}@tamu.edu

Abstract

CLIP has become a cornerstone of multimodal representation learning, yet improving its performance typically requires a prohibitively costly process of training from scratch on billions of samples. We ask a different question: *Can we improve the performance of open-weight CLIP models across various downstream tasks using only existing self-supervised datasets?* Unlike supervised fine-tuning, which adapts a pretrained model to a single downstream task, our setting seeks to improve general performance across various tasks. However, as both our experiments and prior studies reveal, simply applying standard training protocols starting from an open-weight CLIP model often fails, leading to performance degradation. In this paper, we introduce **TuneCLIP**, a self-supervised fine-tuning framework that overcomes the performance degradation. TuneCLIP has two key components: (1) a warm-up stage of recovering optimization statistics to reduce cold-start bias, inspired by theoretical analysis, and (2) a fine-tuning stage of optimizing a new contrastive loss to mitigate the penalization on false negative pairs. Our extensive experiments show that TuneCLIP consistently improves performance across model architectures and scales. Notably, it elevates leading open-weight models like SigLIP (ViT-B/16), achieving gains of up to +2.5% on ImageNet and related out-of-distribution benchmarks, and +1.2% on the highly competitive DataComp benchmark, setting a new strong baseline for efficient post-pretraining adaptation. The code is available at `https://github.com/Optimization-AI/TuneCLIP`.

1 Introduction

Contrastive vision-language models such as CLIP which learn joint image-text representations at scale by training on hundreds of millions of large scale image-text pairs [1, 2] show broad utility across downstream tasks spanning classification, cross-modal retrieval, multimodal reasoning [3, 4] and generation [5]. Recent efforts to improve CLIP have primarily focused on pretraining by constructing ever larger datasets [6], designing novel objective functions [7, 8], or developing refined optimization algorithms [8, 9]. While these directions have advanced the state of the art, they come at staggering cost due to billions of image-text pairs, massive GPU clusters, and days or weeks of computation. In this work, we ask a complementary but equally important question "*How can we unlock more from the CLIP we already have?*", shifting from "*How can we pretrain a better CLIP from scratch?*", which leads to a path that is cheaper, faster, and far more compute-efficient.

A very common way to improve the model is supervised fine-tuning, which is performed on specific datasets of a target domain [10, 11, 12]. These prior works leverage class labels or captions to steer the embedding space. An issue with these methods is that strong adaptation to the target domain can harm generalization contributing to reduced robustness to distribution shifts [13, 14]. Thus,

*Corresponding author.

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

supervised fine-tuning cannot be regarded as a procedure to improve a CLIP model in general, rather, it is a *domain adaptation* for a specific distribution, often at the expense of transferability.

These limitations motivate an alternative paradigm that we study, namely *self-supervised fine-tuning* (SSFT), which is the process of improving a pretrained CLIP model's overall representational quality and general-purpose performance, rather than tailoring it to a specific downstream task. What makes SSFT actually "self-supervised"? Traditionally, supervised fine-tuning is carried out on datasets such as ImageNet, CIFAR, or Flickr [15, 16, 17, 18, 19, 20, 12, 21], which were constructed through human annotation or domain-specific filtering and are thereby inherently biased models toward a particular domain. By contrast, we use large-scale web corpora that were constructed for pretraining CLIP models [2], e.g., [6]. The result is task-agnostic corpora, positioning SSFT on them as representation refinement rather than task adaptation.

A key challenge of applying SSFT to a well-pretrained CLIP model is a sharp performance degradation in the first epoch, as observed in our experiments (cf. Figure 1) and corroborated by prior work [23]. This initial drop is difficult to recover from, even when training is extended for additional epochs. Although this phenomenon may appear related to catastrophic forgetting in continual learning [24, 14, 25, 26], the underlying cause is fundamentally different. Catastrophic forgetting stems from distributional shift introduced by new task data; however, we observe the same performance drop even when training uses the same data as CLIP pretraining (cf. Figure 8 in Appendix F). Our first contribution is a theoretical analysis of this failure mode from an optimization perspective. We establish that the optimization error in contrastive learning is highly sensitive to the error of gradient estimates at initialization, and that this sensitivity is especially pronounced when starting from a well-pretrained model. The root cause lies in a known property of contrastive losses — they do not admit unbiased stochastic gradient estimators [27] — which renders the standard initialization of gradient estimator particularly unreliable and makes early training dynamics difficult to control. This manifests as a sharp and hard-to-recover performance degradation at the onset of fine-tuning, which we term **cold-start bias**.

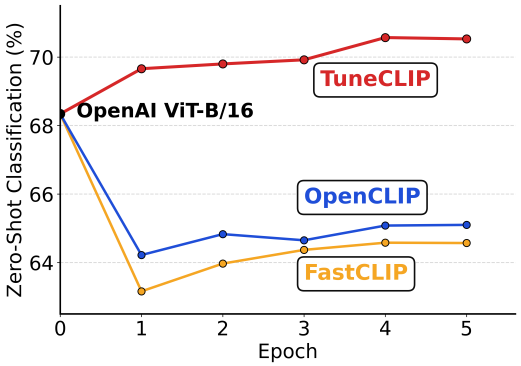


Figure 1: Early-stage performance drop on ImageNet when fine-tuning OpenAI ViT-B/16 model on DFN-12M dataset using OpenCLIP and FastCLIP pipelines.

In addition to the optimization challenge of cold-start bias, SSFT exposes a distinct learning-level issue caused by **false negatives**, whereby data semantically similar to the anchor are mistakenly treated as negatives during training. This problem is exacerbated as the model becomes better at capturing semantic structure, since fine-tuning further amplifies the similarity gap between positive and negative pairs, causing false negatives to increasingly distort the learned embeddings. The downstream effect is a degradation in retrieval performance, which depends critically on fine-grained ranking among semantically similar examples. Although prior work has addressed false negatives in contrastive learning [28, 29, 30, 31], existing methods are limited to the mini-batch level or require non-trivial additional overhead to detect false negatives globally at training time. We sidestep these limitations with a simple yet effective remedy: a hinge-based surrogate loss that integrates naturally into the global contrastive learning framework without any additional cost.

Our contributions are summarized below:

- We provide a theoretical analysis quantifying the cold-start bias, showing how the initial gradient estimation error in contrastive loss optimization influences convergence. To mitigate this issue, we propose *Optimizer Statistics Recovery (OSR)*, which restores accurate first and second-order moment estimates, along with other useful statistics of the initial model, through a warm-up stage.
- To reduce the impact of false negatives, we introduce a simple yet effective remedy: the *hinged global contrastive loss (HGCL)*. This loss penalizes positive and negative pairs only when their

[2]While some studies treat CLIP as weakly supervised due to its web-crawled image-text training data, we follow [22] and classify it as self-supervised.

similarity gap exceeds a margin, thereby avoiding excessive penalization of false negatives. This improves retrieval performance while preserving strong zero-shot classification accuracy.

- We conduct extensive experiments on SSFT across multiple pretrained models and data scales. Our results show consistent improvements over base models and demonstrate superiority to existing standard pretraining approaches that can be used for SSFT.

2 Related Work

Improving CLIP Pretraining. Extensive works improve CLIP pretraining via better objective functions and training recipes. A line of work builds upon the standard mini-batch contrastive loss [1, 2]. For instance, [32] incorporates image masking within the contrastive framework, [33] enforces geometrical consistency in the learned representation of images and texts, while [34] improves CLIP training with language rewrites. [6] has proposed a data filtering network to better filter low-quality data for improving CLIP training. In addition, system-level advances in distributed training and memory efficiency have enabled scalable CLIP pretraining at web scale [35, 36, 2, 37].

However, these approaches typically rely on extremely large batch sizes, leading to significant memory and system constraints. To address this limitation, a parallel line of work replaces mini-batch contrastive loss with a global contrastive loss. Notably, [27] introduce SogCLR to directly optimize a global contrastive objective based on compositional optimization, achieving superior performance without requiring massive batches. Subsequent works [7, 8] further improve this paradigm for CLIP training through individualized temperature optimization. Later, FastCLIP [9] develops an efficient distributed CLIP training framework for optimizing a global contrastive objective. Two recent works [38, 39] address the challenge of estimating the normalization term of the softmax function by training a normalization prediction network.

Empirically, scaling laws suggest that increasing data and model capacity consistently leads to stronger CLIP models. On the data side, large-scale datasets such as Datacomp-1B [40], DFN-2B [6], LAION-5B [41], and the WebLi dataset with 10B data [42] provide high-volume web image–text pairs for training. On the model side, architectures have scaled from small Vision Transformers (ViTs) [43] with a few million parameters to billion-parameter regimes. While data and model scaling is effective for improving performance, it also significantly increases the training budget, making it inaccessible to most researchers and developers. Recently, [44] proposed a novel approach DRRHO-CLIP for improving the scaling law of CLIP training by using a pre-trained CLIP model as a reference model, which significantly reduces the burden on large-scale data.The key distinction between our work and these prior studies is that they train CLIP models from scratch, whereas we fine-tune an already well-pretrained CLIP model.

Improving pretrained CLIP. The most prominent direction in this setting is supervised fine-tuning, which relies on labeled data, pseudo-labels, or task-specific supervision. FLYP [12], WiSE-FT [18] constructs image–text pairs using class labels, while LP-FT [45] performs linear probing, again requiring labeled data. Similarly, [46] leverage label information in a two-stage framework based on General-Purpose Retrieval Fine-Tuning (GPR) [47]. Another line of work improves downstream performance by learning textual prompts corresponding to class labels [48, 49, 50]. In contrast, our work advances a label-agnostic paradigm, aiming instead to enhance CLIP's generality while preserving its robust pretrained representations.

As discussed earlier, SSFT of pretrained CLIP often suffers from early-stage performance degradation. Although not widely emphasized, this phenomenon has been observed in prior work. CLIP-Refine [23] highlights this issue, showing that directly optimizing the contrastive loss can degrade performance relative to the pretrained model. To address this, it introduces image–text feature alignment combined with a hybrid contrastive loss. While effective, its improvements rely on high-quality image–text pairs (e.g., COCO [51]). In our experiments, we directly compare against this framework and demonstrate the advantages of our method using web corpora.

Continual Learning. An orthogonal line of work studies continual learning for CLIP, focusing on mitigating catastrophic forgetting under distributional shifts [24, 14, 25, 26]. These methods are designed to preserve performance as models adapt to new tasks or data distributions. However, we address an optimization issue that causes performance degradation even when the fine-tuning data closely matches the pretraining distribution.

3 Preliminaries

Notations: Let $\mathcal{D} = \{(x_i, z_i)\}_{i=1}^n$ be a dataset of n image–text pairs, where x_i denotes the i-th image and z_i denotes its corresponding text description. Given CLIP model $\mathcal{M}$ (with $|\mathcal{M}|$ parameters), we learn two separate encoders. We define $\mathbf{f}(\cdot)$ and $\mathbf{g}(\cdot)$ as the encoders for images and texts, parameterized by $\boldsymbol{\theta_1}$ and $\boldsymbol{\theta_2}$, respectively. For ease, we define the joint parameter of the image and text encoders as $\boldsymbol{\omega} = [\boldsymbol{\theta_1}, \boldsymbol{\theta_2}]$. To ensure that cosine similarity can be consistent with the inner product, both encoders output ℓ_2 normalized vector representations in $\mathbb{R}^d$. Thus the cosine similarity between an image x_i and a text z_j is $\mathbf{s}_{i,j} = \mathbf{f}(x_i; \boldsymbol{\omega})^\top \mathbf{g}(z_j; \boldsymbol{\omega})$. To discuss algorithms later, we need the notations for a mini-batch, so let us consider $\mathcal{B} \subset \mathcal{D}$ having $B = |\mathcal{B}|$ samples to be a mini-batch sampled from the full dataset $\mathcal{D}$.

Mini-batch Contrastive Loss (MBCL). The standard mini-batch based contrastive loss for a batch $\mathcal{B}$ is given by [1]:

$$\mathcal{L}_{\text{MBCL}}(\boldsymbol{\omega}) = -\frac{1}{|\mathcal{B}|}\sum_{i=1}^{|\mathcal{B}|}\left[\log\frac{\exp(\mathbf{s}_{i,i}/\tau)}{\sum_{j=1}^{|\mathcal{B}|}\exp(\mathbf{s}_{i,j}/\tau)} + \log\frac{\exp(\mathbf{s}_{i,i}/\tau)}{\sum_{j=1}^{|\mathcal{B}|}\exp(\mathbf{s}_{j,i}/\tau)}\right]. \tag{1}$$

which encourages high similarity for positive image–text pairs and low similarity for negative pairs in the shared $\mathbb{R}^d$ space. Here $\tau > 0$ is the temperature parameter. [2] used this loss to train OpenCLIP models.

Global Contrastive Loss (GCL). One limitation of optimizing MBCL is that it requires a large batch size in order to achieve competitive performance. To address this issue, we follow previous works [27] and use a Global Contrastive Loss (GCL). Without loss of generality, let us introduce a pairwise loss $\ell(\mathbf{s}_{j,i} - \mathbf{s}_{i,i})$, which measures the loss on the difference between a negative data pair and a positive data pair. Then we define two functions $\Phi_1(\cdot), \Phi_2(\cdot)$ for image-anchor data and text-anchor data, respectively, i.e.,

$$\Phi_1(\boldsymbol{\omega}, i, \mathcal{D}) = \frac{1}{n}\sum_{z_j \in \mathcal{D}\setminus\{z_i\}} \exp\left(\frac{\ell(\mathbf{s}_{i,j} - \mathbf{s}_{i,i})}{\tau}\right),\ \Phi_2(\boldsymbol{\omega}, i, \mathcal{D}) = \frac{1}{n}\sum_{x_j \in \mathcal{D}\setminus\{x_i\}} \exp\left(\frac{\ell(\mathbf{s}_{j,i} - \mathbf{s}_{i,i})}{\tau}\right),$$

Then GCL can be defined as:

$$\mathcal{L}_{\text{GCL}}(\boldsymbol{\omega}) = \frac{\tau}{n}\sum_{i=1}^{n}\left[\log\big(\varepsilon + \Phi_1(\boldsymbol{\omega}, i, \mathcal{D})\big) + \log\big(\varepsilon + \Phi_2(\boldsymbol{\omega}, i, \mathcal{D})\big)\right], \tag{2}$$

where $\varepsilon > 0$ is a small constant that increases numerical stability. Without explicitly mentioned, we consider $\ell(\cdot) = \cdot$ for GCL as used in [9] for CLIP training from scratch.

Optimization Algorithms. A fundamental challenge of optimizing GCL is that it lacks unbiased stochastic gradient estimator. To see this, the gradient of $\mathcal{L}_{\text{GCL}}(\boldsymbol{\omega})$ is given by

$$\nabla\mathcal{L}_{\text{GCL}}(\boldsymbol{\omega}) = \frac{\tau}{n}\sum_{i=1}^{n}\left[\frac{1}{\varepsilon + \Phi_1(\boldsymbol{\omega}, i, \mathcal{D})}\nabla\Phi_1(\boldsymbol{\omega}, i, \mathcal{D}) + \frac{1}{\varepsilon + \Phi_2(\boldsymbol{\omega}, i, \mathcal{D})}\nabla\Phi_2(\boldsymbol{\omega}, i, \mathcal{D})\right]$$

Since $\Phi_*(\boldsymbol{\omega}, i, \mathcal{D})$ is the denominator, simply using their mini-batch estimator will yield a biased gradient estimator. To address this issue, [27] propose an algorithm SogCLR, which maintains and updates an estimator $u_{i,x}, u_{i,z}$ for each $\Phi_1(\boldsymbol{\omega}, i, \mathcal{D})$ and $\Phi_2(\boldsymbol{\omega}, i, \mathcal{D})$ along the optimization trajectory. At the t-iteration with a mini-batch $\mathcal{B}_t$, they are updated by

$$u_{i,x}^{(t)} = (1-\gamma_t)u_{i,x}^{(t-1)} + \gamma_t\,\Phi_1(\boldsymbol{\omega}_{t-1}, i, \mathcal{B}_t), \qquad u_{i,z}^{(t)} = (1-\gamma_t)u_{i,z}^{(t-1)} + \gamma_t\,\Phi_2(\boldsymbol{\omega}_{t-1}, i, \mathcal{B}_t), \tag{3}$$

Then a stochastic gradient estimator of $\mathcal{L}_{GCL}$ w.r.t. the shared parameters $\boldsymbol{\omega}$ at iteration t is:

$$G(\boldsymbol{\omega}_{t-1}, \mathcal{B}_t) = \frac{\tau}{|\mathcal{B}_t|}\sum_{i\in\mathcal{B}_t}\left[\frac{1}{\varepsilon + u_{i,x}^{(t)}}\,\nabla_{\boldsymbol{\omega}}\Phi_1(\boldsymbol{\omega}_{t-1}, i, \mathcal{B}_t) + \frac{1}{\varepsilon + u_{i,z}^{(t)}}\,\nabla_{\boldsymbol{\omega}}\Phi_2(\boldsymbol{\omega}_{t-1}, i, \mathcal{B}_t)\right]. \tag{4}$$

Then the first-order moment is updated followed by a model parameter update:

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1-\beta_1)G(\boldsymbol{\omega}_{t-1}, \mathcal{B}_t) \\ \boldsymbol{\omega}_t &= \boldsymbol{\omega}_{t-1} - \eta_t m_t. \end{aligned} \tag{5}$$

[9] has designed a distributed optimization framework FastCLIP based on the above algorithm for large-scale CLIP training.

Algorithm 1: Optimizer Statistics Recovery (OSR)

Init: $\boldsymbol{\omega_0}$ (Pretrained), $m_0 \leftarrow [0]^{|\mathcal{M}|}$, $v_0 \leftarrow [0]^{|\mathcal{M}|}$, $u_x^{(0)} \leftarrow [0]^{|\mathcal{D}|}$, $u_z^{(0)} \leftarrow [0]^{|\mathcal{D}|}$
for *iteration* $t = 1$ **to** T **do**
- Sample $\mathcal{B}_t \subset \mathcal{D}$ `// mini-batch sampling`
- $u_{i,x}^{(t)} \leftarrow (1-\gamma_t)u_{i,x}^{(t-1)} + \gamma_t\, \Phi_1(\boldsymbol{\omega_0}, i, \mathcal{B}_t), \forall i \in \mathcal{B}_t$ `// refer (3)`
- $u_{i,z}^{(t)} \leftarrow (1-\gamma_t)u_{i,z}^{(t-1)} + \gamma_t\, \Phi_2(\boldsymbol{\omega_0}, i, \mathcal{B}_t), \forall i \in \mathcal{B}_t$
- Compute $g_t = G(\boldsymbol{\omega_0}, \mathcal{B}_t)$ `// frozen` $\boldsymbol{\omega_0}$
- Update $m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1)g_t$ `// first moment`
- Update $v_t \leftarrow \beta_2 v_{t-1} + (1-\beta_2)(g_t \odot g_t)$ `// second moment`

Return : $m^\star \leftarrow m_T,\ v^\star \leftarrow v_T,\ u^\star \leftarrow \{u_{i,x}^{(T)}, u_{i,z}^{(T)}\}_{i\in\mathcal{D}}$

4 TuneCLIP: A Two-Stage framework for Self-Supervised fine-tuning

As outlined in the problem statement, our goal is to adapt pretrained parameters $\boldsymbol{\omega}_0$ to obtain refined weights $\boldsymbol{\omega}^\star$ that improve performance across diverse domains. In the following two subsections, we will discuss the challenges and present our solutions.

4.1 Stage I: Optimizer Statistics Recovery (OSR)

A naive approach for SSFT with a pretrained model $\mathcal{M}$ with weights $\boldsymbol{\omega}_0$ is to just run OpenCLIP [2] or FastCLIP [9] algorithms on an existing self-supervised learning dataset $\mathcal{D}$ with an initialization of $\boldsymbol{\omega}_0$. Our hypothesis is that an open-weight pretrained model $\boldsymbol{\omega}_0$ (e.g., OpenAI's ViT-B/16) is usually not an optimal model. However, we observe a performance degradation in the first epoch of fine-tuning, see Figure 1, with the details of training deferred to Section 5. This phenomenon is common regardless of the model structure and datasets used for fine-tuning.

To understand this phenomenon, we provide a theoretical analysis of optimization error. We consider the optimization algorithm SogCLR used by FastCLIP, and note that analysis of OpenCLIP's optimization algorithm suffer from the same issue. To run FastCLIP algorithm, we need to initialize several statistics, including m_0 and $u_{i,x}^{(0)}, u_{i,z}^{(0)}, \forall i$. These statistics are usually initialized to mini-batch estimators of the first iteration in standard pretraining from scratch. Below, we show that their estimation error has a great impact on the convergence. To simplify the presentation, we introduce the following notations: $u_x^{(t)} = [u_{1,x}^{(t)}, \cdots, u_{n,x}^{(t)}], u_z^{(t)} = [u_{1,z}^{(t)}, \cdots, u_{n,z}^{(t)}]$, $\Phi_1(\boldsymbol{\omega_0}, \mathcal{D}) = [\Phi_1(\boldsymbol{\omega_0}, 1, \mathcal{D}), \cdots, \Phi_1(\boldsymbol{\omega_0}, n, \mathcal{D})], \Phi_2(\boldsymbol{\omega_0}, \mathcal{D}) = [\Phi_2(\boldsymbol{\omega_0}, 1, \mathcal{D}), \cdots, \Phi_2(\boldsymbol{\omega_0}, n, \mathcal{D})]$. Due to space limitations, all necessary assumptions and theorem proofs in this section are deferred to Appendix A.

Theorem 4.1. *Let us consider the updates in (equation 5) with initializations $u_x^{(0)}, u_z^{(0)}$, and m_0. Under appropriate assumptions, with $1-\beta_1 = O(B\epsilon^2)$, $\gamma = O(B\epsilon^2)$ and $\eta = O(\frac{B^2\epsilon^2}{n})$, we can find an ϵ-stationary point $\boldsymbol{\omega}$ such that $\mathbb{E}[\|\nabla \mathcal{L}_{\mathrm{GCL}}(\boldsymbol{\omega})\|] \leq \epsilon$ in*

$$T = O\left(\frac{n}{B^2\epsilon^4}\left(\Delta_0 + \frac{B}{n}M_0 + U_{x,0} + U_{z,0}\right)\right)$$

iterations, where $\Delta_0 = \mathcal{L}_{\mathrm{GCL}}(\boldsymbol{\omega_0}) - \min_{\boldsymbol{\omega}} \mathcal{L}_{\mathrm{GCL}}(\boldsymbol{\omega})$, $M_0 = \|m_0 - \nabla_\omega \mathcal{L}_{\mathrm{GCL}}(\boldsymbol{\omega_0})\|^2$, $U_{x,0} = \frac{1}{2n}\left\|u_x^{(0)} - \Phi_1(\boldsymbol{\omega_0}, \mathcal{D})\right\|^2$, $U_{z,0} = \frac{1}{2n}\left\|u_z^{(0)} - \Phi_2(\boldsymbol{\omega_0}, \mathcal{D})\right\|^2$.

Remark: The above theorem exhibits how the initial estimation errors of $u_x^{(0)}, u_z^{(0)}$, and m_0 affects the iteration complexity for finding an ϵ-stationary solution. Since a pretrained model $\boldsymbol{\omega_0}$ is already well trained, we expect Δ_0 to be small. However, the initial estimation errors of $u_x^{(0)}, u_z^{(0)}$, and m_0 could be very large if they are initialized poorly. It is these errors that cause the break in convergence and hence the performance degradation at the beginning of training. We refer to this issue caused by the initial estimation errors of statistics $\Phi_1(\boldsymbol{\omega_0}, \mathcal{D}), \Phi_2(\boldsymbol{\omega_0}, \mathcal{D}), \nabla_\omega \mathcal{L}_{\mathrm{GCL}}$ as cold-start bias.

To address the cold-start bias, we propose a simple method that aims to compute a better estimation of statistics $\Phi_1(\boldsymbol{\omega_0}, \mathcal{D}), \Phi_2(\boldsymbol{\omega_0}, \mathcal{D}), \nabla_\omega \mathcal{L}_{\mathrm{GCL}}$ for updating $\boldsymbol{\omega_0}$. The idea is just run the update (equation

Algorithm 2: TuneCLIP Algorithm

Given: $\boldsymbol{\omega}_0$ (Pretrained), dataset $\mathcal{D}$, batch size $|\mathcal{B}|$, epochs E', τ, margin m, γ_t, Adam (β_1, β_2)
$(m^\star, v^\star, \{u^\star_{i,x}, u^\star_{i,z}\}_{i\in\mathcal{D}}) \leftarrow \text{OSR}(\boldsymbol{\omega}_0, \mathcal{D})$ `// Alg. 1`

Init: $\boldsymbol{\omega} \leftarrow \boldsymbol{\omega}_0$; $m_0 \leftarrow m^\star$; $v_0 \leftarrow v^\star$; $u^{(0)}_{i,x} \leftarrow u^\star_{i,x}$, $u^{(0)}_{i,z} \leftarrow u^\star_{i,z}$ for all $i \in \mathcal{D}$

for *iteration* $t = 1$ **to** T' **do**

  Sample $\mathcal{B}_t \subset \mathcal{D}$ `// mini-batch sampling`

  for *each* $i \in \mathcal{B}_t$ **do**

    $\Phi^m_1(\boldsymbol{\omega}, i, \mathcal{B}_t) \leftarrow \frac{1}{|\mathcal{B}_t|}\sum_{z_j\in\mathcal{B}_t\setminus\{z_i\}} \exp\left(\frac{\ell(\mathbf{s}_{i,j}-\mathbf{s}_{i,i})}{\tau}\right)$ `// (8)`

    $\Phi^m_2(\boldsymbol{\omega}, i, \mathcal{B}_t) \leftarrow \frac{1}{|\mathcal{B}_t|}\sum_{x_j\in\mathcal{B}_t\setminus\{x_i\}} \exp\left(\frac{\ell(\mathbf{s}_{j,i}-\mathbf{s}_{i,i})}{\tau}\right)$ `// (8)`

    $u^{(t)}_{i,x} \leftarrow (1-\gamma_t)u^{(t-1)}_{i,x} + \gamma_t\,\Phi^m_1(\boldsymbol{\omega}, i, \mathcal{B}_t)$, $u^{(t)}_{i,z} \leftarrow (1-\gamma_t)u^{(t-1)}_{i,z} + \gamma_t\,\Phi^m_2(\boldsymbol{\omega}, i, \mathcal{B}_t)$

  $\tilde{g}_t \leftarrow \frac{\tau}{|\mathcal{B}_t|}\sum_{i\in\mathcal{B}_t}\left[\frac{1}{\varepsilon + u^{(t)}_{i,x}}\nabla_{\boldsymbol{\omega}}\Phi^m_1(\boldsymbol{\omega}, i, \mathcal{B}_t) + \frac{1}{\varepsilon + u^{(t)}_{i,z}}\nabla_{\boldsymbol{\omega}}\Phi^m_2(\boldsymbol{\omega}, i, \mathcal{B}_t)\right]$

  Update m_t, v_t, and $\boldsymbol{\omega}$ using Adam-style optimizer with gradient $\tilde{g}_t$

Return: $\boldsymbol{\omega}^\star \leftarrow \boldsymbol{\omega}$ `// Best parameters after last iteration`

5) with the model parameter fixed at $\boldsymbol{\omega_0}$. We present the details in Algorithm 1, which is referred to as optimizer statistics recovery (OSR). The following theorem provides a guarantee that the estimation errors of returned statistics of OSR would be much reduced. In practice, we also compute a second moment estimator for using Adam optimizer, which improves performance in our experiments.

Theorem 4.2. *Let Algorithm 1 run for E epochs (equivalently $T = E\cdot\frac{n}{B}$ iterations) with $1-\beta_1 = O(\sqrt{\frac{B}{E}}), \gamma = O(\sqrt{\frac{B}{E}})$, we have that:*

$$\mathbb{E}\left[\frac{1}{2n}\left\|u^{(\tau)}_x - \Phi_1(\boldsymbol{\omega_0}, \mathcal{D})\right\|^2 + \frac{1}{2n}\left\|u^{(\tau)}_z - \Phi_2(\boldsymbol{\omega_0}, \mathcal{D})\right\|^2\right] \leq O\left(\frac{U_{x,0}+U_{z,0}}{\sqrt{BE}} + \frac{1}{\sqrt{BE}}\right), \quad (6)$$

$$\mathbb{E}\left[\left\|m_\tau - \nabla_\omega\mathcal{L}_{\text{GCL}}(\boldsymbol{\omega_0})\right\|^2\right] \leq O\left(\frac{\frac{B}{N}M_0 + U_{x,0}+U_{z,0}}{\sqrt{BE}} + \frac{1}{\sqrt{BE}}\right) \quad (7)$$

where $\tau \in \{0, \ldots, T-1\}$ is randomly sampled.

We observe that $E = 5$ epochs for OSR is good enough to ensure stable training in the second stage of updating the model parameters.

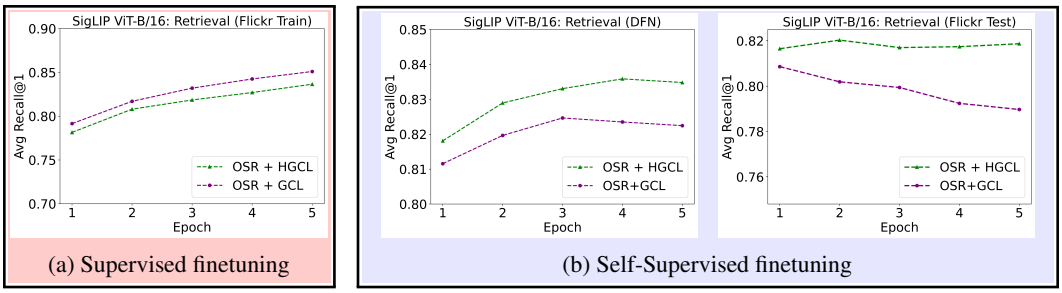
We also note that MBCL is a special case of GCL with $\gamma = 1$, i.e., replacing Eq. (3) with $u^{(t)}_{i,x} = \Phi_1(\omega_{t-1}, i, \mathcal{B}_t)$ and $u^{(t)}_{i,z} = \Phi_2(\omega_{t-1}, i, \mathcal{B}_t)$. Thus, a similar analysis applies to MBCL, yielding a similar dependence on the error of gradient momentum initialization. For more detail analyses, please refer to Appendix. A.3

4.2 Stage II: Hinged Global Contrastive Loss

With accurate initializations of m_0 and $u^{(0)}_{i,x}, u^{(0)}_{i,z}, \forall i$ found by OSR, we continue fine-tuning $\boldsymbol{\omega_0}$ with the SogCLR algorithm. This brings evident improvement across a variety of tasks. However, one issue is that the retrieval performance could still decline as fine-tuning progresses. We depict result of the fine-tuning of SigLIP ViT-B/16 on the DFN dataset (see Figure 2b), where the retrieval performance on the fine-tuning dataset keeps increasing but the retrieval performance on testing data such as Flickr decreases. This trend holds across pretrained models (Figure 5, Appendix E).

We attribute this generalization gap to the prevalence of false negatives in web-scale datasets. By optimizing GCL with $\ell(\cdot) = \cdot$, we keep decreasing the similarity gap $s_{ij} - s_{ii}$ and $s_{ji} - s_{ii}$ across iterations. If (x_i, z_j) are semantically similar, e.g., z_j is the caption of an image x_j that is semantically similar to x_i, then minimizing $s_{ij} - s_{ii}$ would distort well learnt embeddings of x_i, z_j. This strict separation on training data will undermine the testing performance due to distributional shift. This is the reason that leads to the retrieval performance drop.

To mitigate this over-penalization of false negatives, we introduce a simple yet effective remedy by using a hinge-based pairwise surrogate loss $\ell(s_{ij} - s_{ii}) = \max(s_{ij} - s_{ii} + m, 0)^2$ with $m > 0$ being



(a) Supervised finetuning (b) Self-Supervised finetuning

Figure 2: In supervised fine-tuning (red), OSR+GCL outperforms OSR+HGCL (TuneCLIP) because true negative labels justify separating negatives. In contrast, under self-supervised fine-tuning (blue), the absence of such labels makes OSR+HGCL more suitable, leading to improved retrieval performance on Flickr when fine-tuned with SSFT (see Appendix E for OpenAI CLIP & details).

a margin hyperparameter constant. It means that as long as $s_{ii} > s_{ij} + m$, its gradient will become zero and hence will not affect the model updates anymore. Illustrative examples of this phenomenon are provided in Table 22 (Appendix L). Accordingly, we define new $\Phi_1^m(.)$ (image-anchored) and $\Phi_2^m(.)$ (text-anchored) as:

$$\Phi_1^m(\boldsymbol{\omega}, i, \mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{j \in \mathcal{D} \setminus \{i\}} \exp\left(\frac{\ell(\mathbf{s}_{i,j} - \mathbf{s}_{i,i})}{\tau}\right), \qquad \ell(\mathbf{s}_{i,j} - \mathbf{s}_{i,i}) = [\,\mathbf{s}_{i,j} - \mathbf{s}_{i,i} + m\,]_+^2,$$
$$\Phi_2^m(\boldsymbol{\omega}, i, \mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{j \in \mathcal{D} \setminus \{i\}} \exp\left(\frac{\ell(\mathbf{s}_{j,i} - \mathbf{s}_{i,i})}{\tau}\right), \qquad \ell(\mathbf{s}_{j,i} - \mathbf{s}_{i,i}) = [\,\mathbf{s}_{j,i} - \mathbf{s}_{i,i} + m\,]_+^2. \tag{8}$$

Equation 8 leads to the Hinged Global Contrastive Loss (HGCL), defined below.

$$\mathcal{L}_{\text{HGCL}}(\boldsymbol{\omega}) = \frac{\tau}{|\mathcal{D}|} \sum_{i \in \mathcal{D}} \left[\log\big(\varepsilon + \Phi_1^m(\boldsymbol{\omega}, i, \mathcal{D})\big) + \log\big(\varepsilon + \Phi_2^m(\boldsymbol{\omega}, i, \mathcal{D})\big)\right]. \tag{9}$$

We optimize $\mathcal{L}_{\text{HGCL}}$ using the SogCLR algorithm with OSR. Algorithm 2 presents the details of our final algorithm named TuneCLIP, combining OSR with HGCL.

Crucially, we note that the margin m is a hyperparameter that controls how aggressively negatives are separated from the positive. A larger m enforces stricter separation, pushing more false negatives downward until their similarity ($s_{i,j}$ or $s_{j,i}$) lies at least m below the positive score ($s_{i,i}$), even when they start with relatively high similarity. Conversely, a smaller m relaxes this constraint, allowing higher-scoring false negatives to be retained but at the risk of insufficient separation of true negatives. Choosing m therefore presents a *tradeoff* between alleviating the over-suppression of semantically related false negatives and preventing true negatives from remaining too close to the anchor.

5 Experiments

CLIP models & datasets. We explore a range of pretrained CLIP models at different scales (ranging from 151 to 987 million parameters), including OpenAI's CLIP ViT-B/32 and ViT-B/16 [1], LAION's CLIP ViT-B/32 [41] and SigLIP ViT-B/16 [42], where ViT-B/X refers to ViT based image encoder. We report results for fine-tuning OpenAI's ViT-B/16 and SigLIP ViT-B/16 in the main paper, and provide results of fine-tuning other models in Appendix G. We additionally evaluated our method for fine-tuning the state-of-the-art CLIP ViT-H/14 pretrained on DFN-5B [6]. To study how performance scales with data under fixed training conditions, we fine-tune on two subsets of the DFN datasets [6] containing 12 million (DFN-12M) and 60 million (DFN-60M) samples.

Training hyperparameters & algorithms. We run OSR for $E = 5$ epochs, and another $E' = 5$ epochs for fine-tuning. Optimizer used is AdamW [52] (β_1=0.9, β_2=0.98). The CLIP temperature τ remains fixed as provided with checkpoint (no scheduling). We sweep learning rates $\{10^{-4}, 10^{-5}, 10^{-6}\}$. Batch sizes are 256×8 GPUs for ViT-B/16 and 512×8 GPUs for ViT-B/32 CLIPs. The margin m is swept from 0.5 down to 0.01, with values around 0.1 proving to be the most effective across the majority of architectures. More details are provided in Appendices B & C.

Table 1: Summary of mean zero-shot performance across ImageNet & distributional variants, retrieval benchmarks, and the DataComp benchmark, together with wall-clock training time per GPU.

Base Model	Method	Wall C.T. (hrs)	IN & Variants	Retrieval	DataComp
OpenAI ViT-B/16	Baseline	N/A	57.67	57.46	56.26
	FastCLIP	4.21	54.57 (↓)	51.88 (↓)	53.53 (↓)
	OpenCLIP	5.46	54.99 (↓)	57.81	55.11 (↓)
	CLIP-Ref.	1.01	55.96(↓)	58.82	56.18(↓)
	TuneCLIP	**8.62**	**59.36** (+1.69)	**64.12** (+6.66)	**58.62** (+2.36)
SigLIP ViT-B/16	Baseline	N/A	63.12	69.32	62.32
	FastCLIP	4.28	39.22 (↓)	43.37 (↓)	45.80 (↓)
	OpenCLIP	7.55	40.21 (↓)	51.54 (↓)	48.10 (↓)
	CLIP-Ref.	1.07	39.12(↓)	49.83 (↓)	44.72(↓)
	TuneCLIP	**9.27**	**65.58** (+2.46)	**69.44** (+0.11)	**63.47** (+1.15)

Table 2: Impact of OSR statistics transfers on HGCL fine-tuning (OpenAI ViT-B/16 CLIP): (m_t, v_t) yield a large gain, with u_t providing additional improvement for the best overall performance.

(m_t, v_t) u_t	IN & Variants	Retrieval	DataComp	Mean
(×, ×) ×	54.91	58.64	54.49	56.01
(✓, ✓) ×	**59.48** (+4.57)	63.70	58.56	60.58
(✓, ✓) ✓	59.36	**64.12** (+5.48)	**58.62** (+4.13)	**60.70** (+4.69)

Evaluation protocol and metrics. We follow the DataComp protocol [40] and use 38 benchmark datasets. Our main results are reported in three evaluation groups: (1) ImageNet-1k and six robustness variants [16] for assessing zero-shot classification accuracy, (2) MSCOCO [51] and Flickr30k [17] for measuring multi & single-object retrieval performance, and (3) the full DataComp [40] benchmark.

Baselines. We compare with three baselines, FastCLIP [9], OpenCLIP [2] and CLIP-Refine [23]. We run 5 epochs for all except for CLIP-Refine, which is run for 1 epoch as suggested in their work.

Main Results. We present results on three evaluation suites for fine-tuning various models on DFN-12M in Table 1. We observe that TuneCLIP delivers substantial gains over the base model, most notably for OpenAI ViT-B/16 with 6.7% improvement on retrieval and 1.7% improvement on zero-shot classification, while improvements for SigLIP are smaller given its stronger baseline. In contrast, the baseline methods OpenCLIP [2] and FastCLIP [9] not only fail to improve the performance over the base model but also suffer significant performance drop in retrieval and zero-shot classification. Similar to other baselines, CLIP-Refine also fails to consistently improve performance. This behavior aligns with observations in [23], where performance is shown to be sensitive to data quality. In contrast, TuneCLIP avoids such degradation and delivers stable gains. For reference, we also provide more detailed results for all 38 benchmarks in Appendix G.

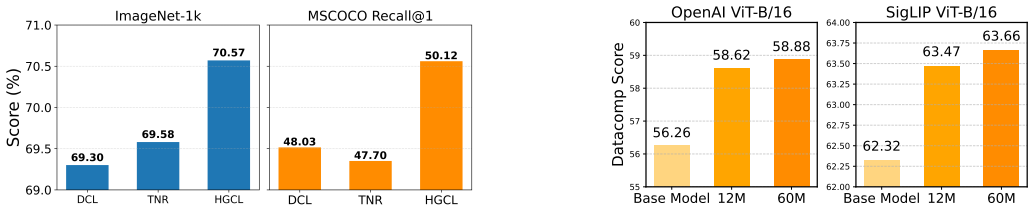
Table 1 also reports wall-clock time and GPU-hours of different methods. Despite a modest overhead from its two-stage design, TuneCLIP consistently improves performance, whereas baselines show limited gains even with similar compute budget. This is illustrated in Figure 9 (Appendix F).

We also plot the curves of zero-shot classification performance on ImageNet-1k across training epochs for different checkpoints of TuneCLIP for fine-tuning OpenAI ViT-B/16 in Figures 1, and for other models in Appendix F.

Finally, TuneCLIP for fine-tuning the state-of-the-art model ViT-H/14-quickgelu [6] achieves new SOTA accuracy on ImageNet and its variants as shown in Appendix J, surpassing ViT-H/14 at 224×224 image resolution by about 1.5% (from 71.80% to 73.23%), while maintaining comparable performance on Retrieval and DataComp.

5.1 Ablation, Scaling & Comparison of TuneCLIP

We begin with an ablation study on the effect of OSR, comparing TuneCLIP with full statistics recovery, partial recovery of m_t and v_t, and no recovery at all. The results are reported in Table 2, which shows that using the full recovered statistics from OSR achieves the best, and the recovery



(a) (b)

Figure 3: (a) Comparison of different false negative mitigation strategies in the second stage. DCL extends debiased contrastive learning [30] to CLIP, TNR removes top negatives during gradient estimation (based on [31]). All use OSR in first stage, initialized from OpenAI B/16. (b) Effect of data scaling on TuneCLIP, demonstrating stable monotonic improvements as data scale increases.

of first and second-order moments is more important than the recovery of u_t (i.e., u_x, u_z). Beyond the ablations conducted within OSR itself, we also compare against simple cold-start mitigation heuristics that practitioners might reasonably try to stabilize fine-tuning, as presented in Appendix I (Table 19). These alternatives offer only limited stability and smaller gains, reinforcing that OSR provides a more effective and reliable solution to cold-start bias.

Table 3: TuneCLIP performance across datasets of varying noise levels.

Method	Data	IN & Variants	Retrieval	DataComp
OpenAI CLIP	–	57.67	57.46	56.26
TuneCLIP	CC12M	57.68 (+0.01)	**65.83** (+8.37)	56.47 (+0.21)
TuneCLIP	DFN12M	**59.36** (+1.69)	64.12 (+6.66)	**58.62** (+2.36)

We also conduct an ablation study comparing GCL with HGCL, both with OSR for supervised fine-tuning and SSFT. For supervised fine-tuning, we fine-tune a pretrained model on the training set of Flickr30k data and evaluate on a testing set of Flickr1k. For SSFT, we finetune the same pretrained model on DFN-12M and evaluate on the same testing set of Flickr1k. The results are shown in Figure 2 for fine-tuning SigLIP ViT-B/16 and in Figure 5 (Appendix E) for fine-tuning OpenAI's CLIP ViT-B/16. These results show that GCL with OSR performs better for supervised fine-tuning, while HGCL with OSR is superior for SSFT, highlighting the role of false negatives in SSFT and the effectiveness of HGCL in improving retrieval performance. In Appendix L, we further show that HGCL reduces the variance of similarity scores for false negatives. Additionally, the true positive (top-1) distribution becomes closer to that of false negatives after fine-tuning. We analyze the sensitivity of the HGCL margin parameter m in Figure 4 (Appendix C).

We further compare HGCL with prior false negative mitigation methods [30, 31] in Figure 3(a), which shows that HGCL consistently outperforms both baselines.

We also evaluate TuneCLIP on CNN-based CLIP, showing improvements beyond Vision Transformer backbones (Table 21, Appendix K). Finally, we examine how TuneCLIP scales with the amount of fine-tuning data and how it performs under varying data quality. As shown in Figure 3(b), performance remains stable even with a fivefold increase (DFN12M to DFN60M). While additional data yields modest gains, improvements are limited as fine-tuning operates on already well-structured pretrained representations. The DFN datasets are constructed by filtering web image-text pairs using a learned filter, and thus form a relatively less noisy source of self-supervised training data. To examine robustness to raw web-scale (noisy corpora), we fine-tune TuneCLIP on CC12M [53], a web dataset with weaker image-text alignment. As shown in Table 3, TuneCLIP consistently achieves positive gains across both datasets, with larger improvements on the less noisy DFN-12M.

6 Conclusion

TuneCLIP solves a core problem in model adaptation by showing how to fine-tune a pre-trained model into a superior version with broad, multi-domain improvements. Our two-stage approach combines optimizer recovery with a hinge-based loss to deliver consistent, interpretable gains across

tasks. This work goes beyond proposing a new method, opening a concrete and promising direction for self-supervised fine-tuning toward more general-purpose foundation model enhancement.

7 Limitations and Future Works

One limitation of this work is that we use all data for self-supervised fine-tuning without data selection or filtering. As a future direction, we consider how to select the most informative data given the knowledge of the pretrained model to accelerate the fine-tuning. Extending our framework beyond CLIP models to other self-supervised architectures (e.g. DINO) is also an interesting direction.

Another limitation is that our framework uses a fixed temperature and a fixed margin, with the margin tuned for the chosen temperature. Making the margin adaptive or learnable is an interesting extension. How to optimize these parameters jointly while preserving the hinge-based loss's ability to mitigate false negatives remains an interesting direction for future work.

Acknowledgments and Disclosure of Funding

We thank Texas A&M University's High Performance Research Computing (HPRC) and the National Center for Supercomputing Applications (NCSA) for providing access to supercomputing resources. A. Mehta, X. Chen, X. Wei, and T. Yang were partially supported by National Science Foundation Award #2306572.

References

[1] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[2] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2818–2829, 2023.

[3] Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. How much can clip benefit vision-and-language tasks? *arXiv preprint arXiv:2107.06383*, 2021.

[4] Zihao Zhao, Yuxiao Liu, Han Wu, Mei Wang, Yonghao Li, Sheng Wang, Lin Teng, Disheng Liu, Zhiming Cui, Qian Wang, et al. Clip in medical imaging: A comprehensive survey. *arXiv preprint arXiv:2312.07353*, 2023.

[5] Tenglong Ao, Zeyi Zhang, and Libin Liu. Gesturediffuclip: Gesture diffusion model with clip latents. *ACM Transactions on Graphics (TOG)*, 42(4):1–18, 2023.

[6] Alex Fang, Albin Madappally Jose, Amit Jain, Ludwig Schmidt, Alexander Toshev, and Vaishaal Shankar. Data filtering networks. *arXiv preprint arXiv:2309.17425*, 2023.

[7] Zi-Hao Qiu, Quanqi Hu, Zhuoning Yuan, Denny Zhou, Lijun Zhang, and Tianbao Yang. Not all semantics are created equal: Contrastive self-supervised learning with automatic temperature individualization. *arXiv preprint arXiv:2305.11965*, 2023.

[8] Zi-Hao Qiu, Siqi Guo, Mao Xu, Tuo Zhao, Lijun Zhang, and Tianbao Yang. To cool or not to cool? temperature network meets large foundation models via dro. *arXiv preprint arXiv:2404.04575*, 2024.

[9] Xiyuan Wei, Fanjiang Ye, Ori Yonay, Xingyu Chen, Baixi Sun, Dingwen Tao, and Tianbao Yang. Fastclip: A suite of optimization techniques to accelerate clip training with limited resources. *arXiv preprint arXiv:2407.01445*, 2024.

[10] Bac Nguyen, Stefan Uhlich, Fabien Cardinaux, Lukas Mauch, Marzieh Edraki, and Aaron Courville. Saft: Towards out-of-distribution generalization in fine-tuning. In *European Conference on Computer Vision*, pages 138–154. Springer, 2024.

[11] Tejas Srinivasan, Xiang Ren, and Jesse Thomason. Curriculum learning for data-efficient vision-language alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5619–5624, 2023.

[12] Sachin Goyal, Ananya Kumar, Sankalp Garg, Zico Kolter, and Aditi Raghunathan. Finetune like you pretrain: Improved finetuning of zero-shot vision models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19338–19347, 2023.

[13] Yuxuan Ding, Lingqiao Liu, Chunna Tian, Jingyuan Yang, and Haoxuan Ding. Don't stop learning: Towards continual learning for the clip model. *arXiv preprint arXiv:2207.09248*, 2022.

[14] Saurav Jha, Dong Gong, and Lina Yao. Clap4clip: Continual learning with probabilistic finetuning for vision-language models. *Advances in neural information processing systems*, 37:129146–129186, 2024.

[15] William Yang, Byron Zhang, and Olga Russakovsky. Imagenet-ood: Deciphering modern out-of-distribution detection algorithms. *arXiv preprint arXiv:2310.01755*, 2023.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[17] Roelof Van Zwol. Flickr: Who is looking? In *IEEE/WIC/ACM International Conference on Web Intelligence (WI'07)*, pages 184–190. IEEE, 2007.

[18] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7959–7971, 2022.

[19] Xiaoyi Dong, Jianmin Bao, Ting Zhang, Dongdong Chen, Shuyang Gu, Weiming Zhang, Lu Yuan, Dong Chen, Fang Wen, and Nenghai Yu. Clip itself is a strong fine-tuner: Achieving 85.7% and 88.0% top-1 accuracy with vit-b and vit-l on imagenet. *arXiv preprint arXiv:2212.06138*, 2022.

[20] Mohammad Fahes, Tuan-Hung Vu, Andrei Bursuc, Patrick Pérez, and Raoul de Charette. *Fine-tuning clip's last visual projector: A few-shot cornucopia*. PhD thesis, Inria, 2024.

[21] Tian Liu, Huixin Zhang, Shubham Parashar, and Shu Kong. Few-shot recognition via stage-wise retrieval-augmented finetuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15086–15097, June 2025.

[22] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ B. Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S. Chen, Kathleen Creel, Jared Quincy Davis, Dorottya Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren E. Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, and et al. On the opportunities and risks of foundation models. *CoRR*, abs/2108.07258, 2021.

[23] Shin'ya Yamaguchi, Dewei Feng, Sekitoshi Kanai, Kazuki Adachi, and Daiki Chijiwa. Post-pre-training for modality alignment in vision-language foundation models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 4256–4266, 2025.

[24] Wenzhuo Liu, Fei Zhu, Longhui Wei, and Qi Tian. C-clip: Multimodal continual learning for vision-language model. In *The Thirteenth International Conference on Learning Representations*, 2025.

[25] Linhui Xiao, Xiaoshan Yang, Fang Peng, Ming Yan, Yaowei Wang, and Changsheng Xu. Clip-vg: Self-paced curriculum adapting of clip for visual grounding. *IEEE Transactions on Multimedia*, 26:4334–4347, 2023.

[26] Saurabh Garg, Mehrdad Farajtabar, Hadi Pouransari, Raviteja Vemulapalli, Sachin Mehta, Oncel Tuzel, Vaishaal Shankar, and Fartash Faghri. Tic-clip: Continual training of clip models. *arXiv preprint arXiv:2310.16226*, 2023.

[27] Zhuoning Yuan, Yuexin Wu, Zi-Hao Qiu, Xianzhi Du, Lijun Zhang, Denny Zhou, and Tianbao Yang. Provable stochastic optimization for global contrastive learning: Small batch does not harm performance. In *International Conference on Machine Learning*, pages 25760–25782. PMLR, 2022.

[28] Tsai-Shien Chen, Wei-Chih Hung, Hung-Yu Tseng, Shao-Yi Chien, and Ming-Hsuan Yang. Incremental false negative detection for contrastive learning. *CoRR*, abs/2106.03719, 2021.

[29] Vicente Balmaseda, Bokun Wang, Ching-Long Lin, and Tianbao Yang. Discovering global false negatives on the fly for self-supervised contrastive learning. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, Proceedings of Machine Learning Research. PMLR / OpenReview.net, 2025.

[30] Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. *Advances in neural information processing systems*, 33:8765–8775, 2020.

[31] Tri Huynh, Simon Kornblith, Matthew R Walter, Michael Maire, and Maryam Khademi. Boosting contrastive self-supervised learning with false negative cancellation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2785–2795, 2022.

[32] Yanghao Li, Haoqi Fan, Ronghang Hu, Christoph Feichtenhofer, and Kaiming He. Scaling language-image pre-training via masking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 23390–23400, 2023.

[33] Shashank Goel, Hritik Bansal, Sumit Bhatia, Ryan A. Rossi, Vishwa Vinay, and Aditya Grover. Cyclip: Cyclic contrastive language-image pretraining. *CoRR*, abs/2205.14459, 2022.

[34] Lijie Fan, Dilip Krishnan, Phillip Isola, Dina Katabi, and Yonglong Tian. Improving CLIP training with language rewrites. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[35] Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*, 2023.

[36] Yihao Chen, Xianbiao Qi, Jianan Wang, and Lei Zhang. Disco-clip: A distributed contrastive loss for memory efficient clip training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22648–22657, 2023.

[37] Yixuan Wei, Han Hu, Zhenda Xie, Ze Liu, Zheng Zhang, Yue Cao, Jianmin Bao, Dong Chen, and Baining Guo. Improving clip fine-tuning performance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5439–5449, 2023.

[38] Xiyuan Wei, Chih-Jen Lin, and Tianbao Yang. NeuCLIP: Efficient large-scale CLIP training with neural normalizer optimization. In *The Fourteenth International Conference on Learning Representations*, 2026.

[39] Haotian Sun, Yitong Li, Yuchen Zhuang, Niao He, Hanjun Dai, and Bo Dai. AmorLIP: Efficient language-image pretraining via amortization. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026.

[40] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, et al. Datacomp: In search of the next generation of multimodal datasets. *Advances in Neural Information Processing Systems*, 36:27092–27112, 2023.

[41] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35:25278–25294, 2022.

[42] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986, 2023.

[43] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[44] Xiyuan Wei, Ming Lin, Fanjiang Ye, Fengguang Song, Liangliang Cao, My T. Thai, and Tianbao Yang. Model steering: Learning with a reference model improves generalization bounds and scaling laws. In *Proceedings of International Conference on Machine Learning*, 2025.

[45] Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. *arXiv preprint arXiv:2202.10054*, 2022.

[46] Konstantin Schall, Kai Uwe Barthel, Nico Hezel, and Klaus Jung. Optimizing clip models for image retrieval with maintained joint-embedding alignment. In *International Conference on Similarity Search and Applications*, pages 97–110. Springer, 2024.

[47] Konstantin Schall, Kai Uwe Barthel, Nico Hezel, and Klaus Jung. Gpr1200: a benchmark for general-purpose content-based image retrieval. In *International Conference on Multimedia Modeling*, pages 205–216. Springer, 2022.

[48] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022.

[49] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16816–16825, 2022.

[50] Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19113–19122, 2023.

[51] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):652–663, 2016.

[52] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[53] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3558–3568, 2021.

[54] Bokun Wang and Tianbao Yang. Finite-sum coupled compositional stochastic optimization: Theory and applications. *arXiv preprint arXiv:2202.12396*, 2022.

[55] Xingyu Chen, Bokun Wang, Ming Yang, Qihang Lin, and Tianbao Yang. Stochastic momentum methods for non-smooth non-convex finite-sum coupled compositional optimization. *Advances in Neural Information Processing Systems*, 38:78671–78712, 2026.

[56] Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. *Advances in neural information processing systems*, 32, 2019.

[57] Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, et al. The visual task adaptation benchmark. 2019.

[58] Vikram V Ramaswamy, Sing Yu Lin, Dora Zhao, Aaron Adcock, Laurens van der Maaten, Deepti Ghadiyaram, and Olga Russakovsky. Geode: a geographically diverse evaluation dataset for object recognition. *Advances in Neural Information Processing Systems*, 36:66127–66137, 2023.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: Yes. Our claims regarding TuneCLIP's improvements over baselines and its superior performance are supported by extensive experiments presented in Section 5 and Appendix G. Theoretical analysis for the framework is presented in Section 4.1 and Appendix A.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discuss the limitations of this work in Section 7. In particular, we note that improved data selection and filtering strategies could further enhance performance, which is currently limited by the availability of noisy data.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: We provide complete proofs and the associated assumptions for all theorems in Appendix A.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: We present the full TuneCLIP algorithm in Section 4 (Algorithm 2). We further detail the architectures, hyperparameters, and experimental setup in Appendix B and C to ensure reproducibility. The code is available at `https://github.com/Optimization-AI/TuneCLIP`.

Guidelines:

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

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: All training datasets and evaluation datasets described in Section 5 are publicly available. The code is available at `https://github.com/Optimization-AI/TuneCLIP`.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: All training and evaluation details are given in Section 5, Appendix Sections B and C.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: Due to the substantial computational cost of large-scale CLIP training, and following prior works such as [2, 44], we do not report error bars or confidence intervals for the main experimental results in Section 5.

   Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Compute resource details are provided in Appendix H.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: Research conducted follows NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [N/A]

Justification: There is no societal impact of the work performed.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All asset owners are properly credited, including datasets, models, and codebases.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: This paper does not release new assets at submission time. All datasets used are publicly available, and code will be released after decisions.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: LLMs were not used in research.

    Guidelines:

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

A Proof for theorems in subsection 4.1

In this part we consider a general FCCO problem:

$$\min_{\omega} \frac{1}{N} \sum_{i=1}^{N-1} f(g_i(\omega))$$

and the corresponding SOX algorithm [54]. As discussed in preliminary, by specifying $N = 2n$, $f(\cdot) = \log(\epsilon + \cdot)$, $g_i(\cdot) = g_i(\cdot, \mathcal{D}) = \Phi_1(\cdot, i, \mathcal{D})$ if $i \leq n$ otherwise $\Phi_2(\cdot, i-n, \mathcal{D})$, we recover the GCL loss and the SogCLR algorithm. We also set $|\mathcal{B}_1| = |\mathcal{B}_2| = B$ in SOX when presenting its convergence analysis to simplify notation. Before starting our proofs, we make the following standard, commonly used assumptions as in [54] under which theorems in subsection 4.1 hold:

Assumption A.1. We assume that:
• $f(\cdot)$ and $\nabla f(\cdot)$ are C_f and L_f-Lipschitz continuous, respectively.
• $g_i(\cdot)$ and $\nabla g_i(\cdot)$ are C_g and L_g-Lipschitz continuous, respectively.

Assumption A.2. There exist constants $\sigma_0 \geq 0$ and $\sigma_1 \geq 0$ such that the following statements hold for $g_i(\omega)$ and $g_i(\omega, \xi_i)$ for $i = 1, \ldots, N$ for any $\omega \in \mathbb{R}^d$ $\mathbb{E}\|g_i(\omega, \xi_i) - g_i(\omega)\|^2 \leq \sigma_0^2$, $\mathbb{E}\left\|\nabla g_i(\omega, \xi_i) - \nabla g_i(\omega)\right\|^2 \leq \sigma_1^2$.

A.1 Technical Lemma

The following descent lemma is commonly used in the analysis of momentum methods for non-convex optimization(see e.g. [54, 55]).

Lemma A.3. *Consider a sequence $\omega_{t+1} = \omega_t - \eta m_{t+1}$ and the L_F-smooth function F and the step size $\eta L_F \leq 1/2$.*

$$F(\omega_{t+1}) \leq F(\omega_t) + \frac{\eta}{2} M_t - \frac{\eta}{2} \left\|\nabla F(\omega_t)\right\|^2 - \frac{\eta}{4} \left\|m_{t+1}\right\|^2, \tag{10}$$

where $M_t \coloneqq \left\|m_{t+1} - \nabla F(\omega_t)\right\|^2$.

We build a recursion for the gradient variance M_t by proving the following lemma.

Lemma A.4. *If $\beta \leq \frac{2}{7}$, the gradient variance M_t can be bounded as*

$$\mathbb{E}\left[M_{t+1}\right] \leq (1-\beta)\mathbb{E}\left[M_t\right] + \frac{2L_F^2\eta^2}{\beta}\mathbb{E}\left[\left\|m_{t+1}\right\|^2\right] + \frac{2\beta^2 C_f^2(\sigma_1^2 + C_g^2)}{B} + 5\beta L_f^2 C_1^2 \mathbb{E}\left[U_{t+1}\right] \tag{11}$$

where $U_t = \frac{1}{N}\left\|u_{t+1} - g(\omega_t; \mathcal{D})\right\|^2$, $u_t = [u_1^{(t)}, \ldots, u_N^{(t)}]^\top$, $g(\omega_t; \mathcal{D}) = [g_1(\omega_t; \mathcal{D}_1), \ldots, g_N(\omega_t; \mathcal{D}_N)]^\top$ and $C_1^2 = C_g^2 + \frac{\sigma_1^2}{B}$. Also note that we follow the tradition usage of β in [54] so $\beta = 1 - \beta_1$ where β_1 is used in algorithm 1.

Remark: We point out that the lemma is similar to lemma 9 in [54] without a term corresponding to $\left\|u_{t+1} - u_t\right\|^2$, the gap is caused by the different usage of u when constructing the overall gradient estimator $G(\omega_{t-1}, \mathcal{B}_t)$: instead of using old u_{t-1} as in SOX we use a newer version u_t in this paper. This would require sampling one more iid minibatch per iteration to derive a bound as shown in the lemma. However in practice we typically sample only a single minibatch.

Lemma A.5. *If $\gamma \leq 1/5$, function value variance U_t can be bounded as*

$$\mathbb{E}\left[U_{t+1}\right] \leq \left(1 - \frac{\gamma B}{4N}\right)\mathbb{E}\left[U_t\right] + \frac{5N\eta^2 C_g^2}{\gamma B}\mathbb{E}\left[\left\|m_{t+1}\right\|^2\right] + \frac{2\gamma^2\sigma_0^2}{N} \tag{12}$$

Remark: We directly drop the negative term in lemma 2 in [54].

A.2 Proof of theorem

Proof of Theorem 4.1. The proof is almost the same as theorem 3 in [54] so we only make necessary clarifications. Summing (10), $\frac{\eta}{\beta}\times$(11), and $\frac{20L_f^2C_1^2N\eta}{\gamma B}\times$(12) leads to

$$\begin{aligned}
&\mathbb{E}\left[F(\omega_{t+1}) - F^* + \frac{\eta}{\beta}M_{t+1} + \frac{20L_f^2C_1^2N\eta}{\gamma B}\left(1-\frac{\gamma B}{4N}\right)U_{t+1}\right] \\
&\leq \mathbb{E}\left[F(\omega_t) - F^* + \frac{\eta}{\beta}\left(1-\frac{\beta}{2}\right)M_t + \frac{20L_f^2C_1^2N\eta}{\gamma B}\left(1-\frac{\gamma B}{4N}\right)U_t\right] - \frac{\eta}{2}\mathbb{E}\left[\|\nabla F(\omega_t)\|^2\right] \\
&\quad - \eta\left(\frac{1}{4} - \frac{2L_F^2\eta^2}{\beta^2} - \frac{100L_f^2N^2C_1^2\eta^2C_g^2}{\gamma^2}\right)\mathbb{E}\left[\|m_{t+1}\|^2\right] + \frac{2\beta\eta C_f^2(\sigma_1^2+C_g^2)}{B} + \frac{40\eta\gamma L_f^2C_1^2\sigma_0^2}{B}.
\end{aligned}$$

Set $\beta = \min\{\frac{B\epsilon^2}{12C_f^2(\sigma_1^2+C_g^2)}, \frac{2}{7}\}$, $\gamma = \min\left\{\frac{B\epsilon^2}{240L_f^2C_1^2\sigma_0^2}, \frac{1}{5}\right\}$, and $\eta = \min\left\{\frac{\beta}{4L_F}, \frac{\gamma B}{30L_fNC_1C_g}\right\}$. Define the Lyapunov function as $\Gamma_t \coloneqq F(\omega_t) - F^* + \frac{\eta}{\beta}M_t + \frac{20L_f^2C_1^2N}{B}\frac{\eta}{\gamma}\left(1-\frac{\gamma B}{4N}\right)U_t$. Then,

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla F(\omega_t)\|^2\right] \leq \frac{2\Gamma_0}{\eta T} + \frac{4\beta C_f^2(\sigma_1^2+C_g^2)}{B} + \frac{80\gamma L_f^2C_1^2\sigma_0^2}{B}, \tag{13}$$

discarding the non-dominant terms and unimportant constants, to guarantee $\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla F(\omega_t)\|^2\right] \leq \epsilon^2$ we need at most

$$T = O\left(\frac{n\Gamma_0}{B^2\epsilon^4}\right) = O\left(\frac{n}{B^2\epsilon^4}(\Delta_0 + \frac{B}{n}M_0 + U_0)\right)$$

iterations, which leads to conclusion by noting that $U_0 = \frac{1}{N}\|u_0 - g(\boldsymbol{\omega_0};\mathcal{D})\|^2 = \frac{1}{2n}\left\|u_x^{(0)} - \Phi_1(\boldsymbol{\omega_0},\mathcal{D})\right\|^2 + \frac{1}{2n}\left\|u_z^{(0)} - \Phi_2(\boldsymbol{\omega_0},\mathcal{D})\right\|^2$.

□

Proof of Theorem 4.2. Note that algorithm 1 is essentially SOX [54] without updating the model parameters ω(i.e. learning rate $\eta = 0$), we can still leverage lemma A.4 and A.5 by plugging $\eta = 0$ into them and have the following bound:

$$\mathbb{E}\left[M_{t+1}\right] \leq (1-\beta)\mathbb{E}\left[M_t\right] + \frac{2\beta^2C_f^2(\sigma_1^2+C_g^2)}{B} + 5\beta L_f^2C_1^2\mathbb{E}\left[U_{t+1}\right] \tag{14}$$

$$\mathbb{E}\left[U_{t+1}\right] \leq \left(1-\frac{\gamma B}{4N}\right)\mathbb{E}\left[U_t\right] + \frac{2\gamma^2\sigma_0^2}{N} \tag{15}$$

Note that now we are not updating ω so $M_t = \|m_{t+1} - \nabla_\omega\mathcal{L}_{\mathrm{GCL}}(\boldsymbol{\omega_0})\|^2$, $U_t = \frac{1}{N}\|u_{t+1} - g(\boldsymbol{\omega_0};\mathcal{D})\|^2$. Rearranging terms and divide both side for (14), (15) by β and $\frac{\gamma B}{4N}$, respectively, then we have:

$$\mathbb{E}\left[M_t\right] \leq \frac{1}{\beta}\mathbb{E}\left[M_t - M_{t+1}\right] + \frac{2\beta C_f^2(\sigma_1^2+C_g^2)}{B} + 5L_f^2C_1^2\mathbb{E}\left[U_{t+1}\right] \tag{16}$$

$$\mathbb{E}\left[U_t\right] \leq \frac{4N}{\gamma B}\mathbb{E}\left[U_t - U_{t+1}\right] + \frac{8\gamma\sigma_0^2}{B} \tag{17}$$

combining the above two inequalities we have

$$\begin{aligned}
\mathbb{E}\left[M_t\right] &\leq \frac{1}{\beta}\mathbb{E}\left[M_t - M_{t+1}\right] + \frac{2\beta C_f^2(\sigma_1^2+C_g^2)}{B} + 5L_f^2C_1^2\left((\frac{4N}{\gamma B}-1)\mathbb{E}\left[U_t - U_{t+1}\right] + \frac{8\gamma\sigma_0^2}{B}\right) \\
&\leq \mathbb{E}\left[\Psi_t - \Psi_{t+1}\right] + \frac{2\beta C_f^2(\sigma_1^2+C_g^2)}{B} + \frac{40\gamma\sigma_0^2L_f^2C_1^2}{B}
\end{aligned} \tag{18}$$

where $\Psi_t = \frac{1}{\beta} M_t + \frac{20NL_f^2C_1^2}{\gamma B}\left(1 - \frac{\gamma B}{4N}\right) U_t$. Sum over $t = 0, 1, \cdots, T-1$ and divide both side by T then we have

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[M_t] \leq \frac{\Psi_0}{T} + \frac{2\beta C_f^2(\sigma_1^2 + C_g^2)}{B} + \frac{40\gamma\sigma_0^2 L_f^2 C_1^2}{B} \tag{19}$$

Convergence of U_t can be easily derived from (17) by summing over $t = 0, 1, \cdots, T-1$ and divide both side by T:

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[U_t] \leq \frac{4NU_0}{\gamma BT} + \frac{8\gamma\sigma_0^2}{B} \tag{20}$$

By setting $\beta = O(\sqrt{\frac{N}{T}}), \gamma = O(\sqrt{\frac{N}{T}})$ and omitting unimportant constants, we have

$$\mathbb{E}_{\hat{t}}\left[\mathbb{E}[M_{\hat{t}}]\right] = \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[M_t] \leq O\left(\frac{M_0}{\sqrt{NT}} + \frac{U_0}{\sqrt{BE}} + \frac{1}{\sqrt{BE}}\right) \tag{21}$$

$$\mathbb{E}_{\hat{t}}\left[\mathbb{E}[U_{\hat{t}}]\right] = \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[U_t] \leq O\left(\frac{U_0}{\sqrt{BE}} + \frac{1}{\sqrt{BE}}\right) \tag{22}$$

which directly leads to the conclusion by noting that $N = 2n, T = \frac{nE}{B}$ and $U_0 = U_{x,0} + U_{z,0}$. □

A.3 Convergence Analysis of MBCL

We only highlight the difference to the analysis in the previous section by noting that MBCL is a special case of GCL with $\gamma = 1$. Note that in this case we have $u_{i,x}^{(t)} = \Phi_1(\omega_{t-1}, i, \mathcal{B}_t)$ and $u_{i,z}^{(t)} = \Phi_2(\omega_{t-1}, i, \mathcal{B}_t)$, which leads to a naive bound:

$$U_t \leq \frac{\sigma_0^2}{B} \tag{23}$$

Plug it in eqn. (11). Then summing (10), $\frac{\eta}{\beta}\times$(11) leads to

$$\begin{aligned}
&\mathbb{E}\left[F(\omega_{t+1}) - F^* + \frac{\eta}{\beta}M_{t+1}\right] \\
&\leq \mathbb{E}\left[F(\omega_t) - F^* + \frac{\eta}{\beta}\left(1 - \frac{\beta}{2}\right)M_t\right] - \frac{\eta}{2}\mathbb{E}\left[\|\nabla F(\omega_t)\|^2\right] \\
&\quad - \eta\left(\frac{1}{4} - \frac{2L_F^2\eta^2}{\beta^2}\right)\mathbb{E}\left[\|m_{t+1}\|^2\right] + \frac{2\beta\eta C_f^2(\sigma_1^2 + C_g^2)}{B} + \frac{5\eta L_f^2 C_1^2\sigma_0^2}{B}.
\end{aligned}$$

Set $B = \Theta(\frac{15L_f^2C_1^2\sigma_0^2}{\epsilon^2})$, $\beta = \min\{\frac{B\epsilon^2}{12C_f^2(\sigma_1^2+C_g^2)}, \frac{2}{7}\}$ and $\eta = \Theta(\beta)$. Define the Lyapunov function as $\Gamma_t \coloneqq F(\omega_t) - F^* + \frac{\eta}{\beta}M_t$. Then,

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla F(\omega_t)\|^2\right] \leq \frac{2\Gamma_0}{\eta T} + \frac{4\beta C_f^2(\sigma_1^2 + C_g^2)}{B} + \frac{10L_f^2C_1^2\sigma_0^2}{B}, \tag{24}$$

discarding the non-dominant terms and unimportant constants, to guarantee $\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla F(\omega_t)\|^2\right] \leq \epsilon^2$, the sample complexity is

$$T * B = O\left(\frac{\Gamma_0}{\epsilon^4}\right) = O\left(\frac{\Delta_0 + M_0}{\epsilon^4}\right)$$

iterations, which concludes the proof.

Remark: The above analysis shows that the iteration complexity of MBCL also depends on the initial estimation error of m_0.

B Additional details on CLIP models

The CLIP models (ViT-B/32 and ViT-B/16) use an embedding dimension of 512 for contrastive learning. In contrast, SigLIP employs a larger embedding dimension of 768. Tables 4 and 5 summarize the configurations of the vision and text encoders, respectively. Table 6 further reports the overall model specifications, including parameter counts and developers.

Table 4: Vision tower configurations of CLIP models.

Model	Image Size	Layers	Width	Patch Size
CLIP ViT-B/32	224	12	768	32
CLIP ViT-B/16	224	12	768	16
SigLIP ViT-B/16	224	12	768	16

Table 5: Text tower configurations of CLIP models.

Model	Context Length	Vocab Size	Width	Heads	Layers
CLIP ViT-B/32	77	49408	512	8	12
CLIP ViT-B/16	77	49408	512	8	12
SigLIP ViT-B/16	64	32000	768	12	12

Table 6: Model specifications of different CLIP variants.

Model	Vision Encoder	Text Encoder	Parameters (M)	Developer
CLIP ViT-B/32	ViT	Transformer	151.28	OpenAI [1]
CLIP ViT-B/16	ViT	Transformer	149.62	OpenAI [1]
CLIP ViT-B/32	ViT	Transformer	151.28	LAION [41]
SigLIP ViT-B/16	ViT	Transformer	203.16	Google [42]

Table 7: Key hyperparameters used for training the models.

Hyperparameter	Value
Image size	224x224
Learning rate (lr)	1e-5
Optimizer	AdamW
Beta1, Beta2	0.9, 0.98
Weight decay (wd)	0.02
Scheduler	Cosine
Precision	AMP (mixed precision)
Margin	0.1
Margin smoothing	2.0
Gamma	0.9
Gamma schedule	Cosine (decay every 4 epochs)
World size	8 (DDP)
Workers	6

C Hyperparameter details

A brief summary of the key hyperparameters is provided in Table 7. Owing to the availability of 40GB A100s and 80GB H100s, we conducted all experiments to an image resolution of 224×224. Across models, the best-performing base learning rate was consistently around 1×10^{-5}. We used cosine scheduling on the learning rate in the second stage of fine-tuning. We adopted AdamW as the

optimizer with momentum parameters $(\beta_1, \beta_2) = (0.9, 0.98)$, and a weight decay of 0.02 to improve generalization. A cosine learning rate scheduler was used to provide smooth decay to γ, following a cosine schedule until the 4th epoch and fixed to 0.9 thereafter. We also applied mixed-precision training (*AMP*) to balance performance and efficiency. Each experiment used a world size of 8 for DDP and 6 data-loading workers per GPU to optimize throughput.

C.1 Ablation on different learning rates

Table 8: Learning rate sweep on TuneCLIP

Learning Rate	ImageNet-1k (%)	MS COCO (Avg R@1) (%)
1e-4	69.66	48.98
1e-5	70.57	50.11
5e-6	70.23	49.30

After sweeping learning rates across $\{10^{-4}, 10^{-5}, 10^{-6}\}$, we observe that performance drops slightly above 10^{-4}, while learning rates in the range of 10^{-6} to 10^{-5} remain comparably strong. For example, Table 8 shows that TuneCLIP achieves stable performance around 1e–5 and 5e–6.

C.2 Ablation on margin m used with HGCL

To study the impact of the margin m, we consider two different CLIP architectures, LAION ViT-B/32 and OpenAI ViT-B/16, and sweep over a representative set of values $m \in \{0.01, 0.05, 0.10, 0.20, 0.40\}$. The resulting ImageNet-1k accuracies are plotted in Figure 4. Using this as a representative metric, we observe that margins around $m = 0.1$ consistently perform well across different model scales.

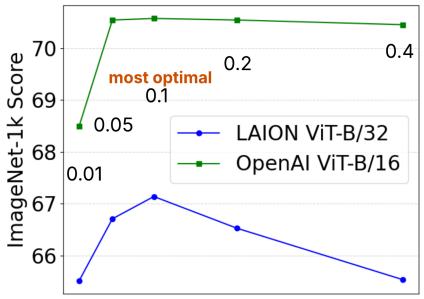


Figure 4: Sensitivity of margin (m).

D Baseline details

Table 9: Training configurations for compared methods.

Method	Loss	Optimization Strategy
FastCLIP [9]	GCL	SogCLR + AdamW
OpenCLIP [2]	MBCL	AdamW
CLIP-Refine [23]	RaFA+HyCD[23]	AdamW
TuneCLIP (ours)	HGCL (ours)	OSR (ours) + SogCLR + AdamW

Table 9 summarizes the training setups of the algorithms used in our comparison. FastCLIP [9] employs the standard Global Contrastive Loss (GCL) with the SogCLR optimization algorithm and AdamW. OpenCLIP [2] relies on a minibatch contrastive loss (MBCL) combined with AdamW. CLIP-Refine [23] uses random feature alignment (RaFA) and hybrid contrastive-distillation (HyCD). Our TuneCLIP introduces the proposed Hinged Global Contrastive Loss (HGCL) loss and leverages Optimizer Statistics Recovery (OSR) alongside SogCLR and AdamW.

E Impact of Hinged Global Contrastive Loss

The controlled study in Figures 2, 5 and 6 is set up as follows: supervised fine-tuning is performed on Flickr30k, while SSFT uses DFN-12M. Train Retrieval trends are computed from 15,000 random DFN samples (SSFT) and 1,000 Flickr30k samples (SFT). Both models are evaluated on the Flickr1k test set.

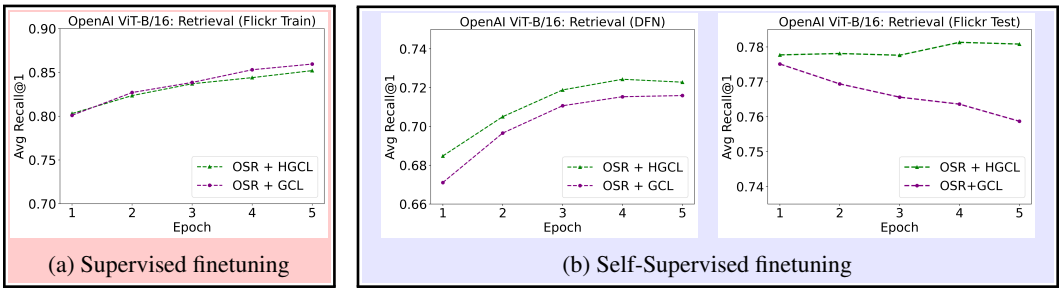


Figure 5: Similar to Figure 2, supervised fine-tuning (red) shows stronger performance with OSR+GCL than with OSR+HGCL (TuneCLIP), since true negative labels justify separating negatives. By contrast, in self-supervised fine-tuning (blue), the absence of such labels makes OSR+HGCL more effective.

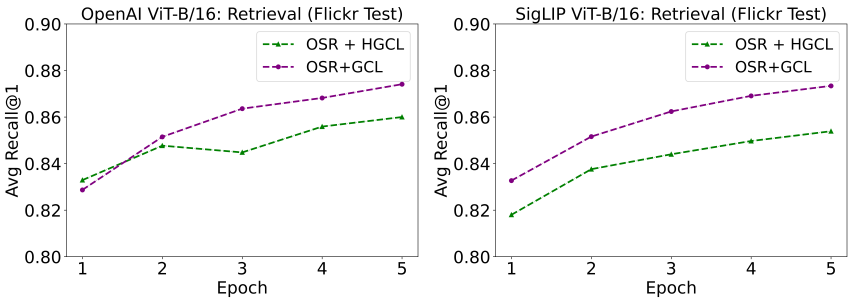
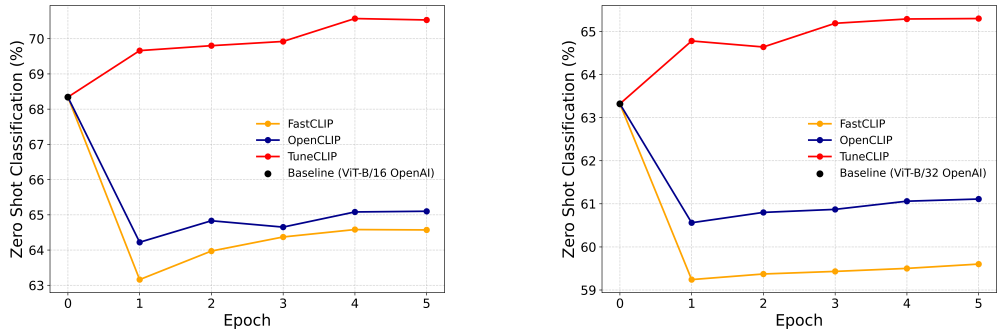


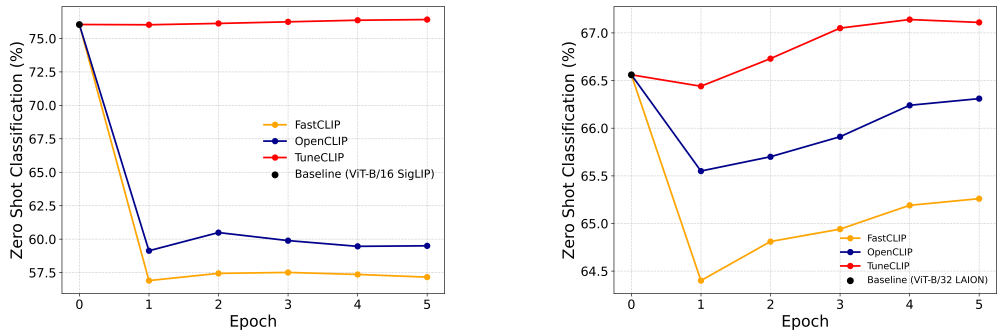
Figure 6: Retrieval trends on the Flickr1k test set under supervised fine-tuning. In the supervised setting, explicit labels guide the separation of positives from negatives, making OSR+GCL outperform OSR+HGCL. By contrast, as shown in Figures 2b and 5b, the absence of supervision and margin regularization in SSFT reverses this trend, with OSR+HGCL (TuneCLIP) achieving superior retrieval performance.

F Fine-tuning performance on ImageNet-1k

Across all four models, the baselines exhibit degraded initialization and fail to recover in early epochs. In contrast, TuneCLIP consistently starts stronger and outperforms them throughout fine-tuning (Figure 7).



(a) OpenAI models.



(b) SigLIP and LAION models.

Figure 7: Zero-shot classification scores (ImageNet-1k) over 5 epochs of fine-tuning for all CLIPs.

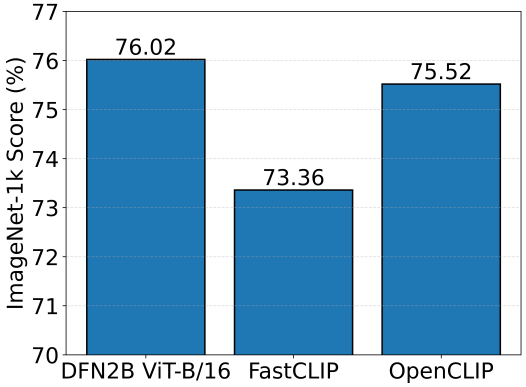


Figure 8: ImageNet-1k accuracy after *one epoch* of fine-tuning on DFN-12M (a subset of DFN2B). Despite matching the pretraining data distribution, naïve fine-tuning methods (FastCLIP, OpenCLIP) exhibit clear performance degradation compared to the DFN2B-pretrained ViT-B/16 baseline.

As shown in Fig. 9, extending fine-tuning beyond the used standard few (i.e. 5-epoch) schedule provides only diminishing returns for both the algorithms. Even when trained for up to 15 epochs, the performance curves plateau, indicating that additional compute does not meaningfully change the

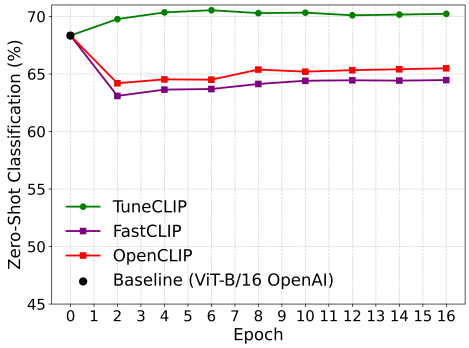


Figure 9: Extended fine-tuning analysis of TuneCLIP and baselines on ImageNet-1k.

relative ordering of methods. Importantly, TuneCLIP preserves a consistent improvement over the baselines.

G Comprehensive results on 38 DataComp benchmarks

We report the gains of TuneCLIP over the base model, with improvements highlighted in (+).

Table 10: Performance of different CLIP models on DataComp Average. DataComp [40] is a highly comprehensive benchmark that spans a diverse collection of datasets, tasks, and distributional variants. Even small improvements on DataComp are particularly meaningful, as they indicate stable gains across heterogeneous and challenging settings rather than isolated benefits on individual datasets.

Base Model	Method	DataComp Average
OpenAI ViT-B/32	Baseline	52.45
	FastCLIP	49.78
	OpenCLIP	51.02
	CLIP-Refine	51.58
	TuneCLIP	**54.34** (+1.89)
OpenAI ViT-B/16	Baseline	56.26
	FastCLIP	53.53
	OpenCLIP	55.11
	CLIP-Refine	56.18
	TuneCLIP	**58.62** (+2.36)
SigLIP ViT-B/16	Baseline	62.32
	FastCLIP	45.80
	OpenCLIP	48.10
	CLIP-Refine	44.72
	TuneCLIP	**63.47** (+1.15)
LAION ViT-B/32	Baseline	56.94
	FastCLIP	55.89
	OpenCLIP	56.75
	CLIP-Refine	**57.39**
	TuneCLIP	57.22 (+0.28)

Table 11: Performance of different CLIP models on small-scale classification benchmarks.

Base Model	Method	CIFAR-10	CIFAR-100	STL-10
OpenAI ViT-B/32	Baseline	89.83	64.23	97.13
	FastCLIP	90.54	69.51	92.68
	OpenCLIP	91.75	71.18	96.36
	CLIP-Refine	89.94	68.15	96.35
	TuneCLIP	**93.63** (+3.80)	**73.87** (+9.64)	**97.20** (+0.07)
OpenAI ViT-B/16	Baseline	90.77	66.95	98.25
	FastCLIP	92.67	71.33	96.58
	OpenCLIP	92.76	70.99	97.87
	CLIP-Refine	91.01	69.10	97.81
	TuneCLIP	**94.40** (+3.63)	**76.14** (+9.19)	**98.26** (+0.01)
SigLIP ViT-B/16	Baseline	92.34	72.23	98.21
	FastCLIP	83.66	53.86	91.18
	OpenCLIP	85.06	56.77	93.81
	CLIP-Refine	76.66	43.52	91.32
	TuneCLIP	**95.20** (+2.86)	**79.91** (+7.68)	**98.37** (+0.16)
LAION ViT-B/32	Baseline	93.58	75.55	**96.56**
	FastCLIP	92.38	75.95	91.73
	OpenCLIP	94.13	76.10	95.16
	CLIP-Refine	94.13	76.07	96.40
	TuneCLIP	**94.22** (+0.64)	**76.46** (+0.91)	96.47 (-0.09)

Table 12: Performance of different CLIP models on ImageNet-1k, ImageNet-Sketch, ImageNet-V2.

Base Model	Method	1K	Sketch	V2
OpenAI ViT-B/32	Baseline	63.32	42.29	55.92
	FastCLIP	59.59	41.26	52.25
	OpenCLIP	61.11	41.86	53.84
	CLIP-Refine	62.16	41.50	55.12
	TuneCLIP	**65.29** (+1.97)	**45.84** (+3.55)	**57.45** (+1.53)
OpenAI ViT-B/16	Baseline	68.34	48.24	61.88
	FastCLIP	64.57	46.49	56.88
	OpenCLIP	65.10	46.15	58.19
	CLIP-Refine	66.36	46.46	59.51
	TuneCLIP	**70.57** (+2.23)	**51.16** (+2.92)	**64.11** (+2.23)
SigLIP ViT-B/16	Baseline	76.04	**67.92**	68.93
	FastCLIP	57.52	12.56	49.04
	OpenCLIP	59.50	10.42	51.00
	CLIP-Refine	57.67	10.86	49.00
	TuneCLIP	**76.41** (+0.37)	65.78 (-2.14)	**69.02** (+0.09)
LAION ViT-B/32	Baseline	66.56	53.65	58.15
	FastCLIP	65.26	53.12	56.92
	OpenCLIP	66.31	53.73	58.46
	CLIP-Refine	66.65	53.83	58.79
	TuneCLIP	**67.14** (+0.58)	**54.46** (+0.81)	**59.10** (+0.95)

Table 13: Performance of CLIP models on ImageNet out-of-distribution variants [15] (A, O, R, ObjectNet). ObjectNet [56] evaluates robustness under real-world viewpoint and background shifts.

Base Model	Method	A	O	R	ObjectNet
OpenAI ViT-B/32	Baseline	**31.55**	47.75	69.33	44.31
	FastCLIP	24.84	47.20	65.21	44.32
	OpenCLIP	27.10	48.85	67.13	46.02
	CLIP-Refine	28.74	48.90	68.46	48.15
	TuneCLIP	30.72 (-0.83)	**51.11** (+3.36)	**70.67** (+1.34)	**46.45** (+2.14)
OpenAI ViT-B/16	Baseline	**49.95**	42.30	77.70	55.31
	FastCLIP	41.37	43.30	75.41	55.37
	OpenCLIP	41.57	44.05	73.23	56.67
	CLIP-Refine	44.53	43.20	74.80	56.84
	TuneCLIP	48.10 (-1.85)	**46.65** (+4.35)	**77.86** (+0.16)	**57.08** (+1.77)
SigLIP ViT-B/16	Baseline	45.41	38.15	**90.30**	55.09
	FastCLIP	21.80	35.85	59.08	38.72
	OpenCLIP	22.73	39.50	58.27	40.08
	CLIP-Refine	21.70	35.96	58.80	39.82
	TuneCLIP	**50.10** (+4.69)	**41.50** (+3.35)	88.33 (-1.97)	**67.93** (+12.84)
LAION ViT-B/32	Baseline	26.26	49.95	76.43	48.81
	FastCLIP	24.84	48.00	76.04	**51.72**
	OpenCLIP	26.92	50.40	76.05	51.01
	CLIP-Refine	26.36	**51.00**	76.37	50.37
	TuneCLIP	**27.04** (+0.78)	50.85 (+0.90)	**76.45** (+0.02)	51.34 (+2.53)

Table 14: Performance of different CLIP models on VTAB and Fairness. The Visual Task Adaptation Benchmark (VTAB) [57] evaluates performance across 12 diverse tasks. We also report performance on two fairness-oriented datasets, Dollar Street and GeoDE [58], which measure robustness to geographic and socioeconomic diversity.

Base Model	Method	VTAB Mean	Fairness Mean
OpenAI ViT-B/32	Baseline	51.81	68.02
	FastCLIP	48.51	69.52
	OpenCLIP	49.23	70.05
	CLIP-Refine	49.26	70.25
	TuneCLIP	**53.50** (+1.69)	**70.73** (+2.71)
OpenAI ViT-B/16	Baseline	53.80	72.45
	FastCLIP	50.44	74.02
	OpenCLIP	52.46	74.23
	CLIP-Refine	51.65	**75.86**
	TuneCLIP	**54.44** (+0.64)	74.96 (+2.51)
SigLIP ViT-B/16	Baseline	60.39	**78.48**
	FastCLIP	49.93	69.33
	OpenCLIP	53.38	71.55
	CLIP-Refine	47.88	71.59
	TuneCLIP	**62.66** (+2.27)	78.44 (-0.04)
LAION ViT-B/32	Baseline	55.08	71.05
	FastCLIP	55.55	69.69
	OpenCLIP	54.68	71.94
	CLIP-Refine	**55.43**	**71.72**
	TuneCLIP	55.09 (+0.01)	71.03 (-0.02)

Table 15: Retrieval performance (Recall@1) on MSCOCO and Flickr datasets.

Base Model	Method	MSCOCO		Flickr	
		IR@1	TR@1	IR@1	TR@1
OpenAI ViT-B/32	Baseline	30.44	50.12	58.78	78.90
	FastCLIP	28.75	43.23	51.49	68.19
	OpenCLIP	33.33	49.79	60.92	76.10
	CLIP-Refine	34.09	52.79	61.82	79.00
	TuneCLIP	**36.74** (+6.30)	**56.16** (+6.04)	**64.71** (+5.93)	**83.30** (+4.40)
OpenAI ViT-B/16	Baseline	33.09	52.42	62.16	82.20
	FastCLIP	31.25	45.80	56.45	74.00
	OpenCLIP	36.46	50.77	65.11	78.90
	CLIP-Refine	34.75	54.68	64.13	81.69
	TuneCLIP	**40.45** (+7.36)	**59.78** (+7.36)	**69.66** (+7.50)	**86.59** (+4.39)
SigLIP ViT-B/16	Baseline	**47.78**	65.74	**74.68**	89.10
	FastCLIP	26.91	36.46	48.33	61.79
	OpenCLIP	34.34	44.58	57.12	70.10
	CLIP-Refine	33.91	40.40	57.59	67.40
	TuneCLIP	47.64 (-0.14)	**66.36** (+0.62)	74.44 (-0.24)	**89.30** (+0.20)
LAION ViT-B/32	Baseline	39.34	56.32	66.78	**84.10**
	FastCLIP	33.37	49.34	58.66	76.99
	OpenCLIP	38.34	54.42	65.03	81.30
	CLIP-Refine	39.31	56.62	66.01	83.39
	TuneCLIP	**39.55** (+0.21)	**57.80** (+1.48)	**66.82** (+0.04)	83.20 (-0.90)

H Compute cost analysis

Table 16: Batch size specifications of different CLIP variants under our distributed training setup.

Model	Local Batch Size	Global Batch Size
OpenAI CLIP ViT-B/32	512	4096
OpenAI CLIP ViT-B/16	256	2048
LAION CLIP ViT-B/32	512	4096
SigLIP ViT-B/16	256	2048

Table 17: Unified comparison of training cost and performance across CLIP backbones.

Base Model	Method	Wall-Clock Time (hrs)	GPU-hours	DataComp
OpenAI ViT-B/32	OpenCLIP	2.66	21.28	51.02
	FastCLIP	2.22	17.76	49.78
	CLIP-Refine	0.58	4.64	51.58
	TuneCLIP	**5.05**	**40.40**	**54.34**
OpenAI ViT-B/16	OpenCLIP	5.46	43.68	55.11
	FastCLIP	4.21	33.68	53.53
	CLIP-Refine	1.01	8.08	56.18
	TuneCLIP	**8.62**	**68.96**	**58.62**
SigLIP ViT-B/16	OpenCLIP	7.55	60.40	48.10
	FastCLIP	4.28	34.24	45.80
	CLIP-Refine	1.07	8.56	44.72
	TuneCLIP	**9.27**	**74.16**	**63.47**
LAION ViT-B/32	OpenCLIP	3.10	24.80	56.75
	FastCLIP	2.33	18.64	55.89
	CLIP-Refine	0.55	4.40	**57.39**
	TuneCLIP	**4.32**	**36.24**	57.22

Table 16 summarizes the local and global batch sizes used for each backbone, where the global batch size is given by $B_{\text{global}} = 8 \times B_{\text{local}}$. Across all backbones, Table 17 additionally reports the wall-clock time (measured as the elapsed time between the start and end of fine-tuning) and the corresponding GPU-hours for all approaches. We compute GPU-hours using the standard relation

$$\text{GPU-hours} = \text{wall-clock time (hours)} \times \#\text{GPUs},$$

so that, in our case, GPU-hours directly reflect wall-clock time scaled by a factor of 8. Since TuneCLIP uses a two-stage procedure (OSR followed by HGCL fine-tuning), it naturally incurs higher compute than single-stage baselines, typically increasing the wall-clock time by about 1.5–$2\times$ for a given backbone. However, this additional cost remains modest and is consistently mirrored by improved performance across all evaluated models. Thus, TuneCLIP provides a favorable cost–performance trade-off with relatively small extra computational overhead compared to baselines and it reliably converts additional compute into gains on related benchmarks.

In addition to reporting the full TuneCLIP cost, we further dissect its two-stage schedule into OSR-only and HGCL-only components in Table 18. Across all backbones, each stage incurs a wall-clock time comparable to standard single-stage fine-tuning for OpenCLIP/FastCLIP.

Table 18: Isolated computational cost of the OSR and HGCL stages in TuneCLIP.

Base Model	Method	Wall-Clock (hrs)	GPU-hours
OpenAI ViT-B/32	OpenCLIP	2.66	21.28
	FastCLIP	2.22	17.76
	OSR only	**2.50**	**20.00**
	HGCL only	**2.55**	**20.40**
OpenAI ViT-B/16	OpenCLIP	5.46	43.68
	FastCLIP	4.21	33.68
	OSR only	**4.27**	**34.16**
	HGCL only	**4.35**	**34.80**
SigLIP ViT-B/16	OpenCLIP	7.55	60.40
	FastCLIP	4.28	34.24
	OSR only	**4.52**	**36.16**
	HGCL only	**4.75**	**38.00**
LAION ViT-B/32	OpenCLIP	3.10	24.80
	FastCLIP	2.33	18.64
	OSR only	**2.20**	**17.60**
	HGCL only	**2.12**	**16.96**

On FLOPS. Since all baseline methods use the same vision and text encoders, GPUs, and training configurations at a given model scale, the FLOPs per training step are effectively identical.

I Comparing other Cold-start bias mitigation strategies

To evaluate simpler cold-start bias mitigations, we consider two realistic alternatives to OSR. First, we apply a short learning-rate warm-up of 500 iterations using momentum SGD, gradually increasing the learning rate from 1×10^{-6} to 1×10^{-5} before switching to the standard fine-tuning stage with the same optimizer. Second, we simulate a large-batch warm-up by computing gradients with a larger global batch size (e.g., increasing OpenAI CLIP ViT-B/16's batch size from 2048 to 4096) while keeping model weights frozen, allowing gradient moments to accumulate before performing normal fine-tuning in the second stage. Infact, this serves as a cheaper approximation to OSR. As shown in Table 19, both strategies provide mild stabilization but yield noticeably smaller improvements than OSR, indicating that OSR remains most effective & reliable approach for mitigating cold-start bias.

Table 19: Comparison of cold-start mitigation strategies for TuneCLIP (on OpenAI CLIP ViT-B/16).

Cold-Start Mitigation	Stage II	OSR	IN & Variants	Retrieval	DataComp
Base Model	N/A	×	57.67	57.46	56.26
Momentum SGD	Momentum SGD	×	54.65	59.32	54.82
Larger Batch Gradients	AdamW	×	58.27	62.92	58.05
OSR	Momentum SGD	✓	57.99	62.08	57.82
OSR	AdamW	✓	**59.36**	**64.12**	**58.62**

J TuneCLIP on State-of-the-art CLIP

TuneCLIP achieves state-of-the-art performance on ImageNet and its distributional variants, improving accuracy from 71.8% to 73.23%. On retrieval and DataComp, the results are slightly lower, but remain within a tolerable band of 1% relative to the baseline.

Table 20: Performance of TuneCLIP on SOTA ViT-H/14-quickgelu across evaluation suites.

Category	Baseline	TuneCLIP
IN & Variants	71.80	**73.23**
Retrieval	74.78	73.78
DataComp	69.61	69.23

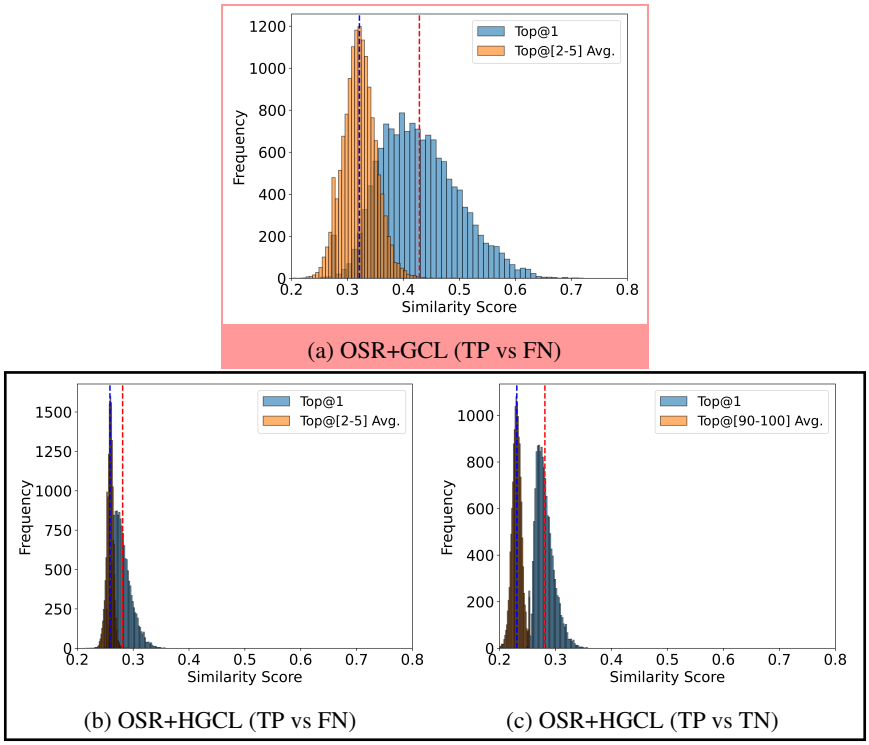
K TuneCLIP on CNN-based CLIP

Table 21: Performance of TuneCLIP on OpenAI CLIP with a ResNet-50 backbone.

Method	ImageNet-1k	MS COCO
OpenAI CLIP ResNet-50	59.82	38.69
TuneCLIP	**60.95**	**45.19**

To evaluate TuneCLIP beyond Vision Transformer backbones, we apply it to the OpenAI CLIP ResNet-50 model. As shown in Table 21, TuneCLIP improves performance across classification as well as retrieval. These results demonstrate its applicability to CNN-based CLIP models.

L False Negative mitigation through HGCL



(a) OSR+GCL (TP vs FN)

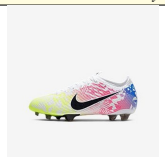
(b) OSR+HGCL (TP vs FN)

(c) OSR+HGCL (TP vs TN)

Figure 10: OSR+GCL shows higher variance among false negatives (std. dev. 0.030). While, OSR+HGCL yields a substantially more compact distribution (std. dev. 0.0061), reducing false negative bias and at the same time ensuring a clear separation between true positives ($\mu = 0.28$) and true negatives ($\mu = 0.23$).

As shown in Figures 10a and 10b, HGCL compresses the gap between true positives and false negatives, producing distributions that are both more compact and overlapping than those under GCL. This analysis, conducted on 15,000 randomly sampled pairs from DFN-12M, highlights HGCL's capacity to reduce variance and counteract false-negative bias. By preserving higher similarity for semantically related negatives, HGCL prevents over-suppression and encourages the model to maintain only a fine-grained distinctions across closely related concepts (true positives and false negatives). This in turn improves generalization, since the learned similarity space better reflects true semantic structure rather than being distorted by aggressive penalization of false negatives. Additionally, to approximate true negatives, we sample from the bottom-ranked retrievals (Top@[90-100]), as these examples are least likely to share semantic overlap with the query and thus provide a reliable baseline for unrelated pairs. Table 22 shows qualitative examples from the above distributions.

Table 22: Qualitative examples of false negatives (text captions) and their similarity scores with the anchor image. Here, s_1 denotes the similarity under OSR+GCL and s_2 under OSR+HGCL. These examples illustrate how OSR+HGCL mitigates excessive suppression of false negatives, allowing them to retain higher similarity scores compared to OSR+GCL. Conversely, false negatives that already exhibit reasonable similarity with the anchor tend to remain stable or slightly reduced, compensating for cases where suppression was more severe. Overall, this yields a calibrated similarity structure in which false negatives are assigned scores that better reflect semantic relatedness.

Anchor Image	**False Negative Captions (s_1, s_2) in Top-5 Retrieved texts**
 *Caption: FABULOUS PEARL NECKLACE DIAMOND CLUSTER 14K GOLD CLASP | eBay*	• Women's Chain Necklaces **s_1=0.2053, s_2=0.2467** • necklace image New Arrivals **s_1=0.1720, s_2=0.2429**
 *Caption: Boys' Soccer Cleats 0026 Shoes. Nike.com*	• running shoes **s_1=0.1561, s_2=0.2346**
 *Caption: dress*	• muslim wedding dresses 3d flower burgundy muslim wedding dresses 2018 arabic custom plus **s_1=0.1713, s_2=0.2238** • Evening Gowns For Mother Of The Bride In Singapore - Prom Dresses **s_1=0.1913, s_2=0.2333**
 *Caption: Chicken noodle soup*	• top down view of bowl filled with white bean tomatillo soup with items surrounding. **s_1=0.2061, s_2=0.2047** • Butternut soup with sriracha. Made it. **s_1=0.2046, s_2=0.2354**
 *Caption: pallets chair*	• B32 Office Chair by Armet 3 **s_1=0.0747, s_2=0.2189** • Burlap Ruffle Chair #burlap #ruffle #furniture **s_1=0.3096, s_2=0.2583**
 *Caption: 2017 Audi S6 4.0T quattro Premium Plus*	• gebraucht Audi S8 plus V8 4.0TFSI tiptr. UPE 154.100,- HeadUp/SD **s_1=0.2358, s_2=0.2272** • Audi Q3 und Audi RS Q3: Facelift und mehr Leistung. **s_1=0.1056, s_2=0.2174**
 *Caption: Men's Crew Neck Jersey T-Shirt - Support With Style Collection for Novak Djokovic*	• Unique Scoop Collar Printed T Shirt **s_1=0.1978, s_2=0.2433** • T-shirt z nadrukiem - white **s_1=0.1707, s_2=0.2407** • HX Hot Sale Colorful Skull 3D Print Harajuku T Shirt Grim Reaper Skull Casual T Shirt Men/women Streetwear T Shirt Tops HX768 1 **s_1=0.1560, s_2=0.2301**