# Getting More Juice Out of Your Data: Hard Pair Refinement Enhances Visual-Language Models Without Extra Data

Anonymous ACL submission

## Abstract

Contrastive Language-Image Pre-training (CLIP) has become the standard for cross-modal image-text representation learning. Improving CLIP typically requires additional data and retraining with new loss functions, but these demands raise resource and time costs, limiting practical use. In this work, we introduce **HELIP**, a cost-effective strategy that improves CLIP models by exploiting challenging text-image pairs within existing datasets in continuous training. This eliminates the need for additional data or extensive retraining. Moreover, HELIP integrates effortlessly into current training pipelines with minimal code modifications, allowing for quick and seamless implementation. On comprehensive benchmarks, HELIP consistently boosts existing models. In particular, within just two epochs of training, it improves zero-shot classification accuracy on ImageNet for SLIP models pre-trained on CC3M, CC12M, and YFCC15M datasets by $3.05\%$, $4.47\%$, and $10.1\%$ , respectively. In addition, on fine-grained classification datasets, HELIP improves the zero-shot performance of CLIP and SLIP by an average of $8.4\%$ and $18.6\%$, and their linear probe performance by an average of $9.5\%$ and $3.0\%$. The code is publicly available at: https://anonymous.4open.science/r/HELIP-NACCL.

## 1 Introduction

Contrastive Language-Image Pretraining (CLIP) (Radford et al., 2021) is quickly becoming the standard for foundation models (Awais et al., 2023) due to its effectiveness for a variety of vision-language tasks without task-specific finetuning (Li et al., 2021; Baldrati et al., 2022). However, web-crawled image-text pairs used for the CLIP model pretraining are often loosely connected, resulting in multiple plausible matches beyond the assigned ones (Wu et al., 2022). Several methods have been presented to improve CLIP models by investigating appropriate matches and utilizing widespread supervision among image-text pairs for training (Li et al., 2022a, 2021; Mu et al., 2022; Radenovic et al., 2023).

Efforts to improve contrastive language-image pretraining models have primarily taken two directions: (1) the addition of objectives to improve the efficacy of supervision (Li et al., 2022a; Mu et al., 2022); and (2) the employment of intra- and inter-modality similarity to select and retrain models using data deemed challenging at the sample level (Li et al., 2021; Radenovic et al., 2023). However, those approaches inevitably require retraining, and those identified as challenging data struggle to bring benefits to model performance. This challenge is partly due to their reliance on finding challenging data within a single batch during training, where truly beneficial challenging data is rare. And, CLIP models' original contrastive loss is not optimally configured to exploit the nuances of difficult data. These limitations restrict the practical application of these methods, especially considering the substantial investments already made in pretraining numerous CLIP models (Li et al., 2022a; Mu et al., 2022); retraining for minimal gains is inefficient. This aspect underscores the need for efficient enhancement strategies that do not rely on additional data collection to improve existing pretrained models.

To improve the existing CLIP models, we introduce the HELIP framework, which involves further training the models with challenging data selected from their original training dataset. HELIP defines and identifies the challenging data at the pair level, distinguishing it from traditional methods that compare sample-level similarities between images and texts. Specifically, HELIP treats each text-image pair as a distinct entity within the joint vision-language space, and defines pairs in close proximity as hard pairs. Furthermore, HELIP intro-

duces the **Hard Pair Mining (HPM)** strategy, a novel approach that moves beyond the traditional use of representation spaces learned by CLIP models. Note, the CLIP space is primarily designed for evaluating sample-level similarities—for instance, comparing an image and text (individually, not as a pair)—lacking in evaluating characteristics at the pair level. HPM transforms the task of discovering pairs in close proximity into a solvable proxy task, with the goal of selecting a pair set that optimally supports the target pair's text-image agreement. HELIP enhances CLIP models not just with the original text-image contrastive loss (Radford et al., 2021), which uniformly pushes all negative samples away from their positive counterpart but also incorporates the **Hard Negative Margin Loss (HNML)** into the loss function. As depicted in Figure 2, HNML imposes an additional geometric structure on the representation space, reflecting the pair-level similarity. Through this approach, HELIP effectively leverages the information within challenging data to boost model performance.

Empirical evidence shows that HELIP improves the performance of existing CLIP models, including pre-trained CLIP, SLIP, and DECLIP, across a variety of benchmarks, such as zero-shot classification, text-image retrieval, and fine-grained linear probing. For zero-shot classification on ImageNet, CIFAR-10, and CIFAR-100, HELIP consistently boosts the performance of all six pre-trained models. Particularly, using HELIP to boost SLIP models pre-trained on CC3M, CC12M, and YFCC15M results in ImageNet zero-shot accuracy gains of $3.05\%$, $4.47\%$, and $10.14\%$, respectively. Further, on seven fine-grained image classification datasets, those pre-trained models achieve better zero-shot and linear probe performance with HELIP. Specifically, the average zero-shot accuracy of CC3M pre-trained CLIP and SLIP are improved by $8.4\%$ and $18.6\%$. The average linear probe accuracy of CC3M pre-trained CLIP and SLIP are improved by $9.5\%$ and $3.0\%$ respectively. Additionally, the performance gain is also valid in terms of zero-shot retrieval, with $1.1$ of R@1 on Flickr30K, and $2.2$ of R@1 on COCO for SLIP with HELIP.

## 2 Related Work

**Vision-Language Pre-training.** Vision-Language Pretraining (VLP) leverages large-scale image-text datasets to learn joint representations transferable to downstream tasks. VLP models are typically classified into single-stream and dual-stream architectures. Single-stream models concatenate text and visual features processed by a single transformer (Li et al., 2019; Chen et al., 2022; Zhang et al., 2020). Dual-stream models use separate encoders for image and text, performing cross-modal interactions at a higher level (Radford et al., 2021; Jia et al., 2021; Li et al., 2022b; Mu et al., 2022; Yao et al., 2022). CLIP (Radford et al., 2021), a dual-stream model, employs contrastive learning with 400M web-crawled image-text pairs to achieve remarkable zero-shot recognition performance. Recent works enhance CLIP's performance by applying self-supervision within the visual modality (Mu et al., 2022) or incorporating nearest neighbor supervision (Li et al., 2022b). While these methods improve performance, they introduce additional computational costs due to data augmentations.

**Contrastive Learning with Hard Negative Samples.** Contrastive learning aims to learn representations by bringing similar examples closer and pushing dissimilar ones apart (Chen et al., 2020a,b; Wang and Isola, 2020). Incorporating hard negative samples into the loss function has been shown to improve performance (Cai et al., 2020; Huynh et al., 2022; Kalantidis et al., 2020; Li et al., 2021; Radenovic et al., 2023; Robinson et al., 2021; Shah et al., 2022). In language-image contrastive learning, approaches like Li et al. (2021) and Radenovic et al. (2023) mine hard negatives using intra- or inter-modality similarity, selecting samples with high cosine similarity in visual or textual features. However, due to the loose alignment in web-crawled data, high similarity in these features doesn't necessarily indicate that pairs are difficult to distinguish. In contrast, we propose a hard sample mining method that discovers similar pairs in the joint vision-language space, efficiently selecting truly challenging samples to improve learning.

## 3 Hard Pairs for Visual-Language Models

In this section, we first define the notations and revisit CLIP for zero-shot recognition in the preliminary section. Next, we introduce the Hard Pairs Mining strategy (HPM), and the associated Hard Negative Margin Loss (HNML), designed to efficiently exploit hard pairs.

### 3.1 Preliminaries

We consider the task of contrastive image-text pretraining. Given an image-caption dataset $\mathcal{D} =$

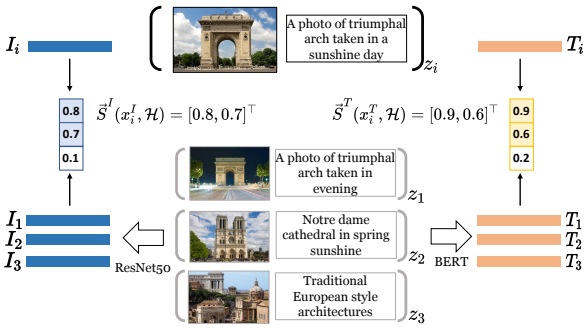



Figure 1: Hard Pair Mining (HPM). Choose hard pairs by optimizing the support set to maximize the agreement prediction of the target pair.

$\{z_i\}_{i=1}^N = \{(x_i^I, x_i^T)\}_{i=1}^N$, $(x_i^I, x_i^T) \in \mathcal{I} \times \mathcal{T}$, the $x_i^I$, $x_i^T$ denote the image and its corresponding caption, $\mathcal{I}$ and $\mathcal{T}$ indicates visual and textual space respectively, and $\mathcal{I} \times \mathcal{T}$ indicates the joint Vision-Language space. The goal is to learn a dual encoder model $\phi = \{\phi_{image}, \phi_{text}\}$, where $\phi_{image}$ represents the image encoder and $\phi_{text}$ denotes the text encoder. We use the shorthand $I_i = \phi_{image}(x_i^I)$ and $T_i = \phi_{text}(x_i^T)$ to denote the encoded representation of an image and its caption, respectively. The contrastive objective of CLIP is formulated as,

$$\ell_{\text{CLIP}} = -\frac{1}{|B|} \sum_{i \in B} \log \frac{\exp\left(sim(I_i, T_i)/\sigma\right)}{\sum_{j \in B} \exp\left(sim(I_i, T_j)/\sigma\right)}, \quad (1)$$

where $sim(\cdot, \cdot)$ is the cosine similarity function, $B$ is a batch of samples and $\sigma$ is a trainable parameter controlling the temperature. Intuitively, the above formulation explicitly aligns the representations of image and text from one pair.

### 3.2 HPM: Hard Pair Mining

In this study, we define *hard pairs* as the pairs that are nearby to a specified target pair within the joint vision-language space, $\mathcal{I} \times \mathcal{T}$, which serves as the domain for pair data. Equation 2 depicts the problem of hard pair mining. Here, $z_i$ represents the target pair, $\mathcal{H}_i$ denotes a set of pairs chosen from the dataset $\mathcal{D}_i = \mathcal{D} \setminus z_i$, and the metric $\mathbf{S}(,)$ quantifies the similarity between the target pair and a set of pairs,

$$\mathcal{H}_i^\star = \arg\max_{\mathcal{H}_i} \mathbf{S}(z_i, \mathcal{H}_i). \quad (2)$$

However, a key challenge arises in defining the similarity metric for pairs, $\mathbf{S}$. Existing CLIP methods (Radford et al., 2021; Li et al., 2022b,a) preliminary focus on aligning an image with its caption (Radford et al., 2021; Li et al., 2022a) from a image-text pair. They rarely emphasize on bringing similar pairs closer while distancing the dissimilar ones, which makes current methods fall short in gauging similarity between two pairs. For instance, the cosine similarity between two pairs is ill-defined, within the context of current methods.

To identify nearby pairs, we introduce the idea of text-image pair agreement maximization. This can be viewed as a proxy task for selecting hard pairs. To illustrate the rationale for using text-image pair agreement as a proxy for selecting hard pairs, we return to the principle obtained from traditional machine learning methods: the prediction of a model on a test sample is substantially influenced by samples in the training dataset that are similar to the test one. For example, the K-Nearest Neighbors (KNN) algorithm classifies a new instance using the K-closest training examples. The linear regression model predicts the output of a test sample using the weighted sum of the training samples, with higher weights given to samples that are more similar to the test sample. Recent empirical and theoretical studies on model memorization and generalization (Chen et al., 2009; Zhang et al., 2021; Stephenson et al., 2021; Brown et al., 2021) also provide support for this. Intuitively, if a pair agreement prediction model trained on a set of pairs predicts a specific target pair as having a high probability of being a matching pair, the target pair is likely to be similar to the matching pairs on which the model was trained. The challenge of selecting hard pairs is transformed into an optimization task centered on the text-image pair agreement, which is formally represented as:

$$\arg\max_{\mathcal{H}_i} \mathbf{S}(z_i, \mathcal{H}_i) = \arg\max_{\mathcal{H}_i} P_{\mathcal{M}}(z_i | \mathcal{H}_i), \quad (3)$$

where $P_{\mathcal{M}}(z_i|\mathcal{H}_i)$ denotes the prediction of a pair agreement model, $\mathcal{M}$, for the pair $z_i$ based on a pair set $\mathcal{H}_i$. This set is a subset of $\mathcal{D}_i$. In this framework, the goal of selecting a hard pair is transformed into identifying a training set $\mathcal{H}_i$ such that the model $\mathcal{M}$ predicts the target pair as a matching pair.

Designing a suitable pair agreement prediction model for this proxy task is a nontrivial endeavor because the model needs to not only predict the pair matching probability but also allow the optimization of the training set, as indicated in Equation 3. Consequently, a conventional deep neural network design becomes unviable due to the impracticality of retraining across all possible sets $\mathcal{H}_i$ from $\mathcal{D}_i$. Taking inspiration from recent work (Norelli et al., 2022), we propose a data-centric design for the agreement prediction model $\mathcal{M}$. As illustrated in

Figure 1, the model leverages two pretrained single-modal encoders, i.e., $f_{\text{image}}$ and $f_{\text{text}}$, to align representations of images and texts in a unified Vision-Language space. Specifically, the model encodes the target pair $z_i$ into $(I_i, T_i)$ using these single-modal encoders. For the visual modality, we determine a similarity vector between the target pair $z_i$ and the dataset $\mathcal{D}_i$. The similarity vector is defined as $\vec{S}^I(x_i^I, \mathcal{D}_i) = [\ldots, sim(I_i, I_j), \ldots]^\top \in \mathbb{R}^{N-1}$. Here $I_j = f_{\text{image}}(x_j^I)$ with $(x_j^I, x_j^T)$ being an element of $\mathcal{D}_i$, and function $sim(\cdot,\cdot)$ denotes the cosine similarity. To counteract noise, values in the vector $\vec{S}^I(x_i^I, \mathcal{D}_i)$ are set to zero if $sim(I_i, I_j) < \tau$. This cleaned-up vector is represented as $\widetilde{S}^I$. The procedure for the textual modality is analogous, producing a vector denoted as $\widetilde{S}^T$. Note, the representations in this shared space are intuitively interpretable: each dimension corresponding to the visual/textual similarity of the input to a unique pair in the multimodal dataset. This interpretable characteristic enables us to directly optimize the supporting set to maximize the pair matching probability:

$$\mathcal{H}_i^\star = \underset{|\mathcal{H}_i|=k}{\arg\max}\, \widetilde{S}^I(x_i^I, \mathcal{H}_i)^\top \widetilde{S}^T(x_i^T, \mathcal{H}_i), \quad (4)$$

where the $\mathcal{H}_i^\star$ is the hard pair set and $k \in \mathbb{R}^+$ is the number of selected pairs which is much less than $|\mathcal{D}|$. The previous problem can be efficiently solved by greedily choosing dimensions that maximize the inner product. Due to the interpretable property, the selected dimensions are corresponding to the desired pairs.

**Mitigation of Noisy Data Impact.** The prior method assumes the target pair $z_i$ to be a suitable matching pair. However, in inherently noisy datasets, such as web-crawled ones like LAION (Schuhmann et al., 2022), mismatched pairs might be present. The potential negative effects of hard pairs generated by these mismatched pairs necessitate the development of a strategy for identifying and eliminating them. We create a pair removal strategy based on the availability of hard pairs: A target pair $z_i$ is deemed as unsuitable and thus removed, if there is a non-empty subset of the mined hard pair set, $\mathcal{H}_i^{sub} \subseteq \mathcal{H}_i^\star$ with $|\mathcal{H}_i^{sub}| > 0$, such that $\widetilde{S}^I(x_i^I, \mathcal{H}_i^{sub})^\top \widetilde{S}^T(x_i^T, \mathcal{H}_i^{sub}) = 0$.

Intuitively, this equation suggests that the number of entries positively supporting the target pair $z_i$ as a matching pair is fewer than $k$. To illustrate how this concept can aid in cleaning noisy data, consider the following example: Suppose the target pair consists of a "cat" image but a "dog" caption (clearly it is a mismatch). For it to be considered a correct match, numerous pairings with same erroneous pattern (i.e., "cat" images paired with "dog" captions) would need to exist in the dataset. By assuming a certain error types are fewer than $k$ throughout the dataset, if no subset of size $k$ within the dataset $\mathcal{D} \setminus z_i$ supports $z_i$ as a matching pair, this signals that the target pair is an outlier, likely due to a labeling error or mismatch. Such outliers can degrade dataset quality, so they are removed to ensure the reliability of hard data.

**Fast Hard Pair Mining (FastHPM).** It is intuitive to infer that for a dataset collected from a single source, the number of intrinsic hard pairs, which are robust enough to enhance the learned representation, will proportionally increase with the size of the dataset originating from that source. To identify $k$ (much less than $|\mathcal{D}|$) qualified hard pairs, a portion of the dataset $\mathcal{D}$ is sufficient. As a result, we present the Fast Hard Pair Mining (FastHPM) approach, which was designed to avoid the time complexity associated with hard pair mining over the entire dataset. FastHPM's objective can be formalized as follows:

$$\mathcal{H}_i^\star \approx \underset{|\mathcal{H}|=k}{\arg\max}\, \widetilde{S}^I(x_i^I, \mathcal{H}_i)^\top \widetilde{S}^T(x_i^T, \mathcal{H}_i), \quad (5)$$

where $\mathcal{H}_i \subseteq \overline{\mathcal{D}}_i$ and $|\overline{\mathcal{D}}_i| = C$ is sampled uniformly from set $\mathcal{D}_i$. In this equation, it's noteworthy that the selection of value $C$ is solely based on the number of hard pairs $k$, instead of the size of $\mathcal{D}_i$. Consequently, this optimization reduces the time complexity of FastHPM to $\mathcal{O}(N)$. The detailed procedure of the hard pair mining algorithm is presented in Appendix A.

### 3.3 HNML: Hard Negative Margin Loss

The image-text contrastive loss $\ell_{CLIP}$, as illustrated in the preliminary section, aligns the true image-text pairs. But it poses no constraints on the overall geometry among data pairs (Goel et al., 2022). After involving hard data into the finetuning stage, equally maximizing the distance for normal negative pairs and hard negative pairs is an undesired way to utilize the information provided by hard negative pairs. The intuition follows directly from Figure 2. In a desired representation space, the similarity between the positive and the hard negative, $\mathbf{S}_1$, should be greater than the similarity between the positive and those normal negatives, $\mathbf{S}_2, \mathbf{S}_3$. Therefore, to impose the additional geo-

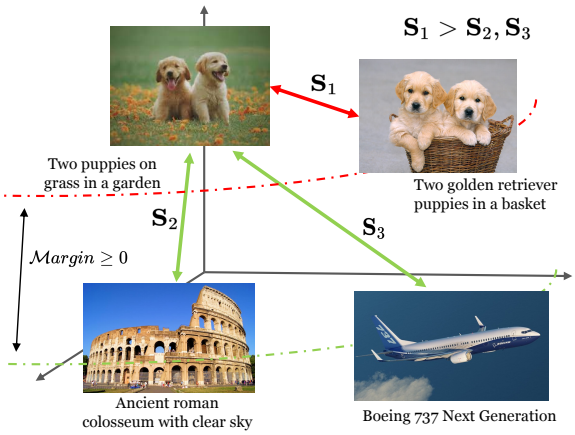



Figure 2: Hard Negative Margin Loss (HNML). Hard negative pairs are closer to the positive than the normal negative pairs.

metric structure, we introduce the Hard Negative Margin Loss (HNML):

$$\ell_{\text{margin}} = \frac{1}{|B|} \sum_{j \in B} \max\left(0, sim(I_i, T_j) - \min_{j' \in \mathcal{H}_i^p} \{sim(I_i, T_{j'})\}\right), \tag{6}$$

where $\mathcal{H}_i^p \subseteq \mathcal{H}_i^\star$ is the hard negative pairs for the target $z_i$ involved in one training batch. Note, the HNML is computationally efficient. No extra inner product computation is required. The geometric regularization is applied over the inner product matrix computed in the original CLIP loss, Equation equation 1. Then, the well-trained model is finetuned with the following loss, where $\gamma$ is the hyperparameter balancing the two losses,

$$\ell_{\text{finetune}} = \ell_{\text{CLIP}} + \gamma \ell_{\text{margin}}. \tag{7}$$

To boost the performance of well-trained CLIP models without introducing extra data and extra parameters, we introduce the further training strategy which involves the preprocessed hard pairs into the batch composition during training. As shown in Figure 3, for text-image pairs within the batch $B$, we randomly sample a subset $B'$ as seeds. Then, for $z_i \in B'$, we randomly select $|\mathcal{H}_i^p| = p$ pairs from $\mathcal{H}_i^\star$. The actual training batch is $\overline{B} = B \bigcup_{i=0}^{|B'|} \mathcal{H}_i^p$. We summarize the training pipeline in appendix A.

# 4 Experiments

In Section 4.2, we empirically investigate HELIP's efficacy in improving zero-shot classification, image-text retrieval, and linear probing performance of existing vision-language models. Section 4.3 examines HELIP's performance with scaled training data. We provide in-depth studies on Hard

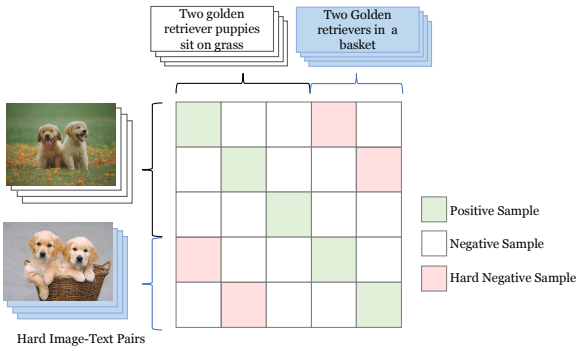



Figure 3: Continuous training CLIP with Hard Pairs. For text-image pairs within a batch, we sample corresponding hard data from the preprocess hard pair set.

Positive Mining (HPM) and Hard Negative Mining with Margin Loss (HNML) in Sections 4.4 and 4.5, respectively. Discussions on robustness over noisy datasets and additional empirical analyses of hard mining methods are deferred to the appendix.

## 4.1 Experimental Setup

**Training Datasets.** We used open-source datasets including CC3M (Sharma et al., 2018), CC12M (Changpinyo et al., 2021), and two 15M subsets of YFCC100M: v1 (Radford et al., 2021) and v2 (Li et al., 2022b). The combined datasets—CC3M, CC12M, and YFCC15M v1 (Open29M (Li et al., 2022b))—were not fully obtained due to expired URLs. Additionally, we sampled 7.5M and 8M subsets from the noisier LAION-5B (Schuhmann et al., 2022), labeled LAION7.5M and LAION8M. Although smaller than the 400M pairs used in CLIP's original study (Radford et al., 2021), these datasets suit our computational resources and have been widely used in language-image pretraining benchmarks (Goel et al., 2022; Li et al., 2022b; Mu et al., 2022).

**Downstream Datasets.** We evaluate HELIP using zero-shot image classification, linear probing, and zero-shot image-text retrieval. Beyond ImageNet (Deng et al., 2009), CIFAR10, and CIFAR100 (Krizhevsky et al., 2009), we assess performance on seven fine-grained classification datasets: Caltech101 (Fei-Fei et al., 2004), Food101 (Bossard et al., 2014), Sun397 (Xiao et al., 2010), Flowers102 (Nilsback and Zisserman, 2008), CUB (Wah et al., 2011), Stanford Cars (Krause et al., 2013), and FGVC Aircraft (Maji et al., 2013). For zero-shot image-text retrieval, we use MS-COCO (Lin et al., 2014) and Flickr30K (Plummer et al., 2015). Implementation details are provided in the appendix.

### 4.2 Main Results and Discussion

**Zero-Shot Classification.** We evaluate the zero-shot performance of CLIP, SLIP, and DECLIP models, along with their HELIP-fine-tuned versions (CLIP-HELIP, SLIP-HELIP, and DECLIP-HELIP), on the CC3M, CC12M, YFCC15M, and Open29M datasets. Table 1 shows that models with HELIP consistently outperform their counterparts. Specifically, on the CC3M dataset, HELIP boosts the ImageNet zero-shot accuracy of CLIP from 19.04% to 19.86% and improves SLIP by over 13% to 26.05%. We include two baseline methods, CYCLIP (Goel et al., 2022) and CLOOB (Fürst et al., 2021), for reference. Using SLIP checkpoints from Mu et al. (2022) pretrained on CC12M, SLIP-HELIP achieves a 4.47% higher zero-shot accuracy on ImageNet than SLIP. Since DECLIP parameters for CC3M and CC12M are unavailable, we compare DECLIP and DECLIP-HELIP on the YFCC15M v2 dataset, also presenting SLIP and DECLIP models pretrained by Li et al. (2022b) evaluated with their pipeline (denoted with $*$). Because templates significantly impact zero-shot tasks, we use our evaluation pipeline (the same as OpenCLIP) for fair comparison; further baseline details are in Appendix B. Both SLIP and DECLIP show improvements with HELIP, averaging increases of 15.49% and 6.74%, respectively. To demonstrate HELIP's efficacy on larger datasets, we evaluated CLIP and CLIP-HELIP on Open29M. The original CLIP model reaches its best zero-shot ImageNet accuracy of 42.32% at epoch 18. Applying HELIP boosts this to 46.33% with just *one additional epoch*, whereas continued training with the original CLIP loss slightly decreases accuracy to 42.25%.

**Zero-Shot Fine-Grained Classification.** Utilizing hard image-text pairs in contrastive learning, HELIP enhances the discriminative power of CLIP's visual embeddings, benefiting fine-grained classification tasks. As shown in Table 2, SLIP-HELIP improves zero-shot accuracy on Caltech101 by 12.88% and 3.95% for models pre-trained on CC3M and CC12M, respectively. Both CLIP and SLIP models consistently improve when augmented with HELIP.

**Zero-Shot Retrieval.** We evaluate HELIP on zero-shot image-to-text retrieval tasks on MS-COCO (Lin et al., 2014) and Flickr30K (Plummer et al., 2015). As shown Table 3, both CLIP and SLIP, pre-trained on CC3M and CC12M , consistently improved by HELIP.

| | Method | ImageNet | CIFAR10 | CIFAR100 |
|---|---|---|---|---|
| CC3M | CYCLIP (Goel et al., 2022) | 22.08 | 51.45 | 23.15 |
| | CLOOB (?) | 23.97 | - | - |
| | CLIP† (Radford et al., 2021) | 19.04 | 33.06 | 13.77 |
| | CLIP†-HELIP | 19.86 | 34.05 | 14.13 |
| | SLIP (Mu et al., 2022) | 23.00 | 65.61 | 34.69 |
| | SLIP-HELIP | **26.05** | **68.18** | **37.77** |
| CC12M | CLIP† (Radford et al., 2021) | 30.27 | 51.07 | 21.94 |
| | CLIP†-HELIP | 32.05 | 52.27 | 24.51 |
| | SLIP (Mu et al., 2022) | 41.17 | 81.30 | 53.68 |
| | SLIP-HELIP | **45.64** | **82.31** | **53.79** |
| YFCC15M | SLIP (Mu et al., 2022) | 25.29 (34.30*) | 60.19 | 26.80 |
| | SLIP-HELIP | 35.43 | 75.49 | 47.84 |
| | DECLIP (Li et al., 2022b) | 36.05 (43.20*) | 78.12 | 50.60 |
| | DECLIP-HELIP | **43.80** | **84.88** | **56.31** |
| 29M | CLIP† (Radford et al., 2021) | 42.32 | 71.98 | 42.73 |
| | CLIP† Cont. Train | 42.25 | 71.72 | 42.66 |
| | CLIP†-HELIP | **46.33** | **77.97** | **48.33** |

Table 1: Zero-shot classification performance on ImageNet, CIFAR-10, and CIFAR-100. Baselines marked with † were trained by us; others use publicly available pre-trained parameters. For SLIP and DECLIP on YFCC15M, we report both our evaluation using OpenCLIP with pre-trained parameters from (Li et al., 2022b) and the results reported in (Li et al., 2022b), marked with $*$.

**Linear Probing.** The linear probing task trains a randomly initialized linear classifier on the feature extracted from the frozen image encoder on the downstream dataset. We train the logistic regression classifier using scikit-learn's L-BFGS implementation (Pedregosa et al., 2011), with maximum 1,000 iterations on those 7 datasets. For each dataset, we search for the best regularization strength factor on the validation set over 45 logarithmically spaced steps within the range 1e-6 to 1e+5. Experimental results in Table 4 demonstrate that both CLIP-HELIP and SLIP-HELIP have consistent improvements over their counterparts on almost all 7 datasets. Note that on CC12M, SLIP-HELIP performs marginally better on 5 out of 7 datasets. It's probably because the self-supervision of SLIP (Mu et al., 2022) within the visual modal can be beneficial for learning fine-grained visual embedding, while SLIP-HELIP doesn't include image self-supervision during the training. And we did not match the training batch size as SLIP (Mu et al., 2022) because of resource limitations. A combination of HELIP and image self-supervision with larger batch size may be a potential direction for achieving better linear probe performance.

### 4.3 HELIP with Scaled Training Data

To investigate the impact of expanded training dataset sizes on the effectiveness of HELIP, we

| Dataset | Method | Caltech101 | Food101 | Sun397 | Flowers102 | CUB | Stanford Cars | FGVC Aircraft | Average |
|---|---|---|---|---|---|---|---|---|---|
| CC3M | CLIP | 42.14 | 13.02 | 27.08 | 13.37 | 3.45 | 1.08 | 1.02 | 14.45 |
| | CLIP-HELIP | 48.08 | 13.11 | 28.94 | 13.61 | 3.70 | 1.17 | 1.11 | 15.67 |
| | SLIP | 54.01 | 16.03 | 29.19 | 12.06 | 4.70 | **1.21** | **1.50** | 16.96 |
| | SLIP-HELIP | **66.89** | **17.05** | **33.69** | **15.16** | **4.85** | 1.19 | 1.29 | **20.12** |
| CC12M | CLIP | 63.78 | 31.53 | 37.86 | 19.56 | 7.32 | 14.22 | 2.49 | 25.25 |
| | CLIP-HELIP | 64.85 | 36.49 | 38.22 | 24.73 | 8.58 | 15.59 | 2.97 | 27.35 |
| | SLIP | 76.33 | 52.33 | 44.96 | **31.81** | 10.50 | 22.53 | 3.06 | 34.50 |
| | SLIP-HELIP | **80.28** | **54.86** | **47.53** | 31.39 | **10.56** | **25.67** | **4.08** | **36.34** |

Table 2: Zero-shot performance on fine-grained image classification. On a variety of fine-grained classification benchmarks, HELIP consistent boosts the model performance compared to the original versions.

| Pretraining Dataset | Method | COCO | | Flickr30K | |
|---|---|---|---|---|---|
| | | R@1 ↑ | R@5 ↑ | R@1 ↑ | R@5 ↑ |
| CC3M | CLIP | 14.4 | 34.1 | 31.7 | 56.0 |
| | CLIP-HELIP | 17.8 | 39.8 | 35.4 | 61.0 |
| | SLIP | 22.3 | 45.6 | 39.6 | 68.6 |
| | SLIP-HELIP | **23.4** | **48.3** | **41.8** | **69.6** |
| CC12M | CLIP | 26.9 | 52.6 | 47.2 | 74.3 |
| | CLIP-HELIP | 27.8 | 54.3 | 48.2 | 75.4 |
| | SLIP | 39.0 | 66.0 | 65.4 | **90.1** |
| | SLIP-HELIP | **39.4** | **67.2** | **66.2** | 89.7 |

Table 3: Zero-shot image-text retrieval results on MSCOCO and Flickr. ↑ indicates higher is better.

trained the CLIP model on the YFCC15M dataset. This training yielded a zero-shot classification accuracy of 25.46% on ImageNet. After applying HELIPand one epoch of training, its performance improved to 26.45%. To summarize the zero-shot performance on ImageNet of both the standard CLIP and its enhanced version, CLIP-HELIP, across different data scales, we have illustrated these results in Figure 4. The results show that HELIP consistently enhances CLIP's performance. Most notably, the largest dataset, Open29M, witnessed a remarkable performance increase of 3.06% with HELIP. This result indicates that HELIP can provide immediate performance enhancements for well-trained CLIP models on larger datasets, such as the private 400M dataset mentioned in (Radford et al., 2021).

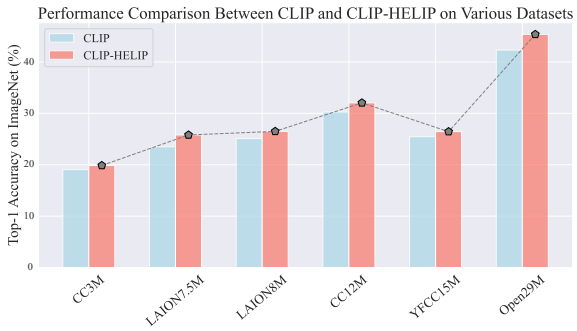



Figure 4: Zero-shot performance on ImageNet for models pre-trained on different dataset sizes.

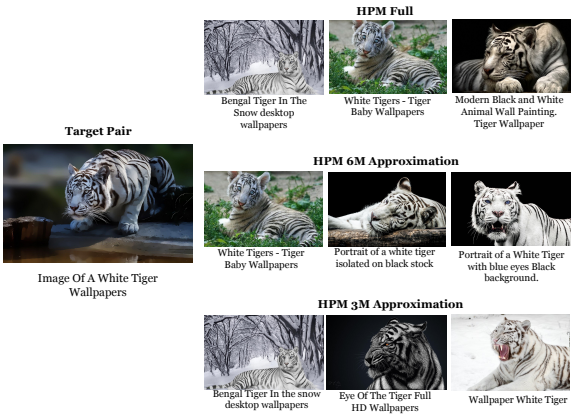



Figure 5: Hard pairs from HPM and fastHPM. FastHPM produces high-quality hard pairs that compete with HPM.

### 4.4 Impact of Hard Negative Margin Loss

We examine the impact of hard negative margin loss (HNML) on the SLIP model's performance. Specifically, we analyze the SLIP model pre-trained on CC3M when further trained with both HPM and HNML, compared to training without HNML. We evaluated zero-shot classification performance across datasets including ImageNet, CIFAR-100, CIFAR-10, Caltech-101, Food-101, and SUN397. As detailed in Table 5, the SLIP model augmented with HPM and HNML outperforms the baseline SLIP and SLIP + HPM models by 4.51 and 3.27 points, respectively. Notably, the model achieves better performance on CIFAR-10 without HNML. We hypothesize that HNML enhances the discriminative power of learned representations by incorporating class distance as a cost metric. Therefore, for classification tasks with a larger number of subclasses, employing HNML during training can improve performance.

### 4.5 Delving into Hard Pair Mining

**Performance Comparison between HPM and FastHPM.** A comparison was made between the zero-shot performances of SLIP models, further

| Dataset | Method | Caltech101 | Food101 | Sun397 | Flowers102 | CUB | Stanford Cars | FGVC Aircraft | Avg. |
|---|---|---|---|---|---|---|---|---|---|
| CC3M | CYCLIP | 80.88 | 54.95 | - | 83.74 | - | 22.72 | 28.02 | - |
| | CLIP | 80.11 | 53.82 | 56.40 | 84.07 | 40.30 | 22.70 | 35.61 | 53.29 |
| | CLIP-HELIP | 82.49 | 59.79 | 59.56 | 87.84 | 46.19 | 30.01 | 42.48 | 58.34 |
| | SLIP | 87.96 | 72.50 | 66.96 | 91.91 | 49.77 | 39.25 | 45.87 | 64.89 |
| | SLIP-HELIP | **89.64** | **73.09** | **67.67** | **93.02** | **53.16** | **42.44** | **48.66** | **66.81** |
| CC12M | CLIP | 85.35 | 68.00 | 64.45 | 87.88 | 48.75 | 57.80 | 40.32 | 64.65 |
| | CLIP-HELIP | 85.87 | 68.89 | 64.95 | 88.36 | 49.41 | 58.55 | 40.17 | 65.17 |
| | SLIP | **92.89** | 83.63 | 74.34 | 94.87 | **60.99** | 73.43 | 52.23 | 76.05 |
| | SLIP-HELIP | 92.85 | **84.25** | **74.74** | **95.09** | 60.53 | **74.23** | **52.36** | **76.29** |

Table 4: Linear probe performance on Fine-grained Image Classification. On average, the linear probe performance of CLIP and SLIP pretrained on CC3M and CC12M are improved.

| | ImageNet | CF10 | CF100 | Caltech101 | Food101 | Sun397 | Avg. |
|---|---|---|---|---|---|---|---|
| SLIP | 23.00 | 65.61 | 34.69 | 54.01 | 16.03 | 29.20 | 37.09 |
| wo HNML | 24.94 | **69.44** | 36.35 | 64.07 | 16.51 | 30.91 | 40.37 |
| w HNML | **26.05** | 68.18 | **37.77** | **66.89** | **17.05** | **33.68** | **41.60** |

Table 5: SLIP continous training with and without hard negative margin loss.

trained with hard pairs obtained from both HPM and fastHPM. This comparison, conducted under three different settings, was summarized in Table 6. Additionally, we established subsets $\widetilde{\mathcal{D}}_i$ of sizes 3M and 6M, and accordingly denoted HELIP with these subset sizes as HELIP-3M and HELIP-6M. Table 6 shows that the zero-shot performances of HELIP-3M and HELIP-6M remain competitive with the global HPM hard pair mining approach. These findings suggest that fastHPM offers an efficient hard pair mining strategy without compromising performance and has the potential to scale up in larger pre-training datasets.

| | Imagenet | CIFAR10 | CIFAR100 |
|---|---|---|---|
| SLIP | 41.17 | 81.30 | 53.68 |
| HELIP- 3M | 45.07 | **82.42** | 55.22 |
| HELIP- 6M | 44.98 | 81.64 | **56.62** |
| HELIP- Full | **45.64** | 82.31 | 53.79 |

Table 6: Zero-shot performance of SLIP with HELIP on CC12M with hard pairs from HPM and fastHPM.

**Visual insights into HPM and FastHPM.** We took the initiative to visualize the hard pairs as identified by the aforementioned three methods. Within Figure 5, the leftmost image-text pairing is earmarked as the target. The pairs in the primary row are those selected via HPM. The subsequent rows, specifically the second and third, present image-text pairings identified by the 6M fastHPM and the 3M fastHPM methods, respectively. Through a comparative visualization, it's evident that the hard pairs pinpointed by fastHPM bear a significant resemblance to the target pair. For readers keen on delving deeper, we've provided an extended set of visualization outcomes in Appendix J.

**Computational Time Analysis.** Table 7 provides a comparison of the computational time required by HPM and fastHPM. The hard negative pairs preparation times listed were measured on 8 V100 GPUs, with the exception of the $*$ symbol, which was measured on a single V100 GPU. Given its efficiency and the performance similarities observed in Table 6, fastHPM emerges as a compelling alternative to the full HPM method.

| | CC3M | CC12M | YFCC15M |
|---|---|---|---|
| HELIP- 3M | - | 2h18min | 3h27min |
| HELIP- 6M | - | 5h3min | 6h19min |
| HELIP- Full | 1h9min* | 9h11min | 17h41min |

Table 7: Preparation time for hard pairs. FastHPM speeds up the hard negative pairs mining process.

## 5 Conclusion

In this work, we present HELIP, a framework that enhances pre-trained CLIP models by more effectively utilizing their original training datasets. HELIPoffers a cost-effective and easily integrable solution for improving existing models without extensive retraining or additional data. Specifically, it treats each text-image pair as a point in the joint vision-language space and identifies hard pairs, those that are close together, using the Hard Pair Mining (HPM) strategy. Furthermore, to efficiently leverage these hard pairs, we introduce the Hard Negative Margin Loss (HNML). Empirically, we found that HELIPboosts the performance of existing checkpoints within a few epochs of continuous training. Evaluations across various benchmarks, including zero-shot classification, image-text retrieval, and linear probing, demonstrate the effectiveness and efficiency of our method. These findings highlight the potential of existing datasets and encourage future research to intelligently maximize their utilization.

## 6 Limitations

While HELIP demonstrates significant improvements over existing CLIP models, several limitations should be acknowledged.

**Dependence on Dataset Quality and Diversity.** The effectiveness of HELIP is inherently tied to the quality and diversity of the original training dataset. If the dataset contains limited variability or is biased toward certain domains, the benefits of mining hard pairs may diminish. In scenarios where datasets are very small, the selection of meaningful challenging pairs becomes more difficult, potentially limiting the overall performance gains.

**Reusing Original Training Data.** HELIP leverages the original pretraining data without incorporating additional external data sources. While this approach reduces the need for costly data collection, it might also inherits biases present in the original dataset.

**Limited Evaluation on Diverse Datasets.** The empirical evaluations of HELIP are conducted on specific datasets like CC3M, CC12M, YFCC15M and subsets of LAION. While these are widely accepted benchmarks, the generalizability of its improvements to real-world applications requires further evaluation.

## References


Muhammad Awais, Muzammal Naseer, Salman Khan, Rao Muhammad Anwer, Hisham Cholakkal, Mubarak Shah, Ming-Hsuan Yang, and Fahad Shahbaz Khan. 2023. Foundational models defining a new era in vision: A survey and outlook. *arXiv preprint arXiv:2307.13721*.

Alberto Baldrati, Marco Bertini, Tiberio Uricchio, and Alberto Del Bimbo. 2022. Conditioned and composed image retrieval combining and partially fine-tuning clip-based features. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2022, New Orleans, LA, USA, June 19-20, 2022*, pages 4955–4964. IEEE.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. 2014. Food-101–mining discriminative components with random forests. In *European conference on computer vision*, pages 446–461. Springer.

Gavin Brown, Mark Bun, Vitaly Feldman, Adam Smith, and Kunal Talwar. 2021. When is memorization of irrelevant training data necessary for high-accuracy learning? In *Proceedings of the 53rd annual ACM SIGACT symposium on theory of computing*, pages 123–132.

Tiffany Tianhui Cai, Jonathan Frankle, David J. Schwab, and Ari S. Morcos. 2020. Are all negatives created equal in contrastive instance discrimination? *ArXiv preprint*.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. 2021. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 9630–9640. IEEE.

Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. 2021. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 3558–3568. Computer Vision Foundation / IEEE.

Feilong Chen, Xiuyi Chen, Shuang Xu, and Bo Xu. 2022. Improving cross-modal understanding in visual dialog via contrastive learning. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7937–7941. IEEE.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020a. A simple framework for contrastive learning of visual representations. In *Proc. of ICML*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

Xinlei Chen, Haoqi Fan, Ross B. Girshick, and Kaiming He. 2020b. Improved baselines with momentum contrastive learning. *ArXiv preprint*.

Yihua Chen, Eric K Garcia, Maya R Gupta, Ali Rahimi, and Luca Cazzanti. 2009. Similarity-based classification: Concepts and algorithms. *Journal of Machine Learning Research*, 10(3).

Yufeng Cui, Lichen Zhao, Feng Liang, Yangguang Li, and Jing Shao. 2022. Democratizing contrastive language-image pre-training: A clip benchmark of data, model, and supervision. *Preprint*, arXiv:2203.05796.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 248–255. IEEE Computer Society.

Li Fei-Fei, Fergus Rob, and Pietro Perona. 2004. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW'04. Conference on*. IEEE.

Andreas Fürst, Elisabeth Rumetshofer, Viet Tran, Hubert Ramsauer, Fei Tang, Johannes Lehner, David P.

Kreil, Michael Kopp, Günter Klambauer, Angela Bitto-Nemling, and Sepp Hochreiter. 2021. CLOOB: modern hopfield networks with infoloob outperform CLIP. *ArXiv preprint*.

Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, et al. 2023. Datacomp: In search of the next generation of multimodal datasets. *ArXiv preprint*.

Shashank Goel, Hritik Bansal, Sumit Bhatia, Ryan A Rossi, Vishwa Vinay, and Aditya Grover. 2022. Cyclip: Cyclic contrastive language-image pretraining. *ArXiv preprint*.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2022. Towards a unified view of parameter-efficient transfer learning. In *Proc. of ICLR*. OpenReview.net.

Tri Huynh, Simon Kornblith, Matthew R. Walter, Michael Maire, and Maryam Khademi. 2022. Boosting contrastive self-supervised learning with false negative cancellation. In *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2022, Waikoloa, HI, USA, January 3-8, 2022*.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *Proc. of ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 4904–4916. PMLR.

Yannis Kalantidis, Mert Bülent Sariyildiz, Noé Pion, Philippe Weinzaepfel, and Diane Larlus. 2020. Hard negative mixing for contrastive learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 2013. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 554–561.

Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi. 2022a. BLIP: bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 12888–12900. PMLR.

Junnan Li, Ramprasaath R. Selvaraju, Akhilesh Gotmare, Shafiq R. Joty, Caiming Xiong, and Steven Chu-Hong Hoi. 2021. Align before fuse: Vision and language representation learning with momentum distillation. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 9694–9705.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*.

Yangguang Li, Feng Liang, Lichen Zhao, Yufeng Cui, Wanli Ouyang, Jing Shao, Fengwei Yu, and Junjie Yan. 2022b. Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. In *Proc. of ICLR*. OpenReview.net.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: common objects in context. In *Proc. of ECCV*.

S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi. 2013. Fine-grained visual classification of aircraft. Technical report.

Norman Mu, Alexander Kirillov, David A. Wagner, and Saining Xie. 2022. SLIP: self-supervision meets language-image pre-training. In *Proc. of ECCV*.

M-E. Nilsback and A. Zisserman. 2008. Automated flower classification over a large number of classes. In *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*.

Antonio Norelli, Marco Fumero, Valentino Maiorca, Luca Moschella, Emanuele Rodolà, and Francesco Locatello. 2022. Asif: Coupled data turns unimodal models to multimodal without training. *ArXiv preprint*.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*.

Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. 2015. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 2641–2649. IEEE Computer Society.

Filip Radenovic, Abhimanyu Dubey, Abhishek Kadian, Todor Mihaylov, Simon Vandenhende, Yash Patel, Yi Wen, Vignesh Ramanathan, and Dhruv Mahajan. 2023. Filtering, distillation, and hard negatives for vision-language pre-training. *CoRR*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proc. of ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proc. of EMNLP*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Joshua David Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. 2021. Contrastive learning with hard negative samples. In *Proc. of ICLR*. OpenReview.net.

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

Anshul Shah, Suvrit Sra, Rama Chellappa, and Anoop Cherian. 2022. Max-margin contrastive learning. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 8220–8230. AAAI Press.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proc. of ACL*, pages 2556–2565, Melbourne, Australia. Association for Computational Linguistics.

Mannat Singh, Laura Gustafson, Aaron Adcock, Vinicius de Freitas Reis, Bugra Gedik, Raj Prateek Kosaraju, Dhruv Mahajan, Ross Girshick, Piotr Dollár, and Laurens Van Der Maaten. 2022. Revisiting weakly supervised pre-training of visual perception models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 804–814.

Cory Stephenson, Suchismita Padhy, Abhinav Ganesh, Yue Hui, Hanlin Tang, and SueYeon Chung. 2021. On the geometry of generalization and memorization in deep neural networks. In *Proc. of ICLR*. OpenReview.net.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. 2011. The caltech-ucsd birds-200-2011 dataset. Technical report, California Institute of Technology.

Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *Proc. of ICML*, volume 119 of *Proceedings of Machine Learning Research*, pages 9929–9939. PMLR.

Bichen Wu, Ruizhe Cheng, Peizhao Zhang, Tianren Gao, Joseph E. Gonzalez, and Peter Vajda. 2022. Data efficient language-supervised zero-shot recognition with optimal transport distillation. In *Proc. of ICLR*. OpenReview.net.

Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. 2010. SUN database: Large-scale scene recognition from abbey to zoo. In *The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, San Francisco, CA, USA, 13-18 June 2010*, pages 3485–3492. IEEE Computer Society.

Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. 2022. FILIP: fine-grained interactive language-image pre-training. In *Proc. of ICLR*. OpenReview.net.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2021. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115.

Shengyu Zhang, Tan Jiang, Tan Wang, Kun Kuang, Zhou Zhao, Jianke Zhu, Jin Yu, Hongxia Yang, and Fei Wu. 2020. Devlbert: Learning deconfounded visio-linguistic representations. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 4373–4382.

## A Appendix: Algorithm

We summarize the Hard Pair Mining (HPM), the fast Hard Pair Mining (fastHPM) and the training pipeline of HELIP in Algorithm 1, 2 and 3 respectively.

Note, in the inner for loop, shown in Algorithm 1, the image and caption representations will be repeatedly computed. To accelerate the hard pair mining and avoid unnecessary computational overhead, we compute and save the encoded image features and text features. Besides, the outer loop is parallelized in the implementation.

## B Appendix: Discussion about baselines

In our experiments, we utilized CLIP, SLIP, and DECLIP as baseline models on CC3M, CC12M, YFCC15M, and Open29M datasets. To ensure

**Algorithm 1:** Hard Pair Mining (HPM)

**Input:** Hard pairs number per sample $k$
Pretrained unimodal vision model: $f_{\text{text}}$
Pretrained unimodal vision model: $f_{\text{image}}$
Dataset
$\mathcal{D} = \{(x_1^I, x_1^T), (x_2^I, x_2^T), \cdots, (x_N^I, x_N^T)\}$
Threshold for visual and textual modality $\tau_I$ and $\tau_T$
**Output:** Hard samples
$\mathcal{H} = [\mathcal{H}_1, \mathcal{H}_2, \cdots, \mathcal{H}_N]$

```
for i ∈ [1, N] do
    s ← [0, 0, ··· , 0]^⊤ ∈ R^N
    I_i ← f_image(x_i^I)
    T_i ← f_text(x_i^T)
    for j ∈ [1, N] do
        I_j ← f_image(x_j^I)
        T_j ← f_text(x_j^T)
        S⃗_j^I ← (I_i · I_j) / (||I_i||_2 · ||I_j||_2) if (I_i · I_j) / (||I_i||_2 · ||I_j||_2) >
          τ_I else 0
        S⃗_j^T ← (T_i · T_j) / (||I_i||_2 · ||T_j||_2) if (T_i · T_j) / (||T_i||_2 · ||T_j||_2) >
          τ_T else 0
        s_j ← S⃗_j^I · S⃗_j^T
    end
    H_i ← arg max(s, k)
    if ∃j ∈ H_i, s_j = 0 then
        H_i = ∅    # Indicate noise sample
end
```

**Algorithm 2:** fast Hard Pair Mining (fastHPM)

**Input:** Hard pairs number per sample $k$
Pretrained unimodal vision model: $f_{\text{text}}$
Pretrained unimodal vision model: $f_{\text{image}}$
Dataset
$\mathcal{D} = \{(x_1^I, x_1^T), (x_2^I, x_2^T), \cdots, (x_N^I, x_N^T)\}$
Threshold for visual and textual modality $\tau_I$ and $\tau_T$
Candidate pool size $C$
**Output:** Hard samples
$\mathcal{H} = [\mathcal{H}_1, \mathcal{H}_2, \cdots, \mathcal{H}_N]$

```
for i ∈ [1, N] do
    Uniformly C samples from Dataset D,
      D̄_i =
      {(x_1^I, x_1^T), (x_2^I, x_2^T), ··· , (x_C^I, x_C^T)}
    s ← [0, 0, ··· , 0]^⊤ ∈ R^N
    I_i ← f_image(x_i^I)
    T_i ← f_text(x_i^T)
    for j ∈ [1, C] do
        I_j ← f_image(x_j^I)
        T_j ← f_text(x_j^T)
        S⃗_j^I ← (I_i · I_j) / (||I_i||_2 · ||I_j||_2) if (I_i · I_j) / (||I_i||_2 · ||I_j||_2) >
          τ_I else 0
        S⃗_j^T ← (T_i · T_j) / (||I_i||_2 · ||T_j||_2) if (T_i · T_j) / (||T_i||_2 · ||T_j||_2) >
          τ_T else 0
        s_j ← S⃗_j^I · S⃗_j^T
    end
    H_i ← arg max(s, k)
    if ∃j ∈ H_i, s_j = 0 then
        H_i = ∅    # Indicate noise sample
end
```

our results are both compelling and reproducible, we primarily employed publicly available checkpoints as our baseline and rigorously tested the effectiveness of HELIP against these checkpoints. On CC3M, the checkpoint of SLIP model is released*. We enhanced its performance by applying HELIP which notably improved the zero-shot performance on ImageNet from 23.00 to 26.05. However, we noticed that the CLIP with ResNet50 on CC3M is missing. To address this, we undertook the pretraining ourselves. Our results were encouraging: the performance of our pretrained CLIP with ResNet50 achieved a score of 19.86, surpassing the 17.10 achieved by SLIP's CLIP with ViT-B/32 as reported in (Mu et al., 2022). This outcome suggests the robustness of our implementation. Besides, consistent with several prior studies, we found that on smaller pretraining datasets, CLIP with ResNet50 outperforms CLIP with ViT-B. On the CC12M dataset, a similar situation arose: while the SLIP checkpoint was available, the CLIP model was absent, leading us to undertake its pretraining. On the YFCC15M (v1) collected by (Radford et al., 2021), we trained the CLIP model. This resulted in a 25.46 score in the ImageNet zero-shot classification, closely aligning with the 26.10 outcome reported by (Cui et al., 2022). Additionally, for the YFCC15M (v2) dataset referenced in (Li et al., 2022b), both SLIP and DECLIP pretrained parameters were made available by (Li et al., 2022b), which we utilized directly as our baselines. On the larger dataset, Open29M, there was a lack of open-source pretrained checkpoints, prompting us to conduct the pretraining ourselves. Notably, the performance of our reimplementation (42.32) closely

*https://github.com/facebookresearch/SLIP#results-and-pre-trained-models

**Algorithm 3:** Hard samplE for boosting contrastive Language-Image Pretrained models (HELIP)

**Input:** $\mathcal{D} = \{(x_1^I, x_1^T), (x_1^I, x_1^T), \cdots, (x_N^I, x_N^T)\}$
Hard Pair Mining algorithm, HPM() # or the fastHPM()
Pretrained unimodal vision model: $f_{\text{text}}$
Pretrained unimodal vision model: $f_{\text{image}}$
Pretrained contrastive language-image model $\{\phi_{\text{image}}, \phi_{\text{text}}\}$
hyperparameters:
  Hard pairs number $k$
  Hard negative margin strength $\gamma$
  Sampled hard negatives number $p$
  Learning ratio $\eta$
  Batch size $b$
  Training iteration number $E$
  Visual and textual modality threshold $\tau_I$ and $\tau_T$
**Output:** CLIP model $\{\phi_{\text{image}}, \phi_{\text{text}}\}$

$\mathcal{H} \leftarrow \text{HPM}(\mathcal{D}, f_{\text{text}}, f_{\text{image}}, k, \tau_I, \tau_T)$
**for** $iter \in [1, E]$ **do**
  $B \leftarrow \{z_1, \ldots, z_b\} \overset{\text{i.i.d.}}{\sim} Uniform(\mathcal{D})$
  **for** $z_i \in B$ **do**
    $\mathcal{H}_i^p \leftarrow \{z_i, \ldots, z_p\} \overset{\text{i.i.d.}}{\sim} Uniform(\mathcal{H}_i)$
    $\overline{B} \leftarrow B \cup \mathcal{H}_i^p$
  **end**
  Compute loss $\ell_{\text{finetune}}$, Equation (6), with samples $\overline{B}$ $\phi_{\text{image}} \leftarrow \phi_{\text{image}} + \eta \cdot \partial_{\phi_{\text{image}}} \ell_{\text{finetune}}$
  $\phi_{\text{text}} \leftarrow \phi_{\text{text}} + \eta \cdot \partial_{\phi_{\text{text}}} \ell_{\text{finetune}}$
**end**

aligns with the results reported by (Li et al., 2022b), indicating the effectiveness of our approach.

## C Appendix: Analysis of the Impact of Subset Size on Hard Pair Selection in FastHPM

In the comparison of HPM and FastHPM detailed in Section 4.5, we explore the efficacy of using 3M and 6M subset sizes of the CC12M dataset in FastHPM for mining hard pairs. The result, Table 6, shows that with a reduced subset size as small as 3 million entries, mining hard pairs and further training with these pairs can boost CLIP to achieve competitive performance with full set for mining.

In this section, we delve deeper into the analysis of hard pairs mined by FastHPM across varying subset sizes. Based on the selection criteria defined by FastHPM (Equation 5), we denote the *selection criteria value* as $\widetilde{S}^I(x_i^I, \mathcal{H}_i^\star(j))^\top \widetilde{S}^T(x_i^T, \mathcal{H}_i^\star(j))$. Here, $\mathcal{H}_i^\star(\cdot)$ represents a pair within the set of hard pairs $\mathcal{H}_i^\star$, mined by FastHPM for a specified target pair $i$ under a given subset size. Additionally, the $j$ in $\mathcal{H}_i^\star(j)$ indicates the $j$-th hard pair within the set $\mathcal{H}_i^\star$. Note, a higher selection criteria value signifies a harder mined pair.

We present the average selection criteria values for top-k hard pairs in Figure 6. As depicted by the grey horizontal line, the average selection criteria values for the top-20 hard pairs selected by FastHPM-1.5M, the top-40 by FastHPM-3M, and the top-80 by FastHPM-6M all approximate 0.477. This figure indicates that a further reduction in the subset size might necessitate adjustments to the number of hard pairs sampled to preserve quality. For instance, in our experiments detailed in Table 6, we uniformly sampled hard pairs for training from the top 50 for HELIP-3M. As Figure 6 suggests, a sampling range of 10 for HELIP-1M might be effective. Particularly, considering that HELIPsignificantly boosted the pre-trained models with just an additional training epoch, as discussed in Section 4.2, selecting one hard pair for each target pair from a pool of 10 will be feasible.

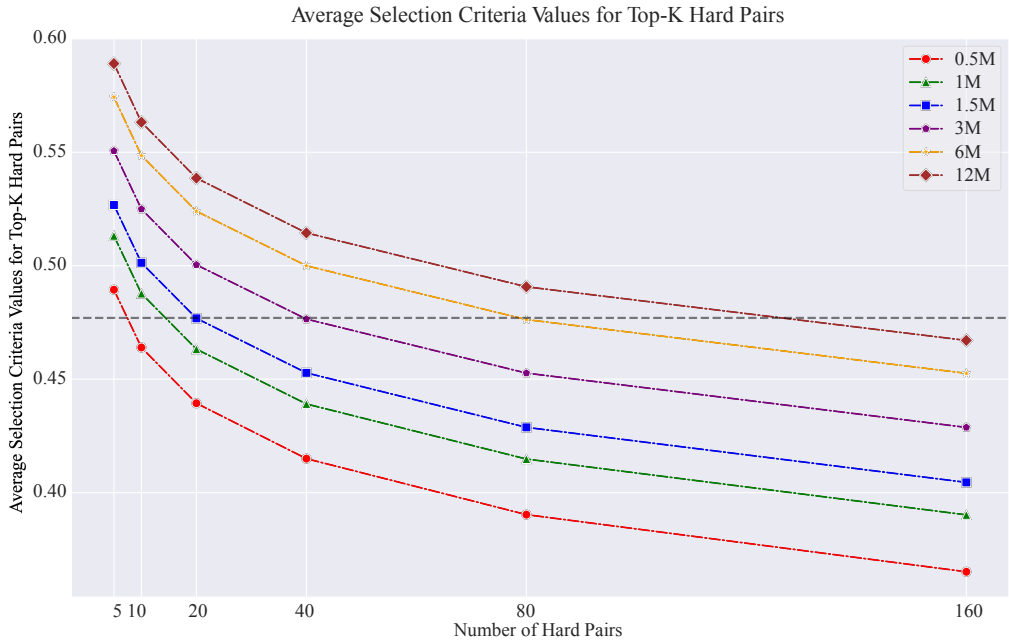



Figure 6: The average selection criteria values for hard pairs mined by FastHPM with different subset sizes.

## D Appendix: Implementation Details

Our experiments are conducted across three distinct architectures: ResNet-50, ViT-B/16, and ViT-B/32, tailored to various datasets and pretrained models. Specifically, for loading the pretrained CLIP model on CC3M and CC12M, the ResNet-50 is used as the image encoder. Besides, to align with existing checkpoints established by Mu et al. (2022), we use ViT-B/16 for SLIP model experiments on CC3M and CC12M, respectively. And, we use ViT-B/32 for pretraining on YFCC15M v1, v2, and Open29M datasets to ensure fair comparison with the previous results (Li et al., 2022b). Furthermore, for the SLIP and DECLIP models, we adapt the pretrained parameters from the publicly available resources[†] The input resolution of the image encoder is 224 $\times$ 224 and the maximum context length of the text encoder is 77. All of our experiments are conducted on 8 V100 GPUs with a batch size of 128 for ViT-B/16 models, and a batch size of 512 for ResNet-50 models and ViT-B/32 models. The dimension of the image and text embeddings is 1024 for ResNet-50 models and 512 for ViT-B/16 and ViT-B/32 models. We set $\tau = 0.5$, $\gamma = 1$, $k = 50$ and $p = 1$ for all the experiments by default. Automatic mixed-precision is used to save GPU memory. To keep the model from overfitting, we use early stopping if there is no performance gain on ImageNet zero-shot accuracy in 5 epochs. It is worth noting that using zero-shot classification performance on ImageNet as a criterion for early stopping is a commonly used practice for the training of CLIP (Radford et al., 2021; Mu et al., 2022).

To reflect that our method is designed to work with few assumptions on encoder, we used encoders pretrained over a single-modal source rather than multimodally pretrained ones when preparing hard negative pairs. Specifically, we used an unsupervised pre-trained vision transformer, DINO VITs8 (Caron et al., 2021), and a Sentence Transformer (SentenceT) (Reimers and Gurevych, 2019) to encode text. For DINO VITs8, the embedding size is 384, while for SentenceT, it is 768.

[†] https://github.com/facebookresearch/SLIP, https://github.com/Sense-GVT/DeCLIP.

## E Appendix: Performance of HELIP on noisy dataset

We expanded our investigation to assess the effectiveness of HELIPon subsets of LAION7.5M and 8M, which are randomly sampled from LAION (Schuhmann et al., 2022). The results are detailed in Table 8. The CLIP model, enhanced with HELIP consistently outperformed its original counterpart on both subsets across a majority of the

evaluated datasets, including ImageNet, CIFAR10, CIFAR100, Caltech, and Food. On the 7.5M subset, HELIP enhances performance across all datasets by an average of 3.6%. Although CLIP scores slightly higher on the Sun dataset, HELIPboosts its overall performance with an average improvement of 2.5% on the 8M subset. These results highlight the enhanced performance achieved through HELIP, demonstrating its robustness and effectiveness in improving existing models that have been pretrained on noisy data.

## F Appendix: Analysis of the Impact of $\tau$ on Hard Pair Selection

To examine the impact of the threshold parameter $\tau$ on the selection of hard pairs, we analyze the similarities in the rankings of hard pairs (using Kendall Rank Similarity) mined by HPM under various $\tau$ values. The hard pairs are ranked by using the selection criteria value mentioned in Appendix C. The results on the CC12M dataset are displayed in Figure 7. We observe that the selection of hard pairs is robust to changes in the $\tau$ value. This resilience is partly because we only mine the top 50 hard pairs, a subset unlikely to be significantly affected when $\tau \leq 0.5$.

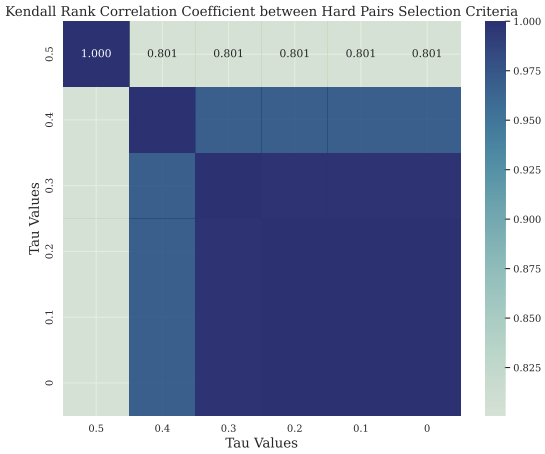



Figure 7: The Impact of $\tau$ on Hard Pair Selection.

## G Appendix: Analysis of the Impact of Mitigating Noisy Data

As presented in Section 3.2, to enhance the overall quality and reliability of the training dataset, data pairs lacking substantial support from the entirety of the training data are considered unsuitable and removed.

This section further empirically analyzes the impact of our noise mitigation strategy by detailing the quantity and nature of pairs removed across various datasets. Specifically, our approach removes 4.67% of the pairs from CC3M, 3.64% from CC12M, and 7.41% from YFCC15M, before continuing with pretraining. Figure 8 visualizes the pairs filtered from CC12M. Notably, our strategy effectively removed pairs such as unavailable images (e.g., two blank or white images in the second row) and mismatched pairs. These results suggest that our noise mitigation strategy can effectively clean the data using two single-modality models before training a CLIP model from scratch.

## H Appendix: Comparison with Other Hard Data Selection Method

We evaluate the efficacy of the proposed method in enhancing the discriminative capacity of learned representations by comparing its zero-shot classification performance with that of other hard data mining strategies. As described in the Section 2, a common way to define hard data is through intra-modality similarity. Hence, we introduce the hard data mining methods depending on (sample level) image similarity mining and text similarity mining and denote them as IM and TM respectively. For a given target pair, we compute the cosine similarity between its image/text representation and that of the remaining dataset. The image and text representations are encoded using a pretrained Resnet50 and BERT, respectively. As the preprocessing step, IM and TM methods mine hard negatives before continuous pretraining. Subsequently, we integrate the mined hard negative pairs into the training pipeline of CLIP and denote them as CLIP+IM and CLIP+TM and optimize the original contrastive loss to fine-tune the model. Additionally, we also include the hard negative contrastive loss, HN-NCE, proposed by Radenovic et al. (2023), as a baseline. HN-NCE upsamples the weight of hard-negatives identified by the current model. As shown in Table 9, when the CC3M pretrained CLIP model is combined with HELIP, the performance of our pair-level hard data mining method significantly outperforms other sample-level techniques. Besides,we observe that compared to the baseline CLIP performance, the introduction of TM and IM methods results in a decline in performance. To better understand the reasons behind this drop, we analyzed the outputs of the TM and IM methods in

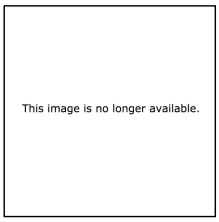
| | ImageNet | CIFAR10 | CIFAR100 | Caltech | Food | Sun | Avg. |
|---|---|---|---|---|---|---|---|
| CLIP-7.5M | 23.5 | 34.6 | 14.5 | 58.9 | 28.6 | 25.3 | 30.8 |
| CLIP-HELIP-7.5M | **25.8** | **39.9** | **16.7** | **61.9** | **34.1** | **28.2** | **34.4** |
| CLIP-8M | 25.1 | 31.1 | 12.9 | 60.9 | 29.5 | **27.5** | 31.2 |
| CLIP-HELIP-8M | **26.5** | **38.8** | **14.6** | **62.3** | **33.1** | 26.6 | **33.7** |

Table 8: Zero-shot performance of CLIP on two LAION subsets.

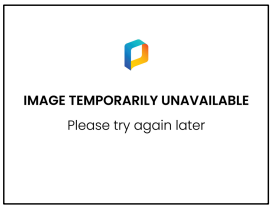
person have sent us images of person .

unfortunately , there are no appropriate pictures of us on the plane .

industry of turquoise , nice way to handle the pass through




my nephew met my cat for the first time .

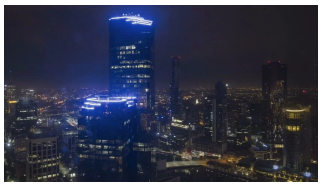



a male hiker looks over a view of granite mountains from the summit .

what should i do if i have bad cuts on the sides of my mouth ?

person - the lady by the lake

a street sign with the name of the place



cloud , sun rays burst through the clouds during sunrise .

Figure 8: Visualization of the image-caption pairs filtered out from CC12M.

detail. In Figure9, we illustrate the data obtained through three distinct preprocessing methods: Hard Pair Mining (HPM), Image Similarity Mining (IM), and Text Similarity Mining (TM). The first row depicts the image-text pairs identified by HPM, while the second and third rows showcase the pairs mined by IM and TM, respectively. For TM (IM displays similar issues), the selected pairs often feature captions that are highly similar or identical, which is typical in data collected from the web. Even though identical pairs may not always be present, repetitions of the same images or text are common. According to the CLIP contrastive loss (Equation 1), the model is forced to push nearly identical caption representations toward and away from two distinct image representations at the same time.This inherent contradiction in objectives contributes to a degradation in performance. To illustrate, consider a target pair $(T_{\text{target}}, I_{\text{target}})$ and a mined pair $(T_{\text{mined}}, I_{\text{mined}})$ using TM, where $T_{\text{target}} \approx T_{\text{mined}}$ but $I_{\text{target}} \not\approx I_{\text{mined}}$. In the contrastive loss framework, the model aims to minimize the distance between $(I_{\text{target}}, T_{\text{target}})$ and maximize the distance between $(I_{\text{target}}, T_{\text{mined}})$. However, the near-identity of $T_{\text{target}}$ and $T_{\text{mined}}$ leads to conflicting optimization targets and a potential decline in performance.

# I Appendix: Impact of different encoders in HPM

We explored the effect of different pretrained encoders on HPM's performance by alternating image and text encoders. Initially, the unsupervised pretrained DINO VITs8 (Caron et al., 2021) was paired with the SentenceT (Reimers and Gurevych,

| | Imagenet | CIFAR10 | CIFAR100 |
|---|---|---|---|
| CLIP | 19.04 | 33.06 | 13.77 |
| CLIP + TM | 16.70 | 28.71 | 9.67 |
| CLIP + IM | 16.93 | 29.22 | 10.42 |
| CLIP + HN-NCE | 19.47 | 29.88 | 11.83 |
| CLIP + HELIP | **19.86** | **34.05** | **14.13** |

Table 9: **Zero-shot performance of CLIP pre-trained on CC3M boosted by hard data mined by different methods.** HELIP shows superior performance, consistently outperforming local/global hard sample mining techniques by a substantial margin.

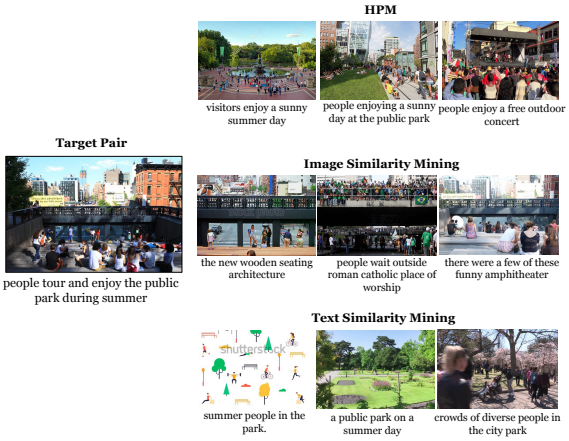



Figure 9: **Hard negative data selected by different methods.** Compared to data mined using the sample-level (image/text modal) similarity, hard pairs mined by HPM are more similar to the target.

2019) transformer, trained on over a billion internet-based sentences. This combination was later swapped for the SWAG VITb16 (Singh et al., 2022) and the T5 (Raffel et al., 2020). Additionally, experiments using OpenAI's CLIP model (Radford et al., 2021) multimodal encoders were conducted. Interestingly, as Table 10 suggests, the encoder choice seemingly has negligible impact on HPM's performance, likely due to the proficiency of current pretrained models in modeling intra-modal similarities. Moreover, the ability to use single-modal pretrained models and still achieve competitive or superior performance implies that there's no assumption of having access to a high-quality CLIP model, such as OpenAI's CLIP-400M.

## J Appendix: More visualization results

We offer further visualization results pertaining to the hard samples mined by various methods. As depicted in Figure 10, the hard samples sourced by HPM closely resemble the target sample (seen at the top left). Interestingly, for samples with fewer objectives, the image and text mining method can

| | ImageNet | CIFAR10 | CIFAR100 | Avg. |
|---|---|---|---|---|
| CLIP Encoders | 19.57 | 33.28 | 13.53 | 22.12 |
| VITs8+SentenceT | 19.86 | 34.05 | 14.13 | 22.68 |
| VITb16+SentenceT | 19.62 | 35.53 | 14.67 | 23.27 |
| VITs8 + T5 | 19.61 | 33.99 | 13.82 | 22.47 |

Table 10: The zero-shot performances of HELIP with different encoders in HPM. HPM's performance is insensitive to the selection of encoders.

identify a reasonably challenging counterpart, as seen in the case of "the harbor in a small village". However, for intricate scenes, only the HPM is capable of yielding sufficiently challenging samples, like the scenario "people touring and enjoying the public park during summer". The dataset acquired from the web encompasses a myriad of such intricate cases. We posit that this is why training with hard samples unearthed by HPM yields more proficient outcomes.

Moreover, we present additional visualization results for hard samples mined via different techniques. Hard samples extracted by HPM exhibit a stronger resemblance to the target sample, as highlighted in Figure 10 (top left). We observed that the image and text mining methods can provide a relatively fitting hard counterpart for simpler samples, like "the harbor in a quaint settlement". However, for more intricate scenes, only the HPM method produces samples of adequate difficulty, such as "people touring and relishing the public park throughout summer". The web-based dataset includes a significant proportion of these complex cases. Consequently, we infer that training with hard samples mined by HPM results in enhanced performance.

## K Appendix: Future work

Moving forward, several possibilities for future research emerge. First, we aim to explore composition-aware fine-tuning for VLMs, which could potentially enable more effective utilization of multimodal information. Moreover, we are intrigued by the prospect of combining parameter-efficient tuning (He et al., 2022) with HELIP potentially further enhancing performance. Another area of interest is scaling up the dataset size and examining the applicability of the scaling law to our method. We also intend to investigate how the integration of our boosting algorithm might alter the multimodal dataset curation algorithm (Gadre et al., 2023). Ultimately, we hope our work will

serve as a catalyst for additional research in the fine-tuning of pre-trained, large-scale multimodal models.

## L Appendix: Potential Risks and Ethical Considerations

While our research contributes to advancements in language-image pretraining, it is important to acknowledge potential risks and ethical considerations associated with our work.

**Privacy Concerns:** The datasets used—CC3M, CC12M, subsets of YFCC15M, LAION7.5M, and LAION8M—are publicly available and sourced from the internet. They may contain personal identifiable information (PII) or images of individuals. We rely on the dataset providers' curation processes to remove personal identifiable information.

**Bias and Fairness Issues:** The datasets may not be representative of all demographics or cultures, potentially leading to models that perform unevenly across different groups. This lack of fairness can perpetuate existing societal biases and inequalities.

**Misuse of Technology:** The models developed could be misused for malicious purposes, such as generating deepfakes, enabling unauthorized surveillance, or creating misleading information, which could have negative societal impacts.

To address these potential risks, the following steps could be considered in future work:

**Enhanced Data Filtering:** Currently, we rely on the curation and filtering processes conducted by the dataset creators for the datasets we used—CC3M, CC12M, YFCC15M subsets, and subsets of LAION. Implementing additional data cleaning procedures to identify and remove personally identifiable information (PII) and offensive content from the training datasets is an important area deserving future study.

**Bias Mitigation Techniques:** Incorporate fairness-aware learning algorithms and conduct thorough evaluations to detect and reduce biases in the model's output.

**Transparency and Accountability:** Provide documentation detailing the data sources, model limitations, and potential biases to inform users and stakeholders.

## M Appendix: Licensing

All datasets used in this study are open-source and utilized in accordance with their respective licenses. Specifically, CC3M and CC12M comprise images with Creative Commons licenses, YFCC100M is distributed under a Creative Commons Attribution license, and LAION-5B consists of web-sourced data under permissive licenses. Our use of these datasets is strictly for research purposes and complies with their licensing terms.

## N Appendix: Use of AI Assistants

We acknowledge the use of AI assistants in the preparation of this work. Specifically, we utilized tools, GPT4-o, for proofreading the manuscript and GitHub Copilot for assisting with coding tasks. These AI tools were employed to enhance productivity and efficiency. All content generated with the assistance of AI was thoroughly reviewed and edited by the authors to ensure accuracy and originality. The responsibility for the final content of this paper rests solely with the authors.

## O Appendix: Artifact Use Consistent with Intended Purpose

In our research, we utilized several publicly available datasets—CC3M, CC12M, YFCC15M subsets, LAION7.5M, and LAION8M—that are widely used within the computer vision and machine learning communities for the purpose of training and evaluating language-image models. The use of these datasets in our study is consistent with their intended purpose, as specified by their creators, which is to advance research in image recognition, captioning, and related fields. All datasets were used strictly for non-commercial, research-oriented objectives, adhering to the access conditions and licenses provided by the dataset providers.

For the artifacts we created during this research, including the trained models and any derived datasets or code, we specify that their intended use is for academic and research purposes only. These artifacts are shared to promote transparency, reproducibility, and further advancement in the field. We ensure that this intended use is compatible with the original access conditions of the datasets we used, particularly considering any restrictions on derivative works or redistribution. When sharing our artifacts, we comply with all applicable licenses

and access terms, and we encourage others who use our artifacts to do the same.

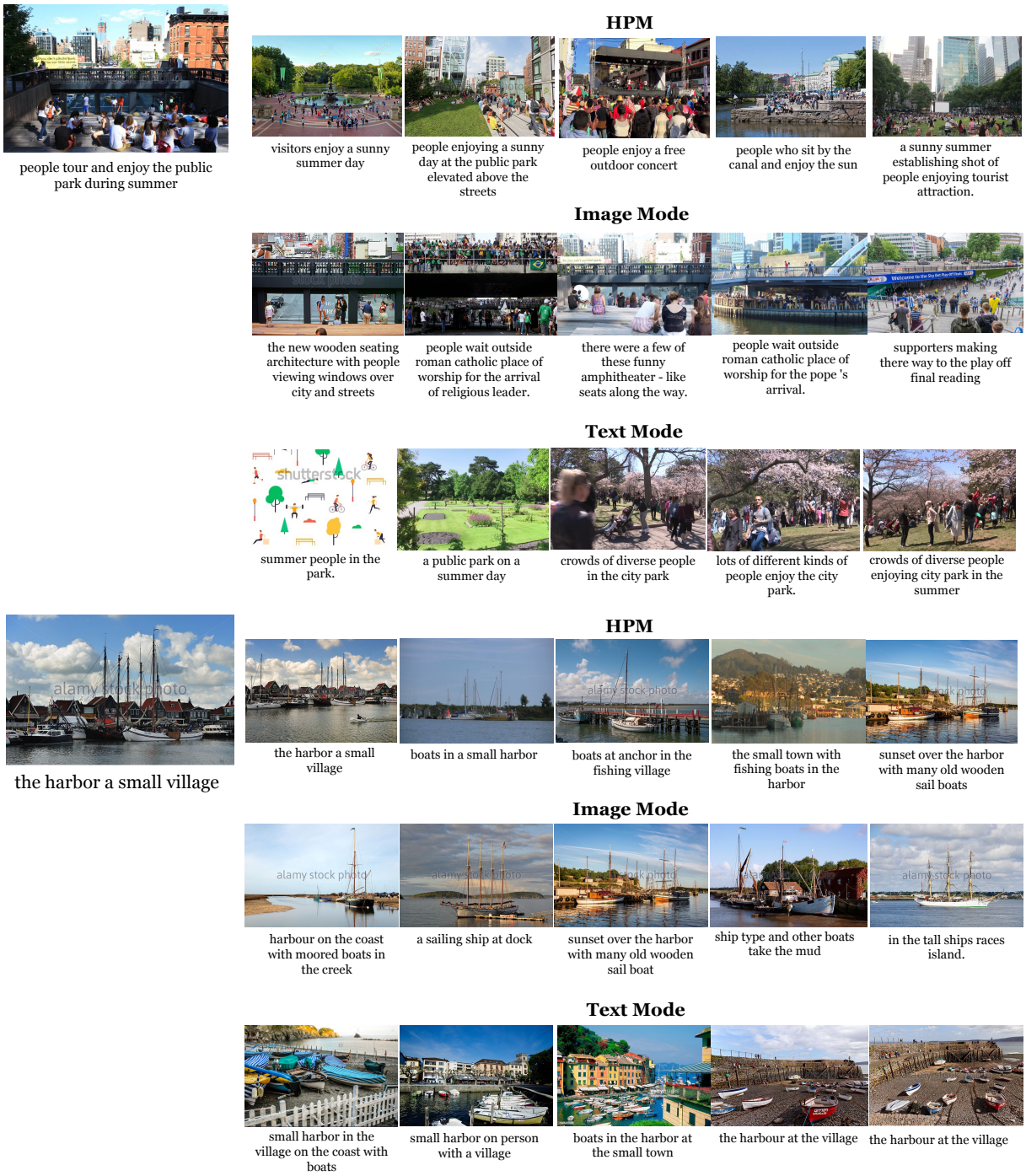



Figure 10: **Hard pairs selected by different methods.**