BREAKING THE LIMITS OF OPEN-WEIGHT CLIP: AN OPTIMIZATION FRAMEWORK FOR SELF-SUPERVISED FINE-TUNING OF CLIP

Anonymous authors
Paper under double-blind review

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.

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 (Radford et al., 2021; Cherti et al., 2023) show broad utility across downstream tasks spanning classification, cross-modal retrieval, multimodal reasoning (Shen et al., 2021; Zhao et al., 2023) and generation (Ao et al., 2023). Recent efforts to improve CLIP have primarily focused on pretraining by constructing ever larger datasets (Fang et al., 2023), designing novel objective functions (Qiu et al., 2023; 2024), or developing refined optimization algorithms (Qiu et al., 2024; Wei et al., 2024). 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.

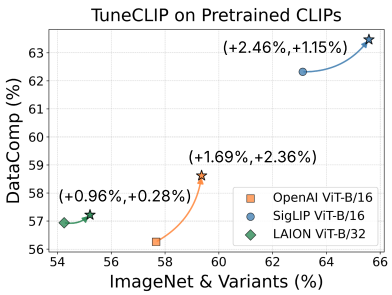


Figure 1: Improvements delivered by TuneCLIP (★) over baseline models on complementary evaluation suites: large-scale DataComp Benchmark (38 datasets) & ImageNet's 7 distributional variants.

A very common way to improve the model is supervised fine-tuning, which is performed on specific datasets of a target domain (Nguyen et al., 2024; Srinivasan et al., 2023; Goyal et al., 2023). 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 (Ding et al., 2022; Jha et al., 2024). Thus, 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 propose, namely *self-supervised fine-tuning* (SSFT), which we define as 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 (Yang et al., 2023; Krizhevsky et al., 2017; Van Zwol, 2007; Wortsman et al., 2022; Dong et al., 2022; Fahes et al., 2024; Goyal et al., 2023; Liu et al., 2025), 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 was constructed for pretraining CLIP models, e.g., (Fang et al., 2023). The result is task-agnostic corpora, positioning SSFT on them as representation refinement rather than task adaptation.

At first glance, SSFT resembles pretraining, yet key nuances render its optimization and learning process substantially more difficult. From the optimization perspective, the contrastive losses in CLIP training lack unbiased stochastic gradient estimators (Yuan et al., 2022). Consequently, the optimization error is heavily influenced by the accuracy of the gradient estimates at initialization. Standard strategies such as zero-initializing the first-order moment in Adam (Kingma & Ba, 2014) can induce large estimation errors, negating the benefits of a good initial model and causing significant performance drops at the start of training (cf. Figure 2). We refer to this issue as **cold-start bias**.

From the learning perspective, self-supervised contrastive learning suffers the issue of false negative data, i.e., those that are semantically similar to the anchor data are mistakenly treated as negatives. This issue becomes more pronounced as models improve. Fine-tuning further amplifies the gap between positive-pair and negative-pair similarities, allowing false negatives to distort embeddings of semantically similar data. Consequently, retrieval performance degrades, since it relies on ranking within a set of highly similar examples.

To address these challenges, we introduce *TuneCLIP*, a novel self-supervised optimization framework designed to enhance state-of-the-art pretrained open-weight CLIP models. **Our contributions** directly address the challenges outlined above:

- 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 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

Contrastive Language-Image Pretraining (CLIP) has emerged as a powerful paradigm for learning joint image-text representations. Following CLIP, several variants have been proposed, including (Zhai et al., 2023; Sun et al., 2023; Yu et al., 2022; Koleilat et al., 2025). CLIP models are trained with image encoders, such as Vision Transformers (ViTs) (Dosovitskiy et al., 2020), ResNets (He et al., 2016), ConvNexts (Woo et al., 2023) and text encoders, including Transformer-based architectures (Vaswani et al., 2017) and BERT (Devlin et al., 2019).

Improving CLIP. Numerous efforts have sought to enhance the efficiency and effectiveness of CLIP pretraining. Several works explore variants of mini-batch contrastive losses to improve representation quality (Li et al., 2023; Chen et al., 2023; Zhai et al., 2023; Shi et al., 2024), while others approximate global contrastive objectives to achieve similar gains (Yuan et al., 2022; Qiu et al., 2023; 2024). In parallel, system-level optimizations focusing on distributed frameworks, memory efficiency, and mixed-precision training have been proposed to further accelerate large-scale CLIP pretraining (Sun et al., 2023; Rasley et al., 2020; Cherti et al., 2023; Wei et al., 2023). While these advances improve the scalability of CLIP training, the high cost of pretraining from scratch continues to motivate methods that adapt and fine-tune existing pretrained CLIPs for downstream tasks.

Improving pretrained CLIP spans several directions, with *supervised fine-tuning* being the most prominent. Numerous studies focus on improving in-distribution retrieval and, by relying on labeled data, are inherently supervision-based (Peleg et al., 2025; Meng et al., 2025; Schall et al., 2024; Mo et al., 2023). Fine-tuning like pretraining has emerged in methods that optimize contrastive objectives with positive and negative pairs defined by labels, for instance by converting class names into textual prompts (Goyal et al., 2023; Wang et al., 2025). The textual prompts for downstream class labels could also be learned to improve the downstream performance (Zhou et al., 2022b;a; Khattak et al., 2023). Such label-dependent adaptation frameworks are designed to fit target domains, which is of no use for the general-purpose robustness (Wortsman et al., 2022; Li et al., 2024). Fine-tuning based on Low-Rank Adaptation (LoRA) (Al Rahhal et al., 2025; Hu et al., 2022) keep the backbone frozen and learn a small low-rank adapter matrix on downstream data, focusing on parameter-efficient adaptation rather than improving the base encoder representations. Others employ curriculum strategies to gradually increase task difficulty (Xiao et al., 2023; Khan et al., 2023). In contrast, our work advances a paradigm of pure self-supervised fine-tuning (SSFT), which uses no labels, pseudo-labels, or teacher models, aiming instead to enhance CLIP's generality while preserving its robust pretrained representations.

Performance degradation in fine-tuning pretrained CLIP is commonly observed. In constrained settings, even modest departures from effective optimization configurations can undermine representation learning and lead to severe degradation (Wortsman et al., 2022; Wei et al., 2023; Mosbach et al., 2020; Wortsman et al., 2023). Furthermore, even when stable training is achieved, multiple studies report a consistent degradation in retrieval performance on other datasets after adaptation on downstream data (Kumar et al., 2022; Peleg et al., 2025; Bafghi et al., 2025), a phenomenon we also observe in our experiments. Consequently, the key problem and the primary motivation for our work is to develop a SSFT strategy that not only avoids these failure modes but also delivers concurrent gains across benchmarks.

It is crucial to distinguish SSFT from *continual learning*. The latter involves training a model on a sequence of tasks over time, with the primary objective of acquiring new knowledge while avoiding catastrophic forgetting of previous tasks Ding et al. (2022); Jha et al. (2024); Xiao et al. (2023); Jiao et al. (2024). In contrast, SSFT aims to improve a pre-trained model through a single adaptation step on a static dataset, enhancing its general capabilities without a sequential task structure.

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 (Radford et al., 2021):

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

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. Cherti et al. (2023) 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 (Yuan et al., 2022) 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:

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

where $\varepsilon > 0$ is a small constant that increases numerical stability. Without explicitly mentioned, we consider $\ell(\cdot) = \cdot$ for GCL as used in (Wei et al., 2024) 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 $\mathcal{L}_{\mathrm{GCL}}(\boldsymbol{\omega})$ is given by

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

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, Yuan et al. (2022) 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:

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

Wei et al. (2024) has designed a distributed optimization framework FastCLIP based on the above algorithm for large-scale CLIP training.

4 TuneCLIP: A Self-supervised optimization framework

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. We will mainly compare with two approaches, OpenCLIP and FastCLIP equipped with an Adam-style optimizer with an initialization $\boldsymbol{\omega}_0$.

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 (Cherti et al., 2023) or FastCLIP (Wei et al., 2024) 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

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 equation 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: $m^\star \leftarrow m_T,\ v^\star \leftarrow v_T,\ u^\star \leftarrow \{u_{i,x}^{(T)}, u_{i,z}^{(T)}\}_{i\in\mathcal{D}}$ `// Transfer to next stage`

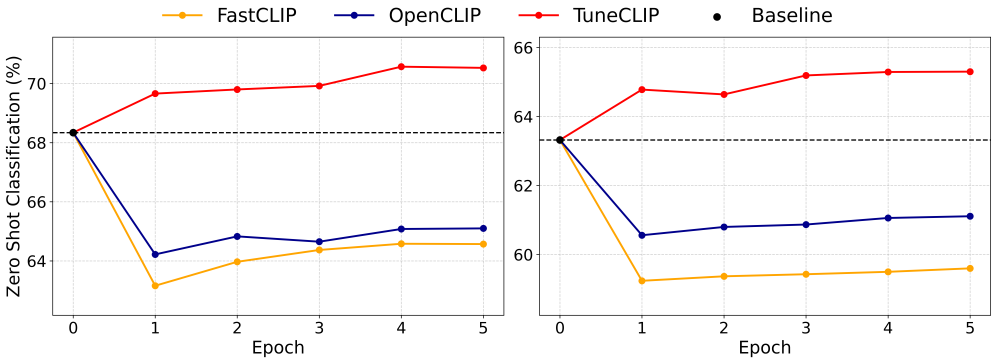


Figure 2: Zero-shot classification (%) performance on ImageNet-1k over 5 fine-tuning epochs for two OpenAI CLIP models (left: ViT-B/16, right: ViT-B/32). While FastCLIP and OpenCLIP show initial degradation and slow recovery, TuneCLIP maintains superior performance throughout fine-tuning.

degradation in the first epoch of fine-tuning, see Figure 2, 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 zeros 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 subsection are deferred to Appendix A.

Theorem 4.1. *Let us consider the updates in (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* $\mathbb{E}[\|\nabla \mathcal{L}_{\mathrm{GCL}}(\boldsymbol{\omega})\|] \le \epsilon$ *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 $\Delta_0 = \mathcal{L}_{\mathrm{GCL}}(\boldsymbol{\omega_0}) - \min_{\boldsymbol{\omega}} \mathcal{L}_{\mathrm{GCL}}(\boldsymbol{\omega})$, $M_0 = \|m_0 - \nabla_\omega \mathcal{L}_{\mathrm{GCL}}(\boldsymbol{\omega_0})\|^2$, $U_{x,0} = \frac{1}{2n}\left\|u_x^{(0)} - \Phi_1(\boldsymbol{\omega_0}, \mathcal{D})\right\|^2$, $U_{z,0} = \frac{1}{2n}\left\|u_x^{(0)} - \Phi_2(\boldsymbol{\omega_0}, \mathcal{D})\right\|^2$.

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

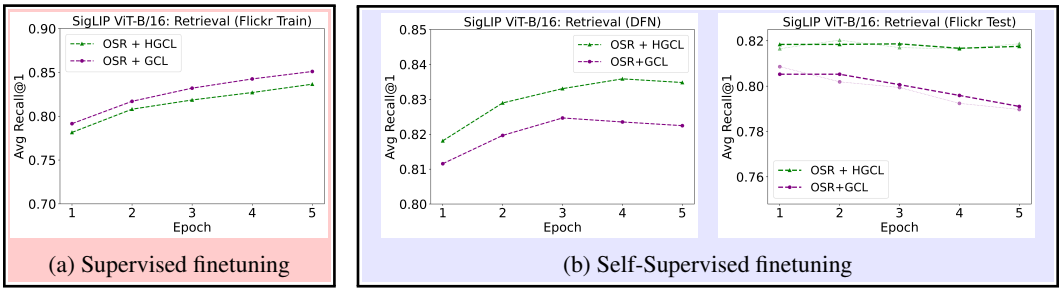


Figure 3: In supervised fine-tuning (red), OSR+GCL outperforms OSR+HGCL (TuneCLIP) because true negative labels justify separating negatives. In contrast, under self-supervised fine-tuning (blue), the absence of such labels makes OSR+HGCL more suitable, leading to improved retrieval performance on Flickr when fine-tuned with SSFT (see Appendix E for OpenAI CLIP & details).

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 to zeros. It is these errors that cause the breaks convergence and hence the performance degradation at the beginning of training. We refer to this issue caused by the initial estimation errors of statistics $\Phi_1(\boldsymbol{\omega_0}, \mathcal{D}), \Phi_2(\boldsymbol{\omega_0}, \mathcal{D}), \nabla_\omega \mathcal{L}_{\rm GCL}$ as cold-start bias.

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

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

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

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.

4.2 Stage II: Hinged Global Contrastive Loss

With accurate initializations of m_0 and $u_{i,x}^{(0)}, u_{i,z}^{(0)}, \forall i$ 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 illustrate a result of the fine-tuning of SigLIP ViT-B/16 on the DFN dataset (see Figure 3b), where the retrieval performance on the fine-tuning dataset keeps increasing but the retrieval performance on testing data such as Flickr decreases. This phenomenon is also prevalent regardless of the pretrained models; see Figure 7 (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 interations. 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 margin hyperparameter constant. It means that as long as $s_{ii} > s_{ij} + m$, its gradient will become

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)
$(m^\star, v^\star, \{u^\star_{i,x}, u^\star_{i,z}\}_{i\in\mathcal{D}}) \leftarrow \text{OSR}(\boldsymbol{\omega}_0, \mathcal{D})$ `// refer 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**
    $\Phi^m_1(\boldsymbol{\omega}, i, \mathcal{B}_t) \leftarrow \frac{1}{|\mathcal{B}_t|}\sum_{z_j\in\mathcal{B}_t\setminus\{z_i\}} \exp\left(\frac{\ell(\mathbf{s}_{i,j}-\mathbf{s}_{i,i})}{\tau}\right)$ `// equation 8`
    $\Phi^m_2(\boldsymbol{\omega}, i, \mathcal{B}_t) \leftarrow \frac{1}{|\mathcal{B}_t|}\sum_{x_j\in\mathcal{B}_t\setminus\{x_i\}} \exp\left(\frac{\ell(\mathbf{s}_{j,i}-\mathbf{s}_{i,i})}{\tau}\right)$ `// equation 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`

zero and hence will not affect the model updates anymore. Illustrative examples of this phenomenon are provided in Table 21 (Appendix K). Accordingly, we define new $\Phi^m_1(.)$ (image-anchored) and $\Phi^m_2(.)$ (text-anchored) as:

$$\Phi^m_1(\boldsymbol{\omega}, i, \mathcal{D}) = \frac{1}{|\mathcal{D}|}\sum_{j\in\mathcal{D}\setminus\{i\}} \exp\left(\frac{\ell(\mathbf{s}_{i,j}-\mathbf{s}_{i,i})}{\tau}\right), \qquad \ell(\mathbf{s}_{i,j}-\mathbf{s}_{i,i}) = [\,\mathbf{s}_{i,j}-\mathbf{s}_{i,i}+m\,]^2_+,$$
$$\Phi^m_2(\boldsymbol{\omega}, i, \mathcal{D}) = \frac{1}{|\mathcal{D}|}\sum_{j\in\mathcal{D}\setminus\{i\}} \exp\left(\frac{\ell(\mathbf{s}_{j,i}-\mathbf{s}_{i,i})}{\tau}\right), \qquad \ell(\mathbf{s}_{j,i}-\mathbf{s}_{i,i}) = [\,\mathbf{s}_{j,i}-\mathbf{s}_{i,i}+m\,]^2_+. \tag{8}$$

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

$$\mathcal{L}_{\text{HGCL}}(\boldsymbol{\omega}) = \frac{\tau}{|\mathcal{D}|}\sum_{i\in\mathcal{D}}\left[\log\big(\varepsilon + \Phi^m_1(\boldsymbol{\omega}, i, \mathcal{D})\big) + \log\big(\varepsilon + \Phi^m_2(\boldsymbol{\omega}, i, \mathcal{D})\big)\right]. \tag{9}$$

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.

Finally, 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

Open-Weight CLIP models. We explore a range of pretrained CLIP models at different scales, including OpenAI's CLIP ViT-B/32, OpenAI's CLIP ViT-B/16, LAION's CLIP ViT-B/32 and SigLIP ViT-B/16, where ViT-B/X refers to the 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 (Fang et al., 2023).

Fine-tuning datasets. To study how performance scales with data under fixed training conditions, we fine-tune on two subsets of the DFN datasets (Fang et al., 2023) containing 12 million (DFN-12M) and 60 million (DFN-60M) samples. DFN datasets are generated by applying *Data Filtering*

Table 1: Summary of mean zero-shot performance across ImageNet variants, retrieval benchmarks, and the DataComp benchmark, together with wall-clock training time (WCT) per GPU. While TuneCLIP delivers consistent improvements across both models, stronger baseline models like SigLIP ViT-B/16 show more modest retrieval gains compared to OpenAI ViT-B/16.

Base Model	Method	WCT. (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 (↓)
	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 (↓)
	TuneCLIP	**9.27**	**65.58** (+2.46)	**69.44** (+0.11)	**63.47** (+1.15)

Networks, which uses a trained model to filter massive uncurated web data into high-quality, task-agnostic corpora. While varying the datasets, the model architecture, optimizer, and schedule are kept fixed.

Training hyperparameters & algorithms. We run OSR for $E = 5$ epochs, and another $E' = 5$ epochs for fine-tuning. Optimizer used is AdamW (Kingma & Ba, 2014) (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 with ablation study on m in Appendix C.1. To ensure consistency and reproducibility, we implement our algorithm using FastCLIP codebase.

Evaluation protocol and metrics. We follow the DataComp protocol (Gadre et al., 2023) and use 38 benchmark datasets. Our main results are reported in three evaluation groups: (1) ImageNet-1k and six robustness variants (Krizhevsky et al., 2017) for assessing zero-shot classification accuracy, (2) MSCOCO or COCO (Vinyals et al., 2016) and Flickr30k (Van Zwol, 2007) for measuring multi as well as single-object retrieval performance, and (3) the full DataComp (Gadre et al., 2023) benchmark. Best model selection is primarily guided by performance on ImageNet-1k.

Main Results. We present results on three evaluation suites for fine-tuning various models on DFN-12M in Table 1. We also plot the curves of zero-shot classification performance on ImageNet-1k during training for different checkpoints of TuneCLIP in Figures 2 and in Figures 9, 10 (Appendix F). Additional detailed results of of zero-shot classification on ImageNet and its variants are shown in Tables 12, 13, and of other tasks are provided in Tables 11, 14, 15. Table 10 summarizes the overall DataComp performance. 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 and FastCLIP not only fail to improve the performance over the base model but also suffer significant performance drop in retrieval and zero-shot classification.

Finally, TuneCLIP for fine-tuning the state-of-the-art model ViT-H/14-quickgelu (Fang et al., 2023) achieves new SOTA accuracy on ImageNet and its variants (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. Compared to the improvements on weaker models, e.g., +1.69% (from 57.67% to 59.36%) on OpenAI ViT-B/16 and +2.46% over SigLIP ViT-B/16 (from 63.12% to 65.58%), the improvement of 1.5% (from 71.80% to 73.23%) is still significant.

Computational Cost & Analysis. We also provide a compute cost analysis for all the algorithms in Appendix H, reporting wall-clock time and GPU-hours across all backbones (Tables 1, 16, 17 & 18). While TuneCLIP incurs higher compute due to its two-stage framework, the overhead remains modest and is consistently accompanied by improved performance across metrics. In contrast, baseline methods cannot even achieve any major improvements even with the same computational costs as ours (refer Figure 10 for extended run of baseline methods).

Table 2: Ablation study on the impact of transferring optimizer statistics from OSR to HGCL fine-tuning using OpenAI ViT-B/16 CLIP. Starting from the baseline without any transferred states, performance is limited across all benchmarks. Introducing (m_t, v_t) transfer yields a substantial jump. Adding u_t on top provides a further boost, resulting in the strongest overall score.

(m_t, v_t) u_t	IN & Variants	Retrieval	DataComp	Mean
$(\times, \times)$ $\times$	54.91	58.64	54.49	56.01
$(\checkmark, \checkmark)$ $\times$	**59.48**	63.70	58.56	60.58
$(\checkmark, \checkmark)$ $\checkmark$	59.36	**64.12**	**58.62**	**60.70** (+4.69)

5.1 Ablation & Scaling 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 recover of first and second-order moments is more important than the recover of u_t (i.e., u_x, u_z). TuneCLIP reaches a +4.7% DataComp gain when all (m_t, v_t, u_t) are used. 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.

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 3 for fine-tuning SigLIP ViT-B/16 and in Figure 7(Appendix E) for fine-tuning OpenAI's CLIP ViT-B/16. The results indicate that for supervised fine-tuning, optimizing GCL with OSR delivers better retrieval performance on the training as well as testing set, while for SSFT, optimizing HGCL with OSR delivers better retrieval performance. This confirms the difference between SSFT and supervised fine-tuning due to the presence of false negatives in SSFT, and corroborates the effectiveness of optimizing HGCL in improving the retrieval performance in case of SSFT. In Figure 11 (Appendix K), we further show that optimizing HGCL achieves smaller variance of similarities scores for false negatives (Top 5 retrieved negative samples). We also observe that the true positive (Top-1) distribution becomes closer to the false negative distribution in the fine-tuning data.

As discussed earlier, standard GCL (without margin-based thresholding) tends to improve classification but simultaneously reduces retrieval scores, with some models such as SigLIP ViT-B/16 and LAION ViT-B/32, falling below their pretrained retrieval baselines due to false-negative over-penalization. As shown in Fig. 4, HGCL mitigates this degradation, preserving classification performance at a comparable level while maintaining retrieval accuracy at or above the original baseline.

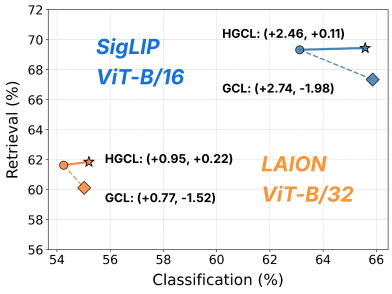


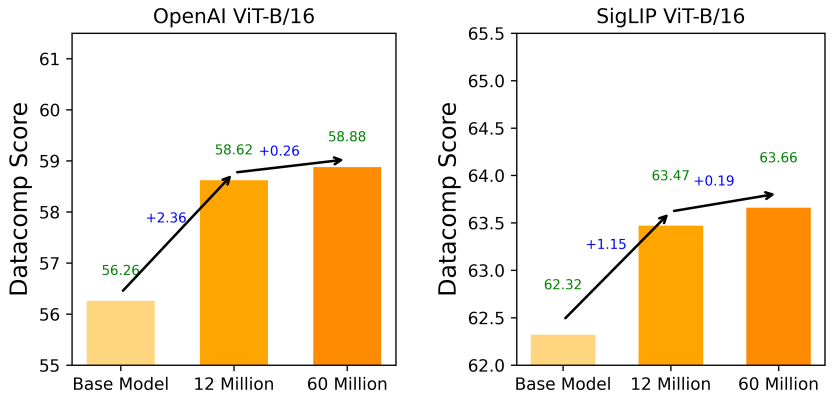
Figure 4: GCL improves classification but can degrades retrieval, whereas HGCL stabilizes retrieval while preserving overall classification gains.

Finally, we analyze how TuneCLIP scales with increasing amounts of fine-tuning data while keeping the model size fixed. As shown in Figure 5, the method maintains stable performance even with a fivefold increase in data on DFN-60M. Although scaling from 12M to 60M samples provides further gains, the improvement is modest because fine-tuning operates on already well-structured pretrained representations. Most generalizable features are retained from pretraining, so additional data primarily reinforces existing alignments rather than discovering new ones.

The DFN datasets are constructed by filtering web image-text pairs using a learned filter, and thus form a relatively clean source of self-supervised training data. To examine how TuneCLIP behaves on noisier corpora, we also fine-tune on the CC12M dataset (Changpinyo et al., 2021), which is a

Table 3: Comparison of TuneCLIP performance with OpenAI CLIP ViT-B/16 when fine-tuned on two different training corpora, the noisier CC12M dataset and the filtered DFN-12M subset.

Method	Data	IN & Variants	Retrieval	DataComp
Base (OpenAI)	×	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)



(a) OpenAI ViT-B/16 (b) SigLIP ViT-B/16

Figure 5: Effect of data scaling on TuneCLIP performance across models.

web corpus with weaker caption-image alignments or less precise texts for an image. As shown in Table 3, TuneCLIP fine-tuned on CC12M still yields a clear improvement and consistent positive gains across the three metrics, while DFN-12M produces even larger average improvements. Overall, these results indicate that TuneCLIP remains effective on both cleaner filtered data and noisier, unfiltered web corpora, rather than relying on any specific property of a dataset.

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, combining optimizer statistics recovery with a hinge-based contrastive loss, provides the mechanism, delivering consistent and dissectible gains across classification, retrieval, and diverse benchmarks. This work thus does more than just propose a new method, it opens a concrete and promising new direction for self-supervised fine-tuning, moving us beyond the limitations of prior art toward truly 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.

References

Mohamad Mahmoud Al Rahhal, Yakoub Bazi, and Mansour Zuair. Lora-clip: Efficient low-rank adaptation of large clip foundation model for scene classification. *Authorea Preprints*, 2025.

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

Reza Akbarian Bafghi, Carden Bagwell, Avinash Ravichandran, Ashish Shrivastava, and Maziar Raissi. Fine tuning without catastrophic forgetting via selective low rank adaptation. *arXiv preprint arXiv:2501.15377*, 2025.

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.

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*, pp. 3558–3568, 2021.

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*, pp. 22648–22657, 2023.

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*, pp. 2818–2829, 2023.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

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.

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.

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.

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.

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

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.

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*, pp. 19338–19347, 2023.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pp. 630–645. Springer, 2016.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

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.

Li Jiao, Lihong Cao, and Tian Wang. Prompt-based continual learning for extending pretrained clip models' knowledge. In *Proceedings of the 6th ACM International Conference on Multimedia in Asia*, pp. 1–8, 2024.

Muhammad Asif Khan, Ridha Hamila, and Hamid Menouar. Clip: Train faster with less data. In *2023 IEEE international conference on big data and smart computing (BigComp)*, pp. 34–39. IEEE, 2023.

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*, pp. 19113–19122, 2023.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Taha Koleilat, Hojat Asgariandehkordi, Hassan Rivaz, and Yiming Xiao. Medclip-samv2: Towards universal text-driven medical image segmentation. *Medical Image Analysis*, pp. 103749, 2025.

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

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.

Kaican Li, Weiyan Xie, Yongxiang Huang, Didan Deng, Lanqing Hong, Zhenguo Li, Ricardo Silva, and Nevin L Zhang. Dual risk minimization: Towards next-level robustness in fine-tuning zero-shot models. *Advances in Neural Information Processing Systems*, 37:66025–66057, 2024.

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*, pp. 23390–23400, 2023.

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)*, pp. 15086–15097, June 2025.

GuangHao Meng, Sunan He, Jinpeng Wang, Tao Dai, Letian Zhang, Jieming Zhu, Qing Li, Gang Wang, Rui Zhang, and Yong Jiang. Evdclip: Improving vision-language retrieval with entity visual descriptions from large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 6126–6134, 2025.

Sangwoo Mo, Minkyu Kim, Kyungmin Lee, and Jinwoo Shin. S-clip: Semi-supervised vision-language learning using few specialist captions. *Advances in Neural Information Processing Systems*, 36:61187–61212, 2023.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. On the stability of fine-tuning bert: Misconceptions, explanations, and strong baselines. *arXiv preprint arXiv:2006.04884*, 2020.

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*, pp. 138–154. Springer, 2024.

Amit Peleg, Naman Deep Singh, and Matthias Hein. Advancing compositional awareness in clip with efficient fine-tuning. *arXiv preprint arXiv:2505.24424*, 2025.

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.

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.

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*, pp. 8748–8763. PmLR, 2021.

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.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 3505–3506, 2020.

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*, pp. 97–110. Springer, 2024.

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.

Liangliang Shi, Jack Fan, and Junchi Yan. Ot-clip: Understanding and generalizing clip via optimal transport. In *Forty-first International Conference on Machine Learning*, 2024.

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*, pp. 5619–5624, 2023.

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.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

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.

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

Ziteng Wang, Siqi Yang, Limeng Qiao, and Lin Ma. Clip-in: Enhancing fine-grained visual understanding in clip via instruction editing data and long captions. *arXiv preprint arXiv:2508.02329*, 2025.

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.

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*, pp. 5439–5449, 2023.

Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. Convnext v2: Co-designing and scaling convnets with masked autoencoders. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16133–16142, 2023.

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*, pp. 7959–7971, 2022.

Mitchell Wortsman, Tim Dettmers, Luke Zettlemoyer, Ari Morcos, Ali Farhadi, and Ludwig Schmidt. Stable and low-precision training for large-scale vision-language models. *Advances in Neural Information Processing Systems*, 36:10271–10298, 2023.

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.

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

Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022.

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*, pp. 25760–25782. PMLR, 2022.

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.

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*, pp. 11975–11986, 2023.

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.

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*, pp. 16816–16825, 2022a.

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, 2022b.

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 Wang & Yang (2022). 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 Wang & Yang (2022) 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

We cite the technical lemma from Wang & Yang (2022) here with slightly changes.

Lemma A.3 (Lemma 8 from Wang & Yang (2022))**.** *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 Wang & Yang (2022) 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 Wang & Yang (2022) 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 Wang & Yang (2022).

A.2 PROOF OF THEOREM

proof of theorem 4.1. The proof is almost the same as theorem 3 in Wang & Yang (2022) so we only make necessary clarifications. Summing equation 10, $\frac{\eta}{\beta}\times$equation 11, and $\frac{20L_f^2C_1^2N\eta}{\gamma B}\times$equation 12

leads to

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

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,

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

discarding the non-dominant terms and unimportant constants, to guarantee $\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\left\|\nabla F(\omega_t)\right\|^2\right] \leq \epsilon^2$ 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 $U_0 = \frac{1}{N}\left\|u_0 - g(\boldsymbol{\omega_0};\mathcal{D})\right\|^2 = \frac{1}{2n}\left\|u_x^{(0)} - \Phi_1(\boldsymbol{\omega_0},\mathcal{D})\right\|^2 + \frac{1}{2n}\left\|u_z^{(0)} - \Phi_2(\boldsymbol{\omega_0},\mathcal{D})\right\|^2$.

□

proof of theorem 4.2. Note that algorithm 1 is essentially SOX Wang & Yang (2022) 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 $M_t = \left\|m_{t+1} - \nabla_\omega\mathcal{L}_{\mathrm{GCL}}(\boldsymbol{\omega_0})\right\|^2$, $U_t = \frac{1}{N}\left\|u_{t+1} - g(\boldsymbol{\omega_0};\mathcal{D})\right\|^2$. Rearranging terms and divide both side for equation 14, equation 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

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

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

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

Convergence of U_t can be easily derived from equation 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

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

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

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

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. Moreover, SigLIP text encoders are configured with `no_causal_mask`, meaning tokens can attend bidirectionally, which differs from the causal masking used in standard CLIP-style transformers. 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. These model configurations are taken from open source implementations of these models.

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
CLIP ViT-B/16	ViT	Transformer	149.62	OpenAI
CLIP ViT-B/32	ViT	Transformer	151.28	LAION
SigLIP ViT-B/16	ViT	Transformer	203.16	Google

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 restricted 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. After experimenting with different values, we found $m = 0.1$ to be a reasonable hyperparameter for most models, and thus adopt it as the default setting. Training was distributed using PyTorch's `DistributedDataParallel` (DDP) to parallelize computation across multiple GPUs and nodes.

In addition to these choices, 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, with γ following a cosine schedule until the 4^{th} epoch and fixed to 0.9 thereafter. We also applied mixed-precision training (*AMP*) to balance performance and efficiency. For margin smoothing, we set the value to 2.0 to stabilize contrastive updates. Each experiment used a world size of 8 for DDP and 6 data-loading workers per GPU to optimize throughput.

C.1 Ablation on margin m used with HGCL

Table 7: Key hyperparameters used for training the models.

Hyperparameter	Value
Image size	224x224 (default)
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

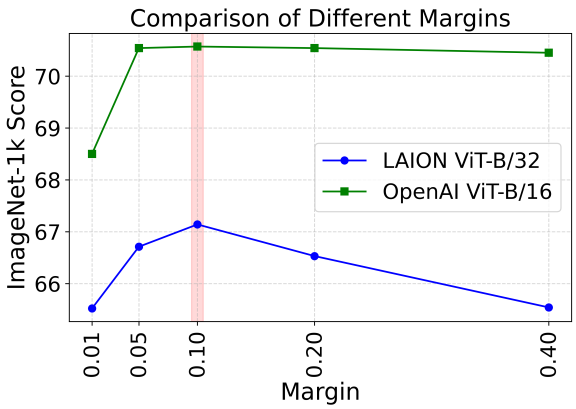


Figure 6: Effect of the HGCL margin hyperparameter m on ImageNet-1k score.

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 6. Using it as a representative metric, we observe that margins around $m = 0.1$ work well for almost all types of models. Based on this trend, we adopt $m = 0.1$ as the margin in all main experiments. Moreover, in the self-supervised setting, there are no class labels or clear ground-truth similarity scores to guide the learning of an adaptive margin. Having the flexibility of keeping truly adaptive margin relies on true reliable positive and negative pairs, which are not available in web-scale datasets. We therefore treat m as a single global hyperparameter, selected using a validation score.

C.2 Ablation on different learning rates

Table 8: Effect of learning rate (`lr`) on ImageNet classification and MSCOCO Retrieval (Average Recall@1) for TuneCLIP with OpenAI ViT-B/16 CLIP.

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 ImageNet-1k and MS COCO retrieval performance around 1e–5 and 5e–6.

C.3 Additional Details on the Distributed Training Framework

We build upon FastCLIP Wei et al. (2024) framework, designed for distributed training and optimized through advanced compositional optimization techniques.

Importantly, all algorithms and proposed variants in this work are implemented within the FastCLIP framework to ensure consistent handling of gradient computation, communication, and optimization dynamics. This allows us to make controlled and fair comparisons, attributing performance differences solely to the algorithmic changes.

D Algorithms compared in the experimentation

Table 9: Training configurations for compared methods.

Method	**Loss**	**Optimization Strategy**
FastCLIP Wei et al. (2024)	GCL	SogCLR + AdamW
OpenCLIP Cherti et al. (2023)	MBCL	AdamW
TuneCLIP (ours)	HGCL (ours)	OSR (ours) + SogCLR + AdamW

Table 9 summarizes the training setups of the algorithms used in our comparison. FastCLIP (Wei et al., 2024) employs the standard Global Contrastive Loss (GCL) with the SogCLR optimization algorithm and AdamW. OpenCLIP (Cherti et al., 2023) relies on a minibatch contrastive loss (MBCL) combined with AdamW. 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 3, 7 and 8 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.

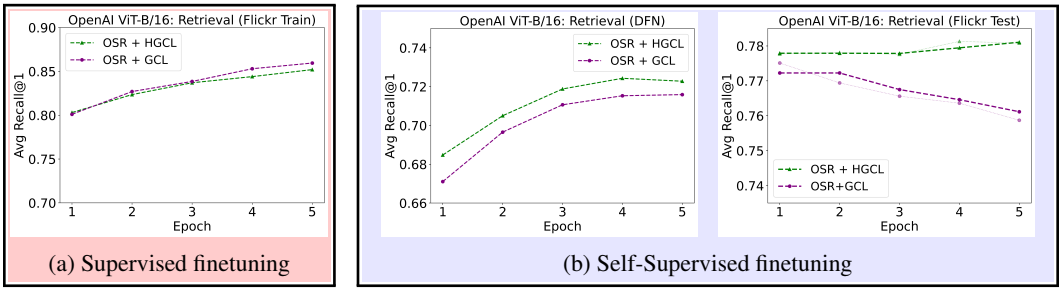


Figure 7: Similar to Figure 3, supervised fine-tuning (red) shows stronger performance with OSR+GCL than with OSR+HGCL (TuneCLIP), since true negative labels justify separating negatives. By contrast, in self-supervised fine-tuning (blue), the absence of such labels makes OSR+HGCL more effective, leading to improved retrieval performance on Flickr when trained with SSFT.

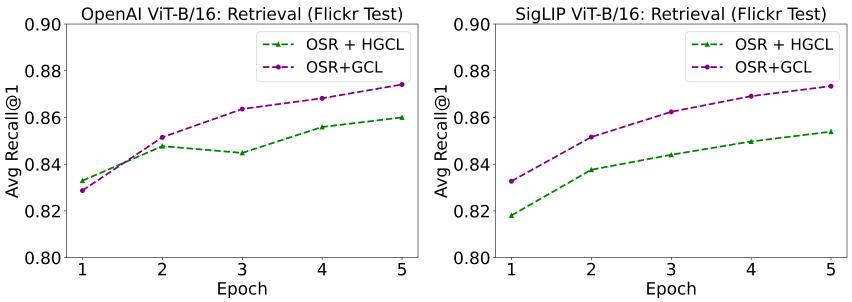
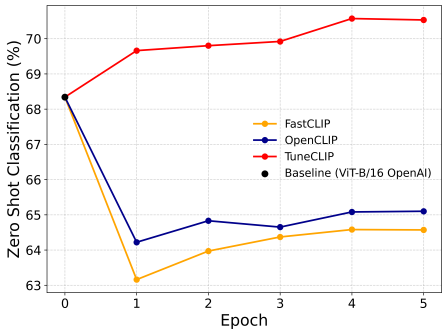


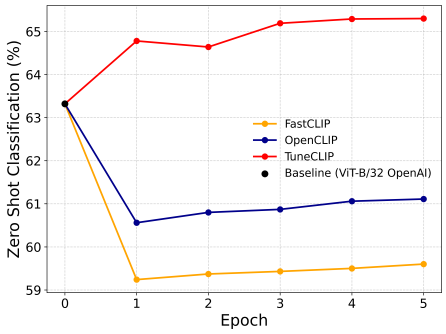
Figure 8: Retrieval trends on the Flickr1k test set under supervised fine-tuning. The left panel shows results for OpenAI ViT-B/16, while the right panel corresponds to SigLIP ViT-B/16. In the supervised setting, explicit labels guide the separation of positives from negatives, making OSR+GCL outperform OSR+HGCL. By contrast, as shown in Figures 3b and 7b, the absence of supervision and margin regularization in SSFT reverses this trend, with OSR+HGCL (TuneCLIP) achieving superior retrieval performance.

F PERFORMANCE TRAJECTORIES DURING FINE-TUNING

Across all four models, we observe that OpenCLIP and FastCLIP exhibit a degraded start and fail to recover within the first few epochs. In contrast, as shown in Figure 9, TuneCLIP consistently outperforms the baseline curves, starting with a boosted score. For LAION ViT-B/32, the initial performance is slightly below the baseline, but by the second epoch it surpasses the baseline, unlike the other two algorithms. This experiment was conducted using ImageNet-1k zero-shot classification accuracy as a representative metric.

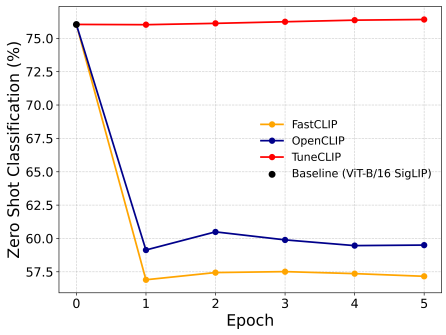


(a) OpenAI ViT-B/16

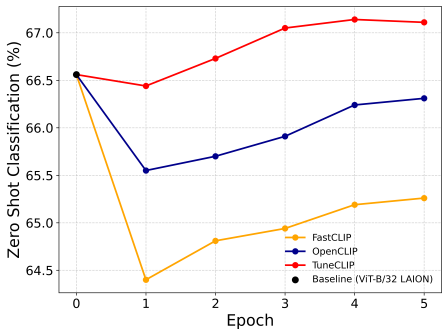


(b) OpenAI ViT-B/32

(c) OpenAI models.



(d) SigLIP ViT-B/16



(e) LAION ViT-B/32

(f) SigLIP and LAION models.

Figure 9: Zero-shot classification performance on ImageNet-1k over 5 epochs of fine-tuning for four ViT models. The dashed line indicates the original pretrained baseline. Across all cases, FastCLIP and OpenCLIP start with degraded performance and recover only gradually, while TuneCLIP consistently achieves higher scores.

F.1 EXTENDED FINE-TUNING

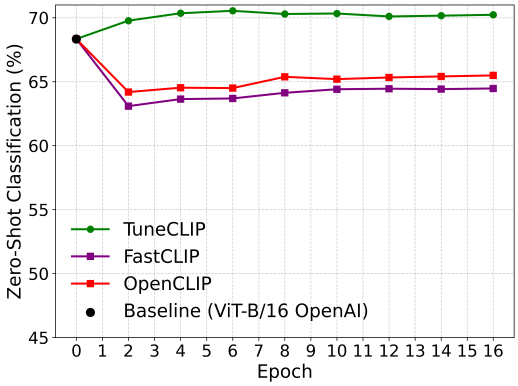


Figure 10: Extended fine-tuning analysis of TuneCLIP and baselines on ImageNet-1k.

As shown in Fig. 10, 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 relative ordering of methods. Importantly, TuneCLIP preserves a consistent improvement over the baselines.

G MORE COMPREHENSIVE RESULTS

We report the gains of TuneCLIP over the baselines, with improvements highlighted in (+). While the majority of important metrics show consistent improvements, a few datasets exhibit small declines (marked in (–)), likely due to task-specific variability. Nevertheless, the overall average performance increases, underscoring the robustness of our approach.

Table 10: Performance of different CLIP models on DataComp Average. DataComp Gadre et al. (2023) 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. In the next sections we show some group-wise results across all variants.

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

Table 11: Performance of different CLIP models on small-scale classification benchmarks. STL-10 is inspired from CIFAR-10, but with higher resolution images.

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
	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
	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
	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
	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, and ImageNet-V2. ImageNet-Sketch is a black-and-white sketch version of 1,000 ImageNet classes collected by Google, while ImageNet-V2 is designed to evaluate robustness under domain shift and avoid adaptive overfitting.

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
	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
	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
	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
	TuneCLIP	**67.14** (+0.58)	**54.46** (+0.81)	**59.10** (+0.95)

Table 13: Performance of different CLIP models on ImageNet out-of-distribution variants Yang et al. (2023) (A, O, R, ObjectNet). ImageNet-A contains adversarially filtered natural images, ImageNet-O includes samples for open-set recognition, ImageNet-R features artistic renditions, and ObjectNet Barbu et al. (2019) 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
	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
	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
	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
	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) Zhai et al. (2019) evaluates performance across 12 diverse tasks. We also report performance on two fairness-oriented datasets, Dollar Street and GeoDE Ramaswamy et al. (2023), 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
	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
	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
	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
	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
	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
	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
	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
	TuneCLIP	**39.55** (+0.21)	**57.80** (+1.48)	**66.82** (+0.04)	83.20 (-0.90)

H COMPUTE COST ANALYSIS

Table 16: 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
	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
	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
	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
	TuneCLIP	**4.32**	**36.24**	**57.22**

Table 17: Batch size specifications of different CLIP variants under our distributed data-parallel training setup. We use `torch.DDP` over $8\times$ GPUs.

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 summarizes the local and global batch sizes used for each backbone under our distributed data-parallel (DDP) setup with 8 GPUs, where the global batch size is given by $B_{\text{global}} = 8 \times B_{\text{local}}$. Across all backbones, Table 16 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 OpenCLIP, FastCLIP, and TuneCLIP. 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 (OpenCLIP/FastCLIP), showing that OSR and HGCL individually are not substantially more expensive than existing baselines.

Table 18: Comparison of fine-tuning cost across four CLIP backbones and four finetuning regimes. Rows show the isolated cost of OSR-only and HGCL-only stages inside 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**

Since all three methods use the same vision, text encoders, same GPUs and training configurations at a given model scale, the FLOPs per training step are effectively the same. Thus, the reported wall-clock time and GPU-hours can be viewed as a direct correlation for the relative total FLOPs across methods.

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 full OSR, indicating that OSR remains the most effective and 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 Bias Mitigation	2nd stage	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**

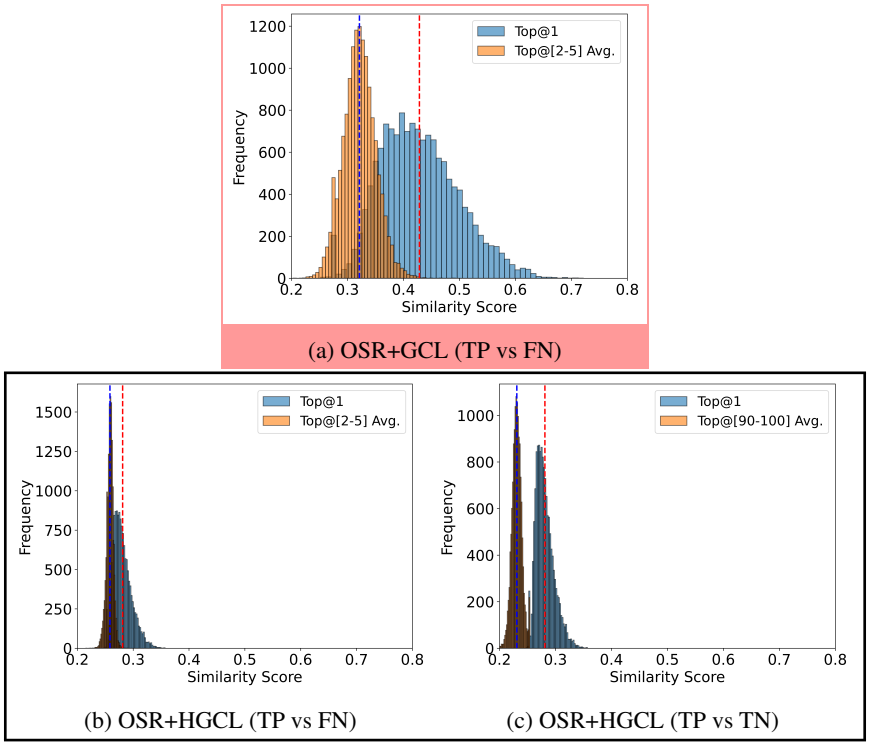
J TuneCLIP on State-of-the-art CLIP

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. While these results do not show dramatic overall gains, they highlight that TuneCLIP scales to very large models and delivers meaningful robustness improvements on ImageNet and variants, which we consider the key takeaway. We report these findings modestly, acknowledging the limited improvements under heavy computational constraints.

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 HGCL and False Negative Mitigations



(a) OSR+GCL (TP vs FN)

(b) OSR+HGCL (TP vs FN)

(c) OSR+HGCL (TP vs TN)

Figure 11: 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 ($\mu = 0.23$).

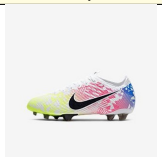
As shown in Figures 11a and 11b, HGCL compresses the gap between true positives and false negatives, producing distributions that are both more compact and overlapping than those under GCL. This analysis, conducted on 15,000 randomly sampled pairs from DFN-12M, highlights HGCL's capacity to reduce variance 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 nega-

tives). 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. Table 21 shows qualitative examples from the above distributions. 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.

L THE USE OF LARGE LANGUAGE MODELS (LLMS)

LLMs were only used to aid or polish writing.

Table 21: 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**