Extra Training Provides a Strong Baseline for CLIP

Alaa Khaddaj* alaakh@mit.edu MIT
Hadi Salman* hady@mit.edu MIT
Andrew Ilyas ailyas@mit.edu MIT
Guillaume Leclerc gleclerc@mit.edu MIT

Aleksander Mądry madry@mit.edu MIT

Abstract

Contrastive Language-Image Pretraining (CLIP) models exhibit good performance on a range of vision tasks. To improve the performance of this class of models even further, several works have proposed to modify the CLIP training procedure. In this work, we show that it is possible to achieve substantial gains using a much simpler strategy. Specifically, existing CLIP models—especially those trained on smaller datasets—tend to be undertrained. As a result, simply extending the training procedure according to a simple heuristic can significantly improve the performance of CLIP models.

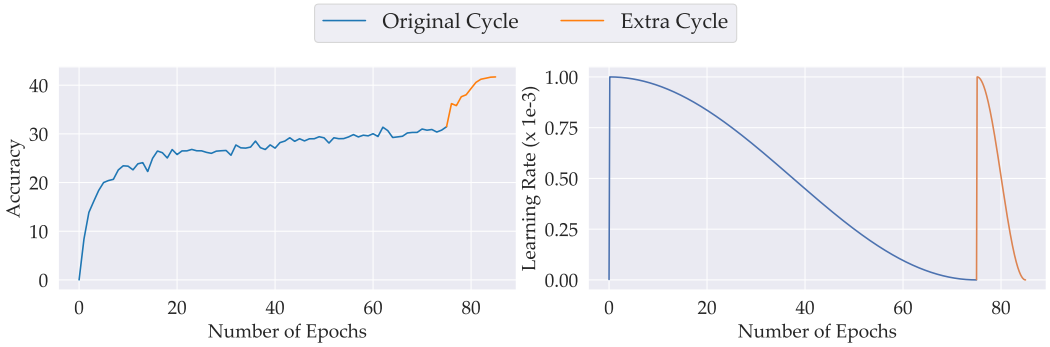


Figure 1: CLIP models trained on smaller datasets are undertrained. The blue curve on the left corresponds to the zero-shot accuracy of a CLIP model trained on CC12M for 75 epochs. This model was trained using the learning rate schedule represented by the blue curve on the right, and achieves a zero-shot ImageNet accuracy of 31%. After resetting the scheduler and training for 10 additional epochs (orange curve on the right), the zero-shot accuracy increases by 10% (orange curve on the left).

1 Introduction

Zero-shot inference has become a popular paradigm in computer vision. In this paradigm, one trains a model on vast amounts of data and leverages the resulting *embeddings* (without fine-tuning or weight updates) to perform inference on downstream tasks [BMR+20; RKH+21]. One popular

*Equal contribution.

R0-FoMo: Workshop on Robustness of Few-shot and Zero-shot Learning in Foundation Models at NeurIPS 2023.

approach to zero-shot image classification is the Contrastive Language-Image Pre-Training (CLIP) [RKH+21] framework. CLIP models perform remarkably well on a range of classification tasks, including ImageNet [RRS+19] and its variants [HZB+19; BMA+19; WGX+19; HBM+20].

The success of CLIP has led to a line of work that aims to further improve the performance of CLIP models [YWV+22; MKW+21; LFH+22; WIK+21]. These approaches typically modify the CLIP architecture or loss function, and achieve better downstream performance compared to a baseline CLIP model trained with a canonical training procedure.

Our contributions. We find that a simple addition to the CLIP training recipe—namely, resetting the learning rate scheduler and training for a few more epochs—can significantly improve CLIP models' downstream utility for zero-shot classification. Improvements from additional training are particularly pronounced for models trained on smaller datasets such as Conceptual Captions 3M and 12M [SDG+18; CSD+21].

Extra training provides a simple yet effective baseline for CLIP performance on these smaller datasets. Our results also suggest that the canonical training procedure for CLIP on small datasets may lead to undertraining.

2 Background

Contrastive language-image pretraining, or CLIP, is a representation learning framework that uses paired image-caption data to learn general representations for both text and images. Specifically, a CLIP model comprises of two neural networks: an image encoder E_I mapping images to embedding vectors in $\mathbb{R}^d$ and a text encoder E_T mapping text to the same space.

Given a dataset of image-caption pairs (x_i, t_i), where x_i is an image and t_i is the corresponding textual caption, CLIP models encode the image x and the caption t using the relevant encoders into an image embedding z_I and a caption embedding z_T, respectively, that belong to the same space. CLIP models are then trained by matching the corresponding image and caption embeddings of a given batch, i.e., by minimizing the cross-entropy loss between the embeddings of matching image-caption pairs, and maximizing the loss between non-matching pairs.

CLIP models trained on large datasets, e.g., LAION [SVB+21], achieve remarkable results [RKH+21], however, are very expensive to train. Instead, many works opt to train CLIP models on smaller datasets, such as Conceptual Captions 3M and 12M [SDG+18; CSD+21].

Zero-shot classification. This framework lends itself naturally to "zero-shot" classification. In particular, given a new dataset of image-label pairs, we can embed the natural-language description of each possible label using the text encoder. We can then classify each image by first encoding it using the image encoder, then finding the closest label in the embedding space.

Performance. CLIP models demonstrate impressive zero-shot performance on several image classification tasks, such as ImageNet classification [DDS+09]. Recent large CLIP models (trained on LAION [SVB+21]) surpass 90% accuracy on ImageNet [FSW+23]. Furthermore, CLIP models tend to be quite robust to distribution shift. For example, the achieve high accuracy on the corresponding ImageNet variations benchmarks [RRS+19; HZB+19; BMA+19; WGX+19; HBM+20].

Given the impressive performance of CLIP models, there are a number of works interested in further improving it. For example, prior work has proposed modifying the training objective [MKW+21; LFH+22], imposing additional supervision during training [LLZ+22], leveraging additional data augmentations [FAR+23], or imposing a particular structure on the learned representations [FRL+22; GBB+22].

3 Extra Training Provides a Strong Baseline for CLIP

In this section, we investigate the performance of CLIP models and show they might be undertrained when trained on smaller datasets. We then propose a simple modification to the training procedure that significantly improves performance, and study its effectiveness when applied to models trained

on larger datasets. We finally compare our method with other proposed approaches and discuss the implications of our results on the research community.

3.1 Method

CLIP models are typically trained with a "cosine" learning rate schedule, where the learning rate increases linearly to a maximum value, then decays on a cosine curve (see Figure 1). Our "extra training" procedure consists of two steps, to be performed after the model's training has concluded. Specifically, we first reset the learning rate schedule, along with the optimizer's parameters. We then train for k more epochs.

Note that extra training is not equivalent to simply training the model for more epochs, since the optimizer is reset to its original state. Instead, our method is closer to initializing the CLIP model weights to that of a model pretrained on the same training dataset. Consequently, the implementation of our method is straightforward and consists of first training a CLIP model on the dataset then use the resulting weights to initialize a new CLIP model to be trained (for a small number of epochs) on the same dataset.

3.2 Results

We study the effect of extra training on CLIP's downstream performance in two contexts, which we call the *small-data* and *big-data* regimes. In the small-data regime, we train our own CLIP models using OpenCLIP's recipe [IWW+21] and match their reported accuracies. In the big-data regime, we leverage one of the OpenCLIP models pretrained on LAION-400M [SVB+21].

In the small-data regime, CLIP models are undertrained. Applying our extra training cycle outlined in Section 3.1 improves significantly and consistently the performance of several CLIP models on a range of downstream tasks. For example, the ImageNet zero-shot accuracy of several CLIP models increases by an average of 8% after applying our extra training cycle (see Table 1 for more results). Note that simply training for longer does not improve performance, as the accuracy of the CLIP model saturates after few training epochs (see Figure 1).

Model	ImageNet	ImageNet-V2	ImageNet-A	ImageNet-O	ImageNet-R	ImageNet-Sketch	ObjectNet
ResNet-50	41.7 (+11.3%)	35.6 (+9.84%)	9.44 (+4.39%)	47.5 (+8.15%)	53.5 (+11.3%)	31.1 (+8.55%)	28.2 (+8.24%)
ViT-B-32	38.6 (+7.83%)	33.1 (+7.14%)	7.76 (+2.75%)	43.9 (+7.70%)	52.8 (+10.2%)	31.6 (+7.95%)	22.1 (+6.63%)
ViT-B-16	44.9 (+8.40%)	38.0 (+7.09%)	12.0 (+3.59%)	45.5 (+5.60%)	60.5 (+9.77%)	35.1 (+8.17%)	30.3 (+7.73%)

Table 1: Our simple training procedure consistently improves the performance of CLIP models trained on CC12M. This table shows the zero-shot accuracy of several CLIP models on different downstream tasks after applying our simple strategy. The numbers in parentheses represent the absolute change in zero-shot accuracy on the corresponding downstream classification task. Note that the performance of several CLIP models improves significantly on each downstream task. For example, applying our simple strategy on a ResNet-50 CLIP model leads to a zero-shot accuracy of 41.7% on ImageNet—an improvement of 11.3% compared to the performance reported by the literature. See Appendix B for additional results.

In the large-data regime, CLIP models are less undertrained. Our previous experiment reveals that CLIP models trained on smaller datasets might be undertrained. In this section, we investigate whether CLIP models trained on large-scale datasets, such as LAION-400M [SVB+21], might be undertrained. To this end, we consider a CLIP model (with ViT backbone) pretrained on LAION-400M [IWW+21] and we apply the additional training procedure for 15 extra epochs (on the same dataset). Comparing the zero-shot performance of the new model to that of the original model, we observe that both models achieve similar performance on a range of datasets (see Table 2), suggesting that undertraining is less of an issue at scale.

https://github.com/mlfoundations/open_clip

	ImageNet	ImageNet-A	ImageNet-O	ImageNet-R	ImageNet-Sketch	ImageNet-V2	ObjectNet
OpenCLIP	62.94	**21.69**	53.45	73.40	49.39	**55.11**	**43.91**
Ours	**63.29**	19.80	**56.05**	**75.51**	**52.82**	54.84	43.09

Table 2: CLIP models trained on large datasets are less likely to be undertrained. Comparison of the performance of two ViT-B-32 CLIP models on several downstream tasks. The first row corresponds to a public CLIP model trained on LAION 400M [IWW+21], while the second row corresponds to applying our strategy to the public model. Note that training for an extra cycle leads to similar performance, which suggests that CLIP models trained on large datasets are less likely to be undertrained.

3.3 Discussion and Implications

In Section 3.2, we show that CLIP models trained on smaller datasets might be undertrained and propose a simple strategy to mitigate undertraining. In this section, we compare our proposed strategy with other approaches employed to improve the performance of CLIP models.

Specifically, we consider several previously proposed methods for refining the basic CLIP training procedure [CWL+22; GBB+22; FRL+22; LLZ+22; FAR+23]. These approaches modify the CLIP training objective or impose additional structure on the learned representations. On smaller datasets, CLIP models trained using the proposed approaches show an accuracy improvement compared to baseline CLIP models. However, it turns out that simply applying the additional training procedure from this work yields competitive results (see Table 3). Our results underscore the importance of having strong baselines for CLIP training, and of applying proposed approaches to large datasets when compute is available.

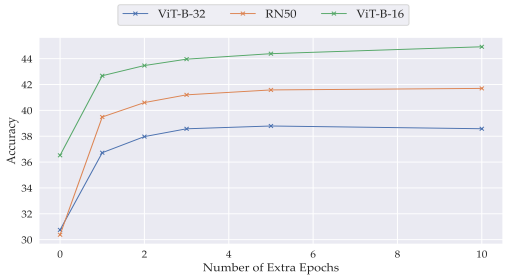
Method	Pretraining	
	CC3M	CC12M
CLIP (baseline) [RKH+21]	20.6	36.5
ProtoCLIP [CWL+22]	21.5	–
CyCLIP [GBB+22]	22.1	–
CLOOB [FRL+22]	24.0	–
DeCLIP [LLZ+22]	27.2	41.0
CLIP (Improved) [FAR+23]	27.4	44.4
CLIP (Ours)	24.2	41.7

Table 3: Comparison of different CLIP training strategy. The first row displays ImageNet zero-shot accuracy of ResNet-50 CLIP model trained using the canonical training scheme. Subsequent rows display the performance of CLIP models trained according to different strategies in the literature. Last row presents the performance obtained by applying our additional training strategy. Note that the various approaches in the literature improve the performance of the original model.

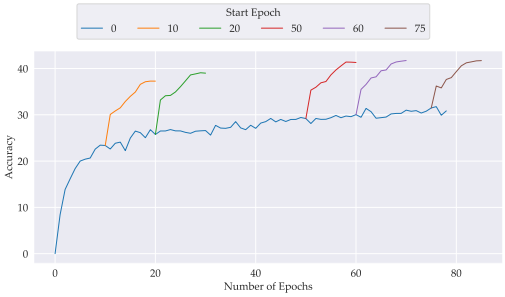
3.4 Studying and Mitigating Undertraining

In Section 3.2, we show that CLIP models trained on smaller datasets might be undertrained and propose a simple strategy to improve the performance of CLIP models. In this section, we study how applying our strategy at different epochs and for different durations affects the performance of CLIP models.

How many extra epochs should we train for? As we show in Section 3.2, resetting the learning rate scheduler and training for a fixed number of additional epochs improves CLIP performance. We now want to study how the zero-shot performance of several CLIP models changes as we vary the number of additional training epochs K.



(a) Applying the additional training procedure for few extra epochs is enough to improve performance. The ImageNet zero-shot accuracy of several CLIP models (y-axis) increases as we apply the additional training procedure for more epochs (x-axis). Note that the performance improvement saturates after applying the procedure for only three additional epochs. See Appendix B for additional results.



(b) Applying our strategy to improve performance. The blue curve corresponds to the accuracy of the original CLIP model. Each other curve represents the zero-shot accuracy of the CLIP model after applying our strategy with different starting points. For example, the orange curve corresponds to applying our strategy after 10 training epochs. Note that applying our strategy earlier during training leads to a performance improvement beyond the final accuracy reached after training for 75 epochs. See Appendix B for additional results.

To this end, we train three CLIP models (ResNet-50, ViT-B-32 and ViT-B-16), and then apply the extended training procedure to them, each time with a different number of additional training epochs. We observe that for models trained on CC12M, performance improvement saturates after applying three extra epochs (see Figure 2a). This suggests that the extra overhead needed to achieve peak performance can be fairly minor.

At which epoch should we apply the extended training? So far, we have been applying the extended training after the original model training was completed. However, would it be beneficial to apply it earlier in the training?

It turns out that such an earlier application can improve the model performance beyond what can be achieved with the full training cycle. To fully examine this phenomenon, we explore stopping the original training procedure at different epochs, and in each case we restart the training (and the learning rate scheduler) for 10 additional epochs. We observe that applying this extended training as early as 10 epochs into the original training cycle already improves performance beyond what is achieved after a complete training of the original model (see Figure 2b). For example, our model reaches an accuracy of 37% after a total of only 20 epochs, higher than the original model's final accuracy of 31% that was trained on 75 epochs.

Using a cyclic learning rate can improve CLIP training As mentioned in Section 3.2, CLIP models typically employ a single-cycle cosine learning rate scheduler [LH17] (see Figure 1). We showed, however, that 1) applying this schedule for a single cycle leads to suboptimal performance, and 2) employing an additional *short* cycle boosts accuracy. This additional cycle is reminiscent of cyclic learning rate schedulers. To investigate whether the use of such schedulers improves performance, we train a CLIP model using a multicycle cosine learning rate scheduler. Using this strategy, we obtain CLIP models that—with much fewer training epochs—outperform the CLIP models trained with the standard cosine learning rate schedule (see Figure 3).

4 Related Work

Zero-shot inference is an increasingly popular paradigm where the goal is to solve a specific task that is unseen during training [RKH+21]. One notable example of zero-shot models is the CLIP

The first 10 epochs correspond to the original training cycle, while the second 10 epochs correspond to our strategy.

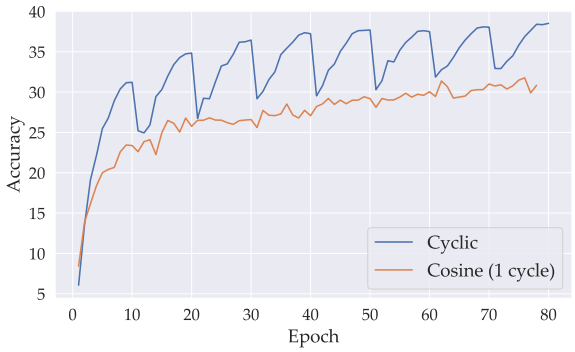


Figure 3: Applying a cyclic learning rate schedule improves performance. Each curve represents the ImageNet zero-shot accuracy of a ResNet-50 CLIP model as a function of the number of training epochs. The orange curve corresponds to the standard training strategy using a cosine LR scheduler, while the blue curve corresponds to training the CLIP model with a cyclic LR Scheduler. Note that applying a cyclic LR improves performance. See Appendix B for additional results.

model [RKH+21] that achieved state-of-the-art results on a range of zero-shot classification tasks, most notably ImageNet and its variants [RRS+19; HZB+19; BMA+19; WGX+19; HBM+20].

Given the remarkable performance of CLIP models, several works have proposed approaches to improve their performance. Some of these works modify the underlying training procedure. For example, FLIP masks at random a subset of the ViT input tokens and drops them [LFH+22]. Other works have proposed using strong and weak data augmentations (for both vision and text backbones) when computing the similarity between an image and a caption [FAR+23].

Another line of work modifies the objective function used for training the CLIP model. For instance, SLIP [MKW+21] combines an image contrastive loss (SimCLR [CKN+20]) with the CLIP objective function [RKH+21]. Other approaches propose clustering similar images and captions ([CWL+22]), imposing geometrical consistency between an image-caption pair embeddings ([GBB+22]), imposing additional supervision ([LLZ+22]) or using Hopfield networks to control the covariance of the learned representations [FRL+22].

Finally, additional works proposed pretraining the vision backbone prior to training the CLIP model [FWX+22; FSW+23; SFW+23], or ensembling several trained CLIP models [IWW+21].

5 Conclusion

In this paper, we have demonstrated that CLIP models trained on smaller datasets might be undertrained. To improve the performance of such CLIP models, we propose a simple additional training procedure, and demonstrate its effectiveness and competitiveness with existing approaches. This suggests that the methods proposed to improve CLIP performance should be tested at a larger scale in order to accurately reflect their potential benefits.

Acknowledgements

Work supported in part by the NSF grants CNS-1815221 and DMS-2134108, and Open Philanthropy. This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR001120C0015.

References

[BMA+19] 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". In: *Neural Information Processing Systems (NeurIPS)*. 2019.

[BMR+20] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. "Language models are few-shot learners". In: *arXiv preprint arXiv:2005.14165* (2020).

[CKN+20] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. "A simple framework for contrastive learning of visual representations". In: *International conference on machine learning*. 2020.

[CSD+21] 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: *Computer Vision and Pattern Recognition*. 2021.

[CWL+22] Delong Chen, Zhao Wu, Fan Liu, Zaiquan Yang, Yixiang Huang, Yiping Bao, and Erjin Zhou. "Prototypical Contrastive Language Image Pretraining". In: *CoRR* (2022).

[DDS+09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. "Imagenet: A large-scale hierarchical image database". In: *Computer Vision and Pattern Recognition (CVPR)*. 2009.

[FAR+23] Enrico Fini, Pietro Astolfi, Adriana Romero-Soriano, Jakob Verbeek, and Michal Drozdzal. "Improved Baselines for Vision-Language Pre-Training". In: *arXiv preprint arXiv:2305.08675*. 2023.

[FRL+22] Andreas Fürst, Elisabeth Rumetshofer, Johannes Lehner, Viet T. Tran, Fei Tang, Hubert Ramsauer, David Kreil, Michael Kopp, Günter Klambauer, Angela Bitto, and Sepp Hochreiter. "CLOOB: Modern Hopfield Networks with InfoLOOB Outperform CLIP". In: *Neural Information Processing Systems (NeurIPS)*. 2022.

[FSW+23] Yuxin Fang, Quan Sun, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. "EVA-02: A Visual Representation for Neon Genesis". In: *arXiv preprint arXiv:2303.11331* (2023).

[FWX+22] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. "EVA: Exploring the Limits of Masked Visual Representation Learning at Scale". In: *arXiv preprint arXiv:2211.07636* (2022).

[GBB+22] Shashank Goel, Hritik Bansal, Sumit Bhatia, Ryan Rossi, Vishwa Vinay, and Aditya Grover. "CyCLIP: Cyclic Contrastive Language-Image Pretraining". In: *Neural Information Processing Systems (NeurIPS)*. 2022.

[HBM+20] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. *The Many Faces of Robustness: A Critical Analysis of Out-of-Distribution Generalization*. 2020. arXiv: 2006.16241 [cs.CV].

[HZB+19] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. "Natural adversarial examples". In: *arXiv preprint arXiv:1907.07174* (2019).

[IWW+21] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. *OpenCLIP*. 2021.

[KSL19] Simon Kornblith, Jonathon Shlens, and Quoc V Le. "Do better imagenet models transfer better?" In: *computer vision and pattern recognition (CVPR)*. 2019.

[LFH+22] Yanghao Li, Haoqi Fan, Ronghang Hu, Christoph Feichtenhofer, and Kaiming He. "Scaling Language-Image Pre-training via Masking". In: *arXiv preprint arXiv:2212.00794*. 2022.

[LH17] Ilya Loshchilov and Frank Hutter. "SGDR: Stochastic Gradient Descent with Warm Restarts". In: *International Conference on Learning Representations (ICLR)*. 2017.

[LLZ+22] Yangguang Li, Feng Liang, Lichen Zhao, Yufeng Cui, Wanli Ouyang, Jing Shao, Fengwei Yu, and Junjie Yan. "Supervision Exists Everywhere: A Data Efficient Contrastive Language-Image Pre-training Paradigm". In: *International Conference on Learning Representations*. 2022.

[MKW+21] Norman Mu, Alexander Kirillov, David Wagner, and Saining Xie. "SLIP: Self-supervision meets Language-Image Pre-training". In: *arXiv preprint arXiv:2112.12750*. 2021.

[RKH+21] 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: *arXiv preprint arXiv:2103.00020*. 2021.

[RRS+19] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. "Do ImageNet Classifiers Generalize to ImageNet?" In: *International Conference on Machine Learning (ICML)*. 2019.

[SBV+22] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. "Laion-5b: An open large-scale dataset for training next generation image-text models". In: *arXiv preprint arXiv:2210.08402*. 2022.

[SDG+18] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. "Conceptual Captions: A Cleaned, Hypernymed, Image Alt-text Dataset For Automatic Image Captioning". In: *Association for Computational Linguistics*. 2018.

[SFW+23] Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. "EVA-CLIP: Improved Training Techniques for CLIP at Scale". In: *arXiv preprint arXiv:2303.15389* (2023).

[SIE+20] Hadi Salman, Andrew Ilyas, Logan Engstrom, Ashish Kapoor, and Aleksander Madry. "Do Adversarially Robust ImageNet Models Transfer Better?" In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2020.

[SVB+21] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. "LAION-400M: Open Dataset of CLIP-Filtered 400 Million Image-Text Pairs". In: *arXiv preprint arXiv:2111.02114* (2021).

[WGX+19] Haohan Wang, Songwei Ge, Eric P Xing, and Zachary C Lipton. "Learning robust global representations by penalizing local predictive power". In: *Neural Information Processing Systems (NeurIPS)* (2019).

[WIK+21] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo-Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. "Robust fine-tuning of zero-shot models". In: *arXiv preprint arXiv:2109.01903*. 2021.

[YWV+22] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. "CoCa: Contrastive Captioners are Image-Text Foundation Models". In: *arXiv preprint arXiv:2205.01917*. 2022.

A Experimental Setup

Pretraining datasets. In our experiments, we train our CLIP models on three datasets of increasing size, namely CC3M [SDG+18], CC12M [CSD+21], and LAION-400M [SBV+22]. Each of these dataset contains image-caption pairs of datapoints which are using to train CLIP model via contrastive learning.

Models. We consider three different models in our experiments: ResNet-50, ViT-B-32, and ViT-B-16. We first pretrain a version of these models from scratch on the above datasets (except for LAION-400M), matching the results of publicly available models on OpenCLIP. For LAION-400M, we use the checkpoint available on OpenCLIP.

Validation datasets. To evaluate the performance of our models, we use several datasets, including ImageNet variations such as ImageNet-V2, ImageNet-A, ImageNet-R, ImageNet-S, and ObjectNet [RRS+19; HZB+19; BMA+19; WGX+19; HBM+20], as well as suite of transfer learning datasets used in [KSL19; SIE+20]. We utilize the CLIP benchmarks repository to evaluate all of our models.

Hyperparameters. When training our CLIP models on CC3M [SDG+18] and CC12M [CSD+21], we use hyperparameters similar to the ones employed in [IWW+21]. Specifically, we use train our models for a total of 75 epochs using a global batch size of 2,560 (256 samples per GPU), a learning rate of 10^-3, and a weight decay of 0.5.

OpenCLIP repository can be found here https://github.com/mlfoundations/open_clip.
CLIP benchmarks can be found here https://github.com/LAION-AI/CLIP_benchmark.

B Additional Results

B.1 How Many Extra Epochs?

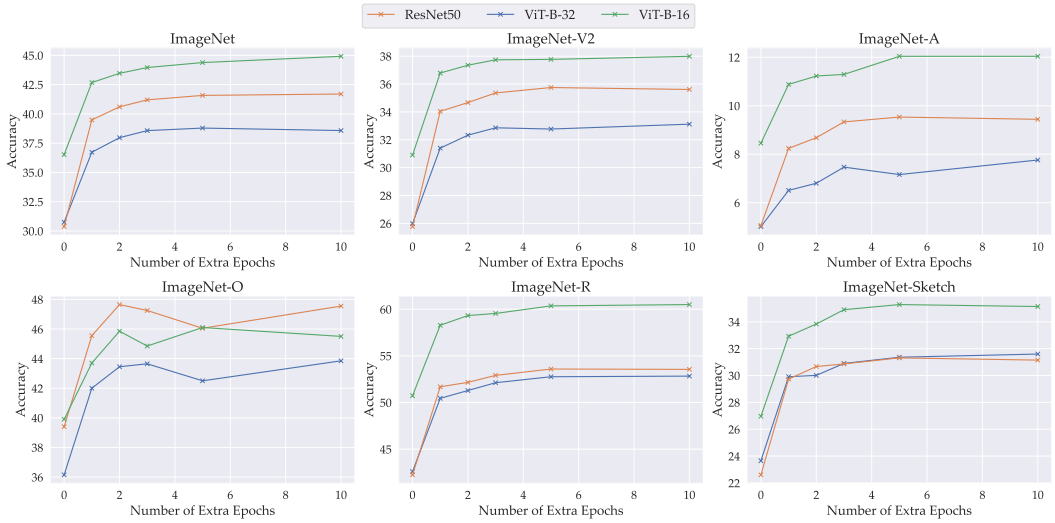


Figure 4: Applying the additional training procedure for few extra epochs is enough to improve performance. The zero-shot accuracy of several CLIP models (y-axis) increases as we apply the additional training procedure for more epochs (x-axis). Note that the performance improvement saturates after applying the procedure for only three additional epochs.

B.2 When to Apply our Strategy?

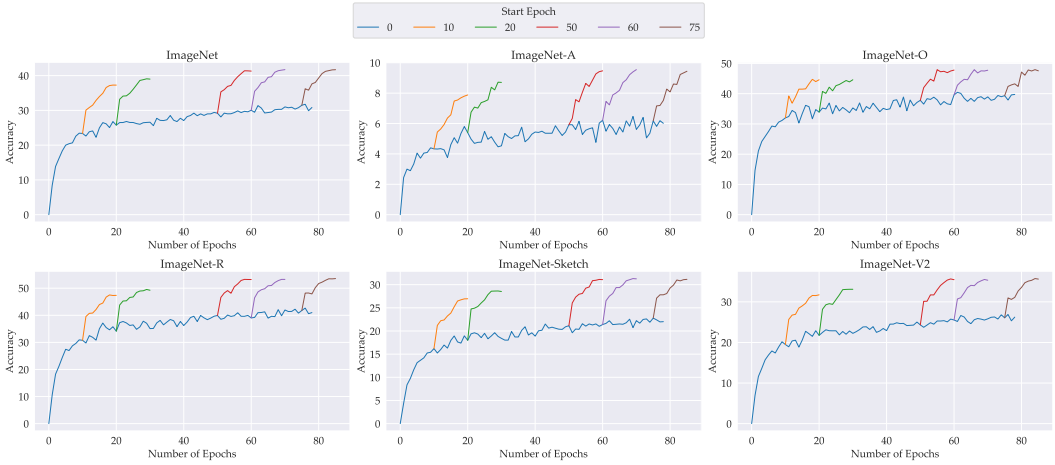


Figure 5: Applying our strategy early during training already improves performance. The blue curve corresponds to the accuracy of the original CLIP model. Each non-blue curve represents the zero-shot accuracy of the CLIP model after applying our strategy with different starting points. For example, the orange curve corresponds to applying our strategy on the CLIP model after it has been trained for 10 epochs (out of 75 epochs in total). Note that applying our strategy earlier during training leads to a performance improvement beyond the final accuracy reached by the model trained for 75 epochs.

Cyclic LR Scheduler

This is a supplementary figure to Figure 3 which shows how cyclic learning rate schedule, instead of a cosine one, can lead to better zero-shot performance for CLIP models trained from scratch.

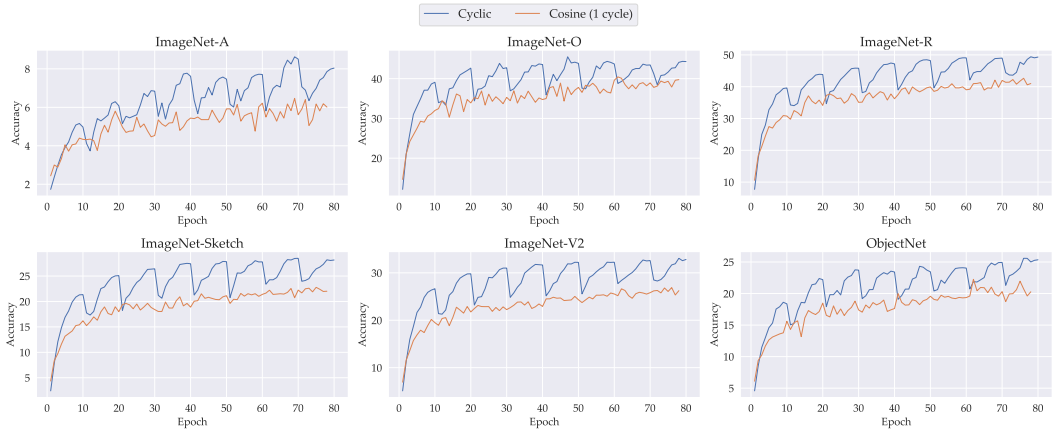


Figure 6: Applying a cyclic learning rate schedule improves performance. Each curve represents the zero-shot accuracy of a ResNet-50 CLIP model as a function of the number of training epochs. The orange curve corresponds to the standard training strategy using a cosine LR scheduler, while the blue curve corresponds to training the CLIP model with a cyclic LR Scheduler. Note that applying a cyclic LR improves performance.

B.3 Additional Zero-Shot Results on Downstream Tasks

Here we show the performance improvements of our models on a suite of transfer learning tasks, and a range of tasks from the CLIP benchmarks repository.

Model	Caltech101	Cars	CIFAR10	CIFAR100	DTD	FGVC Aircraft
ResNet-50	76.4 (+6.03%)	26.2 (+13.3%)	49.4 (+18.5%)	27.5 (+13.0%)	22.1 (+1.86%)	2.67 (+1.05%)
ViT-B-32	77.3 (+3.39%)	19.2 (+7.26%)	81.3 (+9.69%)	43.0 (+2.15%)	21.4 (-0.80%)	2.31 (+0.15%)
ViT-B-16	79.1 (+3.69%)	26.8 (+9.03%)	80.0 (+2.06%)	48.2 (+4.09%)	23.1 (-0.60%)	2.52 (+0.00%)

Table 4: Our simple training procedure consistently improves the performance of CLIP models trained on CC12M. This table shows the zero-shot accuracy of several CLIP models on different downstream tasks after applying our simple strategy. The numbers in parentheses represent the absolute change in zero-shot accuracy on the corresponding downstream classification task.

Model	Flowers	Pets	STL10	SUN397	SVHN
ResNet-50	34.6 (+11.1%)	62.0 (+13.0%)	89.6 (+3.20%)	47.5 (+2.93%)	13.6 (+6.93%)
ViT-B-32	34.0 (+10.6%)	57.8 (+2.80%)	92.0 (+2.47%)	47.3 (+2.47%)	22.6 (+5.42%)
ViT-B-16	37.8 (+14.1%)	64.7 (+12.2%)	93.9 (-0.20%)	48.6 (-0.60%)	19.7 (+2.45%)

Table 5: Our simple training procedure consistently improves the performance of CLIP models trained on CC12M. This table shows the zero-shot accuracy of several CLIP models on different downstream tasks after applying our simple strategy. The numbers in parentheses represent the absolute change in zero-shot accuracy on the corresponding downstream classification task.

CLIP benchmarks can be found here https://github.com/LAION-AI/CLIP_benchmark