MobileCLIP: Fast Image-Text Models through Multi-Modal Reinforced Training

Pavan Kumar Anasosalu Vasu* Hadi Pouransari* Fartash Faghri* Raviteja Vemulapalli
Oncel Tuzel
Apple
{panasosaluvasu,mpouransari,fartash,r_vemulapalli,otuzel}@apple.com

Abstract

Contrastive pretraining of image-text foundation models, such as CLIP, demonstrated excellent zero-shot performance and improved robustness on a wide range of downstream tasks. However, these models utilize large transformer-based encoders with significant memory and latency overhead which pose challenges for deployment on mobile devices. In this work, we introduce MobileCLIP – a new family of efficient image-text models optimized for runtime performance along with a novel and efficient training approach, namely multi-modal reinforced training. The proposed training approach leverages knowledge transfer from an image captioning model and an ensemble of strong CLIP encoders to improve the accuracy of efficient models. Our approach avoids train-time compute overhead by storing the additional knowledge in a reinforced dataset. MobileCLIP sets a new state-of-the-art latency-accuracy tradeoff for zero-shot classification and retrieval tasks on several datasets. Our MobileCLIP-S2 variant is 2.3× faster while more accurate compared to previous best CLIP model based on ViT-B/16. We further demonstrate the effectiveness of our multi-modal reinforced training by training a CLIP model based on ViT-B/16 image backbone and achieving +2.9% average performance improvement on 38 evaluation benchmarks compared to the previous best. Moreover, we show that the proposed approach achieves 10×-1000× improved learning efficiency when compared with non-reinforced CLIP training. Code and models are available at https://github.com/apple/ml-mobileclip

1. Introduction

Large image-text foundation models, such as CLIP [47], have demonstrated excellent zero-shot performance and improved robustness [15] across a wide range of downstream tasks [30]. However, deploying these models on mobile devices is challenging due to their large size and high latency.

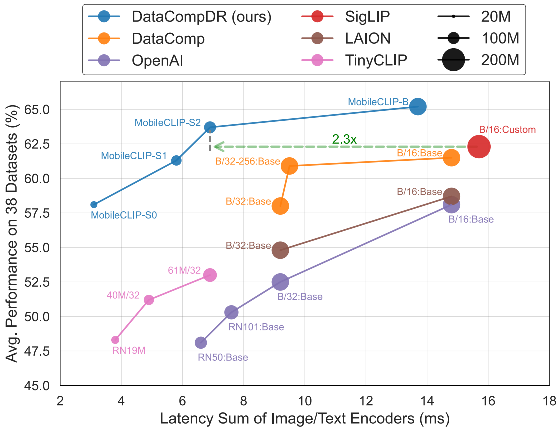


Figure 1. **MobileCLIP models are fast and accurate.** Comparison of publicly available CLIP models with MobileCLIP trained on our DataCompDR dataset. Latency is measured on iPhone12 Pro Max.

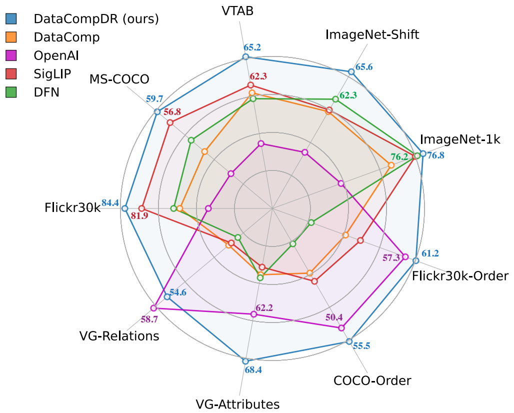


Figure 2. **DataCompDR dataset improves all metrics.** Zero-shot performance of CLIP models with ViT-B/16 image encoder.

Our goal is to design a new family of aligned image-text encoders suitable for mobile devices. There are two main challenges towards realizing this goal. First, there is a tradeoff between runtime performance (e.g., latency) and the accuracy of different architectures, therefore we should be able to quickly and thoroughly analyze different architectural designs. Large-scale training of CLIP models is computation-

[*] Equal contribution.

ally expensive, hindering rapid development and exploration of efficient architecture design. On the other hand, standard multi-modal contrastive learning [47] at small-scale results in poor accuracies, which do not provide a useful signal to guide architecture design choices. Second, reduced capacity of smaller architectures leads to subpar accuracy that can be improved with a better training method.

To overcome these challenges, we develop a novel training approach based on the dataset reinforcement method [14]: i) reinforce a dataset once with additional information, and ii) use the reinforced dataset several times for experimentation. For a given compute budget, training with the reinforced dataset results in improved accuracy compared to the original dataset. We propose a multi-modal variant of dataset reinforcement for training efficient CLIP models. Specifically, we reinforce the image-text DataComp [18] dataset by adding synthetic captions and embeddings from a strong ensemble of pretrained CLIP models (Fig. 3), obtaining DataCompDR. We introduce two variants of our reinforced dataset, DataCompDR-12M suited for rapid iteration on efficient model design and DataCompDR-1B for best large-scale training performance.

Training with DataCompDR shows significant learning efficiency improvement compared to the standard CLIP training. For example, with a single node of 8×A100 GPUs, we achieve 61.7% zero-shot classification on ImageNet-val [8] in approximately one day when training a ViT-B/16 [12] based CLIP from scratch on DataCompDR-12M. Training with DataCompDR-1B sets new state-of-the-art performance on several metrics (Fig. 2) while still using a fraction of the training compute budget compared to previous works.

Utilizing DataCompDR, we explored the design space and obtained a new family of mobile-friendly aligned image-text encoders called MobileCLIP with a better latency-accuracy tradeoff compared to the previous works (Fig. 1). We exploit several architectural design techniques to obtain efficient image and text encoders, including structural reparametrization [9–11, 21, 61] and convolutional token mixing [62]. MobileCLIP includes S0, S1, S2, and B variants covering various sizes and latencies for different mobile applications. Our fastest variant, MobileCLIP-S0, is approximately 5× faster and 3× smaller than the standard OpenAI ViT-B/16 CLIP model [47], but has the same average accuracy. Our contributions are as follows:

- We design a new family of mobile-friendly CLIP models, *MobileCLIP*. Variants of MobileCLIP use hybrid CNN-transformer architectures with structural reparametrization in image and text encoders to reduce the size and latency.
- We introduce multi-modal reinforced training, a novel training strategy that incorporates knowledge transfer from a pre-trained image captioning model and an ensemble of strong CLIP models to improve learning efficiency.
- We introduce two variants of our reinforced datasets: DataCompDR-12M and DataCompDR-1B. Using DataCompDR, we demonstrate 10x-1000x learning efficiency in comparison to DataComp.
- MobileCLIP family obtains state-of-the-art latency-accuracy tradeoff on zero-shot tasks, including marking a new best ViT-B/16 based CLIP model.

2. Related Work

Efficient learning for CLIP. One can improve learning efficiency through utilizing an enhanced training objective. Examples include image masking [17, 37, 55, 71], unimodal self-supervision [35, 43], fine-grained image-text alignment [72], contrastive learning in image-text-label space [69], and pairwise Sigmoid loss [77]. CLIPA [34] proposed training at multi-resolutions for cost-effective training. These methods are complementary to our proposed method.

CLIP training dataset is often comprising noisy image-text pairs obtained at web-scale. Since the original CLIP model [47], several works have demonstrated improved results on large-scale and filtered datasets [16, 18, 51, 52, 77]. Complementary to data collection and filtering, recent works show that using visually enriched synthetic captions generated from a pretrained captioning model, along with real captions, can improve the quality of CLIP models [32, 45, 70]. Our proposed reinforced multi-modal dataset also benefits from synthetically generated captions, which we show are crucial for improved learning efficiency.

Previous works like DIME-FM [56], extends unimodal distillation [26] with a focus on zero-shot classification. TinyCLIP [68] trains compact CLIP models via cross-modal affinity mimicking and weight inheritance. Multi-modal distillation is also explored in setups where the student is a fused vision-language model for specific tasks [31, 64, 65]. Our proposed multi-modal reinforced training also includes cross-modal affinity mimicking [68]. Further, we extend unimodal model ensembling [33, 46] to multimodal setup, and store targets obtained from an ensemble of CLIP models.

Offline knowledge distillation methods [14, 54, 76] have been proposed recently to mitigate the training-time overhead cost due to running large teacher models. We extend the *dataset reinforcement* strategy [14] to the multi-modal setup of CLIP. Our proposed reinforced multi-modal datasets result in significant accuracy improvement without adding a training-time computational overhead.

Efficient architectures for CLIP. Recently there have been a wide range of architectures that have shown great promise for accomplishing vision tasks on resource constraint devices. These architectures can be broadly classified into purely convolutional [11, 23, 27, 28, 41, 48, 50, 61], transformer based [12, 40, 59] and convolution-transformer hybrids like [22, 36, 38, 44, 53, 62]. Similarly there are transformer based [63] and convolution-transformer hybrids like [20, 67] for text encoding. There have been works like [68], that prune ViT architectures to obtain smaller and faster CLIP models or works like [3] that reduce image-text tokens for faster inference of vision-language models. These models can still be quite large and inefficient to be deployed on a mobile device. In our work, we introduce an improved convolution-transformer hybrid architecture for both vision and text modalities, that improve over recent state-of-the-art like [22, 38, 44, 53]. The optimizations introduced in [3, 68] can be used to further improve efficiency of our models.

3. Multi-Modal Reinforced Training

Our multi-modal reinforced training leverages knowledge transfer from an image captioning model and a strong ensemble of pretrained CLIP models for training the target model. It consists of two main components: i) leveraging the knowledge of an image captioning model via synthetic captions, and ii) knowledge distillation of image-text alignments from an ensemble of strong pre-trained CLIP models. We follow the dataset reinforcement strategy of [14] and store the additional knowledge (synthetic captions and teacher embeddings) in the dataset (see Fig. 3), thereby avoiding any additional training time computational overhead such as evaluating the captioning model or the ensemble teacher.

3.1. Dataset Reinforcement

Synthetic captions. Image-text datasets used to train CLIP models are mostly sourced from the web, which is inherently noisy. Recent efforts such as DataComp [18] and data filtering networks [16] improve the quality of web-sourced datasets by using extensive filtering mechanisms. While these filtered datasets have lower noise, the captions may still not be descriptive enough. In order to boost the visual descriptiveness of the captions we use the popular CoCa [74] model and generate multiple synthetic captions $\boldsymbol{x}^{(i,s)}_{\text{syn}}$ for each image $\boldsymbol{x}^{(i)}_{\text{img}}$ (see Fig. 3a). Ablations on the number of synthetic captions generated per image are provided in Sec. 5.1. Figure 5 shows some examples of synthetic captions generated by the CoCa model. Real captions in comparison to synthetic captions are generally more specific but noisier. We show (Tab. 3a) a combination of both real and synthetic captions is crucial to obtain best zero-shot retrieval and classification performance.

Image augmentations. For each image $\boldsymbol{x}^{(i)}_{\text{img}}$, we generate multiple augmented images $\hat{\boldsymbol{x}}^{(i,j)}_{\text{img}}$ using a parametrized augmentation function $\mathcal{A}$:

$$\hat{\boldsymbol{x}}^{(i,j)}_{\text{img}} = \mathcal{A}(\boldsymbol{x}^{(i)}_{\text{img}}; \boldsymbol{a}^{(i,j)}) \,, \tag{1}$$

where $\boldsymbol{a}^{(i,j)}$ are the augmentation parameters that are sufficient to reproduce $\hat{\boldsymbol{x}}^{(i,j)}_{\text{img}}$ from $\boldsymbol{x}^{(i)}_{\text{img}}$ (see Fig. 3a). Ablations on the number and different kinds of augmentations used per image are provided in Tabs. 4a and 13, respectively.

Ensemble teacher. Model ensembling is a widely used technique for creating a stronger model from a set of independently trained ones [33, 46]. We extend this technique to multi-modal setup and use an ensemble of K CLIP models as a strong teacher (see Sec. 5.1 for our teacher ablations). We compute the feature embeddings of these models for augmented images $\hat{\boldsymbol{x}}^{(i,j)}_{\text{img}}$ and synthetic captions $\boldsymbol{x}^{(i,s)}_{\text{syn}}$ obtaining d_k-dimensional vectors $\psi^{(i,j,k)}_{\text{img}}$ and $\psi^{(i,s,k)}_{\text{syn}}$ for the k-th teacher model. We also compute the teacher embeddings $\psi^{(i,k)}_{\text{txt}}$ of the ground-truth captions $\boldsymbol{x}^{(i)}_{\text{txt}}$ (see Fig. 3b).

Reinforced dataset. We store the image augmentation parameters $\boldsymbol{a}^{(i,j)}$, synthetic captions $\boldsymbol{x}^{(i,s)}_{\text{syn}}$, feature embeddings $\psi^{(i,j,k)}_{\text{img}}$, $\psi^{(i,s,k)}_{\text{syn}}$ and $\psi^{(i,k)}_{\text{txt}}$ of the CLIP teachers as additional knowledge in the dataset along with the original image $\boldsymbol{x}^{(i)}_{\text{img}}$ and caption $\boldsymbol{x}^{(i)}_{\text{txt}}$ (see Fig. 3c). Note that dataset reinforcement is a one-time cost that is amortized by several efficient model training and experimentation.

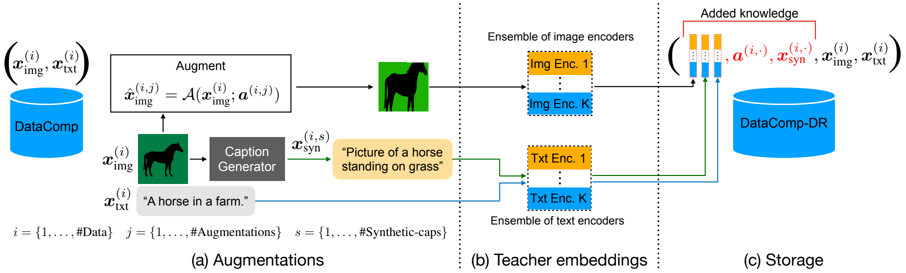
3.2. Training

Loss function. Intuitively, our loss function distills the affinity matrix between image-text pairs from multiple image-text teacher encoders into student image-text encoders. Let $\mathcal{B}$ denote a batch of b (image, text) pairs and $\Psi^{(k)}_{\text{img}}, \Psi^{(k)}_{\text{txt}} \in \mathcal{R}^{b\times d_k}$ the matrices of d_k-dimensional image and text embeddings, respectively, of the k-th model in the teacher ensemble for batch $\mathcal{B}$. Correspondingly, we denote the image and text embedding matrices of the target model by $\Phi_{\text{img}}, \Phi_{\text{txt}} \in \mathcal{R}^{b\times d}$. For given $\boldsymbol{U}$ and $\boldsymbol{V}$ matrices, let $\mathcal{S}_\tau(\boldsymbol{U}, \boldsymbol{V}) \in \mathcal{R}^{b\times b}$ denote their similarity matrix obtained by applying row-wise Softmax operation to $\boldsymbol{U}\boldsymbol{V}^\top/\tau$, where τ is a temperature parameter. Our training loss consists of two components, the standard CLIP [47] loss $\mathcal{L}_{\text{CLIP}}(\mathcal{B})$ and a knowledge distillation loss $\mathcal{L}_{\text{Distill}}(\mathcal{B})$:

$$\begin{aligned}
\mathcal{L}_{\text{Total}}(\mathcal{B}) &= (1-\lambda)\mathcal{L}_{\text{CLIP}}(\mathcal{B}) + \lambda\mathcal{L}_{\text{Distill}}(\mathcal{B}), \qquad (2)\\
\mathcal{L}_{\text{Distill}}(\mathcal{B}) &= \frac{1}{2}\mathcal{L}^{\text{I2T}}_{\text{Distill}}(\mathcal{B}) + \frac{1}{2}\mathcal{L}^{\text{T2I}}_{\text{Distill}}(\mathcal{B}),\\
\mathcal{L}^{\text{I2T}}_{\text{Distill}}(\mathcal{B}) &= \frac{1}{bK}\sum_{k=1}^{K} \text{KL}(\mathcal{S}_{\tau_k}(\Psi^{(k)}_{\text{img}}, \Psi^{(k)}_{\text{txt}}) \| \mathcal{S}_{\widehat{\tau}}(\Phi_{\text{img}}, \Phi_{\text{txt}})),
\end{aligned}$$

where KL denotes Kullback-Leibler divergence, $\mathcal{L}^{\text{T2I}}_{\text{Distill}}$ is computed by swapping the text and image embedding terms of $\mathcal{L}^{\text{I2T}}_{\text{Distill}}$, and λ is a tradeoff parameter.

Efficient training. Training on the reinforced dataset is as simple as modifying the data loader and loss function to exploit additional knowledge stored in the dataset and has the same training cost as standard CLIP training (see


$\left(x_{\text{img}}^{(i)}, x_{\text{txt}}^{(i)}\right)$
DataComp
Augment
$\hat{x}_{\text{img}}^{(i,j)} = \mathcal{A}(x_{\text{img}}^{(i)}; a^{(i,j)})$
$x_{\text{img}}^{(i)}$
Caption Generator
$x_{\text{syn}}^{(i,s)}$
"Picture of a horse standing on grass"
$x_{\text{txt}}^{(i)}$
"A horse in a farm."
$i = \{1, \ldots, \#\text{Data}\}$ $j = \{1, \ldots, \#\text{Augmentations}\}$ $s = \{1, \ldots, \#\text{Synthetic-caps}\}$
Ensemble of image encoders
Img Enc. 1
Img Enc. K
Txt Enc. 1
Txt Enc. K
Ensemble of text encoders
Added knowledge
$\left(, a^{(i,\cdot)}, x_{\text{syn}}^{(i,\cdot)}, x_{\text{img}}^{(i)}, x_{\text{txt}}^{(i)} \right)$
DataComp-DR

(a) Augmentations (b) Teacher embeddings (c) Storage

Figure 3. Illustration of multi-modal dataset reinforcement with one image augmentation and one synthetic caption. In practice, we use multiple image augmentations and synthetic captions.

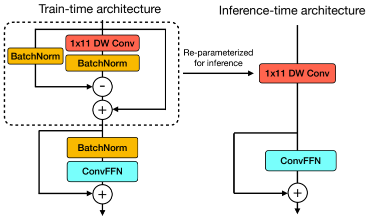


Figure 4. Architecture of convolutional and reparameterizable blocks, called Text-RepMixer used in MobileCLIP's text encoder MCt.

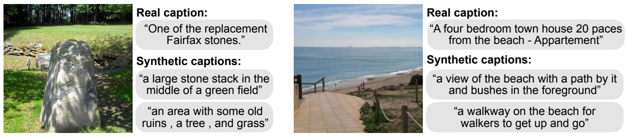

Real caption:
"One of the replacement Fairfax stones."
Synthetic captions:
"a large stone stack in the middle of a green field"
"an area with some old ruins , a tree , and grass"
Real caption:
"A four bedroom town house 20 paces from the beach - Appartement"
Synthetic captions:
"a view of the beach with a path by it and bushes in the foreground"
"a walkway on the beach for walkers to get up and go"

Figure 5. Real vs synthetic captions.

Tab. 4d). For every sample, we read the image $x_{\text{img}}^{(i)}$ and the corresponding ground-truth caption $x_{\text{txt}}^{(i)}$ from the dataset. Then, we randomly load one of stored augmentation parameters $a^{(i,j)}$ and reproduce the augmented image $\hat{x}_{\text{img}}^{(i,j)}$. We also randomly load one of synthetic captions $x_{\text{syn}}^{(i,s)}$. Finally, we read the stored embeddings, $\psi_{\text{img}}^{(i,j,k)}$, $\psi_{\text{syn}}^{(i,s,k)}$, and $\psi_{\text{txt}}^{(i,k)}$, corresponding to the K teacher models.

Using this loaded data, we construct two data batches, $\mathcal{B}_{\text{real}}$ corresponding to (augmented image, real caption) pairs and $\mathcal{B}_{\text{syn}}$ corresponding to (augmented image, synthetic caption) pairs, and compute our training loss in Eq. (2) separately on $\mathcal{B}_{\text{real}}$ and $\mathcal{B}_{\text{syn}}$. Our final loss is given by

$$\sum_{\mathcal{B} \in \{\mathcal{B}_{\text{real}}, \mathcal{B}_{\text{syn}}\}} \mathcal{L}_{\text{Total}}(\mathcal{B}). \tag{3}$$

Note that we can compute the total loss after a forward pass of the student model without any extra teacher-related computations since the teacher embeddings required to compute the distillation loss are readily available as part of the dataset.

4. Architecture

4.1. Text Encoder

CLIP [47] model paired the vision transformer with a classical transformer comprising of self-attention layers for text encoding. While this model is effective, smaller and more efficient models are preferred for mobile deployment. Recently, works like [67] have shown that convolutions can be just as effective for text encoding. In contrast, we found that purely convolutional architectures significantly underperform their transformer counterparts. Instead of using a fully convolutional architecture for text encoding, we introduce a hybrid text encoder which makes use of 1-D convolutions and self-attention layers.

For hybrid text encoder, we introduce *Text-RepMixer*, a convolutional token mixer that decouples train-time and inference-time architectures. Text-RepMixer is inspired by reparameterizable convolutional token mixing (RepMixer) introduced in [62]. At inference, skip connections are reparameterized. The architecture is shown in Fig. 4. For Feed-Forward Network (FFN) blocks, we augment linear layers with an additional depthwise 1-D convolution of similar kernel dimensions as the token mixer, to obtain *ConvFFN* blocks. This structure is similar to the convolutional blocks used in [20], the main difference being the use of batchnorm and the ability to fold it with the succeeding depthwise 1-D convolutional layer for efficient inference. The design choices for *Text-RepMixer* is discussed in Appx. F. In order to find the optimal design for our hybrid text encoder, we start with a purely convolutional text encoder and start replacing convolutional blocks systematically with self-attention layers (see Tab. 5). Tab. 1, show the efficacy of our text encoder when compared with CLIP's base text encoder. Our model is smaller, faster and obtains similar performance as the larger base text encoder when paired with efficient backbones like ViT-S/16.

4.2. Image Encoder

Recent works have shown the efficacy of hybrid vision transformer for learning good visual representations. For MobileCLIP, we introduce an improved hybrid vision transformer called MCi based on the recent FastViT [62] architecture with certain key differences explained below.

In FastViT, an MLP expansion ratio of 4.0 is used for FFN blocks. Recent works like [39, 68] exposed the significant amount of redundancy in linear layers of FFN block. To improve parameter efficiency, we simply lower the expansion ratio to 3.0 and increase the depth of the architecture. By doing so, we retain the same number of parameters in the image encoder. The stage configuration for the three variants are described in Appx. A. MCi0 has similar stage configuration as [61]. MCi1, is a deeper version of MCi0 and MCi2 is a wider version of MCi1. The stage compute ratios in our variants are similar to [61]. We find that this design has a minimal impact on latency, but a good improvement in capacity of the model, reflected in the downstream task performance, see Appx. B. In Tab. 1, we compare our MCi encoder with a similar sized FastViT-MA36 when used as image encoders in a CLIP model. Our model obtains much better zero-shot IN-val performance while being 16.3% faster.

Text Enc.	Latency (txt)	0-shot IN-val
Base	3.3	53.4
MCt (Ours)	**1.6**	**53.6**

Image Enc.	Latency (img)	0-shot IN-val
FastViT-MA36	4.3	58.9
MCi2 (Ours)	**3.6**	**60.0**

Table 1. **(a) Base vs. MCt** text encoders with ViT-S/16. **(b) FastViT vs. MCi** image encoders with Base text encoder. Trained for 30k iters (∼0.24B seen samples) on DataCompDR-12M.

5. Experiments

In this section, we present our experimental setup and results.

Evaluation. We evaluate image-text models using the evaluation benchmark of DataComp [18]. Specifically, we report zero-shot classification on the ImageNet validation set [8], and its distribution shifts including ImageNet-V2 [49], ImageNet-A [25], ImageNet-O [25], ImageNet-R [24], and ObjectNet [1], which we report their average as IN-Shift. For zero-shot image-text retrieval, we report recall@1 on MSCOCO [5] and Flickr30k [73] datasets. Further, we report average performance on all 38 datasets in DataComp evaluations. We also evaluate our models on Visual Genome Relation, Visual Genome Attributes, Flickr30k-Order and COCO-Order datasets which are part of the recent Attribute, Relation and Order (ARO) benchmark [75]. In the remainder, IN-val refers to zero-shot accuracy on ImageNet validation set and Flickr30k refers to average zero-shot recall@1 for image-text and text-image retrieval. All reported metrics are obtained without any fine-tuning.

Training setup. We have two setups for ablations and large-scale experiments. For ablations, we train on datasets with 12.8M image-text pairs using a global batch size of 8,192 and 8×NVIDIA-A100-80GB GPUs for 30-45k iterations. For large-scale training, we use a global batch size of 65,536 with 256×A100 GPUs for 200k iterations. All models are trained from scratch (see details in Appx. B).

Dataset. We train on the image-text dataset of DataComp dataset [18]. We use the Bestpool filtered subset of 1.28B samples that provides them with best performance at the largest dataset scale. We refer to this set as DataComp-1B. For fast experimentation, we create a fixed subset of 12.8M uniformly sampled pairs which we call DataComp-12M. DataComp-12M was not studied in [18] but in our experiments, we observed that DataComp-12M consistently achieves better performance compared with the Bestpool subset of DataComp-medium with comparable samples.

λ	Syn. Captions	Strong Aug.	Ens. Teacher	IN-val	Flickr30k
0	✗	✗	✗	44.5	41.8
0	✓	✗	✗	51.9	69.3
1	✓	✗	✗	54.5	66.1
1	✓	✓	✗	59.3	70.5
1	✓	✓	✓	61.7	72.0
0.7	✓	✓	✓	60.7	74.2

Table 2. **Summary of ablations.** We train on DataCompDR-12M for 30k iterations. All ablations are on ViT-B/16:Base. We highlight our main choices with blue and alternative tradeoffs with gray. We underline numbers within 0.5% of the maximum.

DataCompDR: Reinforced DataComp. We reinforce the DataComp dataset using our multi-modal dataset reinforcement strategy. In particular, we create DataCompDR-1B and DataCompDR-12M by reinforcing DataComp-1B and DataCompDR-12M. We have a one-time generation process, the cost of which is amortized over multiple architectures and extensive ablations. We generate 5 synthetic captions per image using the `coca_ViT-L-14` model in OpenCLIP [29], and strong random image augmentations (10 for DataCompDR-1B and 30 for DataCompDR-12M). We compute embeddings of an ensemble of two strong teachers (`ViT-L-14` with pretrained weights `datacomp_xl_s13b_b90k` and `openai` in OpenCLIP) on augmented images as well as real and synthetic captions. Embeddings are 1536-D concatenations of 2×768-D vectors. We store all reinforcements using lossless compression and BFloat16. We analyze all of our choices in Sec. 5.1. One seen sample for DataCompDR is a triplet of one randomly augmented image, one ground-truth caption, and one randomly picked synthetic caption.

MobileCLIP architectures. Our MobileCLIP architectures are formed as pairs of MCi:MCt architectures. In particular, we create 3 small variants MobileCLIP-S0 (MCi0:MCt), MobileCLIP-S1 (MCi1:Base), and MobileCLIP-S2 (MCi2:Base), where Base is a 12-layer Transformer similar to the text-encoder of ViT-B/16 based CLIP [47]. We also train a standard pair of ViT-B/16:Base and refer to our trained model as MobileCLIP-B.

Benchmarking latency. To measure latency, we use the input sizes corresponding to the respective methods. For iPhone latency measurements, we export the models using Core ML Tools (v7.0) [58] and run it on iPhone12 Pro Max with iOS 17.0.3. Batch size is set to 1 for all the models. We follow the same protocol as described in [61].

5.1. Ablation Studies

In this section, we analyze the effect of each component in our training and architecture. Unless otherwise stated, we use ViT-B/16:Base encoders trained on DataComp-12M for 30k iterations with global batch size of 8k (∼20 epochs). Table 2 summarizes the analysis of our training.

$\mathcal{B} \in$	$\{\mathcal{B}_{\text{real}}\}$	$\{\mathcal{B}_{\text{syn}}\}$	$\{\mathcal{B}_{\text{real}}$ or $\mathcal{B}_{\text{syn}}\}$	$\{\mathcal{B}_{\text{real}}, \mathcal{B}_{\text{syn}}\}$
IN-val	56.4	49.8	57.3	61.7
Flickr30k	57.0	72.2	68.6	72.0

(a) Real vs synthetic sampling in Eq. (3) ($\lambda = 1.0$).

λ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
IN-val	54.4	56.3	57.4	58.2	59.5	60.3	60.7	61.5	61.6	61.7
Flickr30k	71.4	71.5	71.8	72.2	73.8	73.6	74.2	73.1	73.2	72.0

(b) Ablation on the loss coefficient (λ) in Eq. (2).

Table 3. **Ablation on the loss.** The tradeoff between IN-val and Flickr30k is controlled by the synthetic sampling and loss coefficient. We train for 30k iterations.

Strong image augmentations. In contrast to uni-modal supervised and self-supervised methods for vision with strong augmentations [13, 60], CLIP training recipes [47] often use light image augmentations to avoid image-text misalignment. However, several works [2, 14, 46] demonstrated the efficacy of strong augmentations in a distillation setup. In Tab. 2 we show that strong image augmentations improve distillation performance (+4.8% on IN-val and +4.4% on Flickr30k). We provide detailed ablation on the effect of image augmentations in Appx. C.

Synthetic captions. Similar to image augmentations, synthetic captions (or caption augmentations) can further improve the performance of CLIP models, particularly on image-text retrieval. For regular CLIP training ($\lambda = 0$), we observe in Tab. 2 that including batches with both synthetic and real captions results in +7.4% on IN-val and +27.5% on Flickr30k performance improvements. In Tab. 3a, we observe a similar trend for CLIP training with distillation loss only ($\lambda = 1$). In Tab. 3b, we analyze the effect of λ and observe a tradeoff where $\lambda = 1.0$ is optimal for IN-val while $\lambda = 0.7$ is optimal for Flickr30k. Prior work that exploit synthetic captions primarily focus on improved retrieval [32, 70] while distillation works focus on zero-shot classification [56]. In our large-scale experiments, we balance the tradeoff for MobileCLIP-B using $\lambda = 0.75$ and use $\lambda = 1.0$ for our small variants.

Ensemble teacher. We find that using an ensemble of strong CLIP models as a teacher in our multi-modal reinforced training is crucial to achieving +2.4% IN-val improvement (Tab. 2). We also observe that the most accurate models are not the best teachers. See Appx. D for a comprehensive analysis of different teacher models.

Number of image augmentations and synthetic captions. We generate multiple image augmentations and synthetic captions and store them efficiently along with the teacher embeddings. We investigate the effectiveness of the number of augmentations and synthetic captions in Tabs. 4a and 4b. We train models with up to 30 image augmentations and 5 synthetic captions for 45k iterations (∼30 epochs). We observe that the performance nearly saturates at 5 augmentations and 2 synthetic captions suggesting each augmentation can be reused multiple times before the added knowledge is fully learned by the model. When needed, fewer augmentations and synthetic captions can help reduce the generation time and storage overhead. For maximal performance, we reinforce DataCompDR-12M and DataCompDR-1B with 10 and 30 augmentations, respectively, and 5 synthetic captions.

Num. Aug.	1	2	5	10	15	20	25	30
IN-val	60.63	63.27	64.81	64.74	64.49	64.92	64.78	64.74
Flickr30k	69.61	71.74	74.76	74.46	73.90	74.29	73.27	75.66

(a) Effect of the number of augmentations.

Num. Caps.	0	1	2	3	4	5
IN-val	60.67	64.88	65.19	65.19	64.81	64.74
Flickr30k	62.26	73.82	74.27	73.91	74.07	75.66

(b) Effect of the number of synthetic captions.

Dataset	Image	Text	Syn.	Aug. Params	Text Emb.	Image Emb.	Size (TBs)
DataComp-12M	✓	✓	✗	✗	✗	✗	0.9
	✓	✓	✓	✓	✗	✗	0.9
DataCompDR-12M	✓	✓	✓	✓	5+1	30	1.9
DataComp-1B	✓	✓	✗	✗	✗	✗	90
DataCompDR-1B	✓	✓	✓	✓	5+1	10	140

(c) Total storage for samples stored in individual Pickle Gzip files and BFloat16 embeddings. +1 refers to the ground-truth caption. For further size reductions see Tab. 16.

Dataset	$\mathcal{B} \in$	$\mathcal{L}_{\text{CLIP}}$	$\mathcal{L}_{\text{Distill}}$	Stored Syn. Caption	Stored Embeddings	Time (hours)
DataComp-12M	$\{\mathcal{B}_{\text{real}}\}$	✓	✗	✗	✗	1.3
-	$\{\mathcal{B}_{\text{real}}, \mathcal{B}_{\text{syn}}\}$	✓	✓	✗	✗	21.1
-	$\{\mathcal{B}_{\text{real}}, \mathcal{B}_{\text{syn}}\}$	✓	✓	✓	✗	4.1
DataCompDR-12M	$\{\mathcal{B}_{\text{real}}, \mathcal{B}_{\text{syn}}\}$	✓	✓	✓	✓	1.3

(d) Training time per epoch (12.8M samples) on 8×A100-80GB.

Table 4. **Ablations on storage/cost.** Training on DataCompDR has no time overhead. We train for 45k iterations (∼30 epochs).

Num. Self-attn.	6	4	2	1	0
Num Params. (M)	44.5	42.4	40.4	39.3	38.3
Latency (ms)	1.9	1.6	1.4	1.3	1.2
IN-val	60.9	60.8	60.2	60.0	57.9

Table 5. **Ablation on architecture.** Effect of the number of self-attention layers in MCt. We train for 30k iterations.

Training time. A major advantage of reinforced training is the minimal time difference with non-reinforced training. We provide the wall-clock times in Tab. 4d for regular CLIP training as well as training with online distillation and a caption generator. We measure the time for training on one epoch of DataCompDR-12M on a single node with 8× A100-80GB GPUs. An epoch takes 1562 iterations with global batch size 8192 on DataCompDR-12M. Without any dataset reinforcement, training is 16× slower while with partial reinforcements of synthetic captions it is 3× slower.

Storage size. We report the storage requirements for our reinforced datasets compared with the original DataComp dataset. We report the storage size of one file per image-text pair. If present, we store all corresponding reinforcements in the same file. We store files in the Pickle format and compress each file with Gzip compression. The image-text embeddings are saved in BFloat16. We report the total storage size for 12.8M samples of DataCompDR-12M and 1.28B samples of DataCompDR-1B in Tab. 4c. We provide analysis on additional size reductions in Appx. E and verify that using BFloat16 does not impact the accuracy. For minimal storage overhead, we recommend 5 augmentations/synthetic captions for 30 epochs on DataCompDR-12M and 2 for 10 epochs on DataCompDR-1B which are based on our ablations in Tabs. 4a and 4b.

Hybrid text encoder. We ablate over the number of Text-RepMixer blocks that can effectively replace self-attention layers with negligible impact on zero-shot performance. For this ablation, we choose a 6-layer purely convolutional text encoder and systematically introduce self-attention layers in the middle. From Tab. 5, we find that even introducing a single self-attention layer substantially improves the zero-shot performance. The best tradeoff is with 2 blocks of Text-RepMixer and 4 blocks of self-attention layers. This variant, MCt, obtains similar performance as the pure transformer variant, while being 5% smaller and 15.8% faster.

5.2. Small Scale Regime

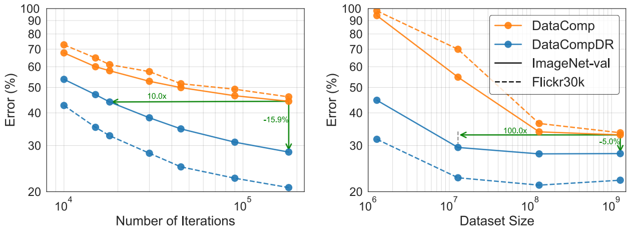
In Tab. 6, we compare methods trained on datasets with 12-20M samples, a relatively small range for fast exploration (e.g., architecture search). MobileCLIP-B trained on DataCompDR-12M with less than 370M samples significantly outperforms all other methods with up to 4× longer training. Also MobileCLIP-B shows great scaling with number of seen samples (65.3→71.7%) in comparison to previous work SLIP [43](42.8→45.0%). In comparison to CLIPA [34] which uses multi-resolution training for efficiency, training with DataCompDR-12M is more efficient: CLIPA obtains 63.2% with 2.69B multi-resolution seen samples (which has equivalent compute as ∼0.5B 224^2 seen samples), that is worse than MobileCLIP-B's 65.3% with only 0.37B seen samples. Further, TinyCLIP-39M/16 in comparison to MobileCLIP-S2 has higher latency and less accuracy, and TinyCLIP-8M/16 is significantly less accurate than MobileCLIP-S0 (41.1% vs 59.1%) while having a close latency (2.6 ms vs 3.1 ms).

5.3. Learning Efficiency

Training longer with knowledge distillation is known to consistently improve performance for classification models [2]. In Fig. 6a we show our reinforced training also benefits from longer training, achieving 71.7% ImageNet-val zero-shot accuracy after 120 epochs using only a 12M subset of DataComp-1B. In comparison, non-reinforced training at best reaches 55.7% accuracy.

Name	**Dataset**	**Seen Samples**	**Latency (ms) (img+txt)**	**Zero-shot IN-val**
CLIP-B/16 [43, 47]	CC-12M [4]	0.39B	11.5 + 3.3	36.5
CLIP-B/16 [43, 47]	YFCC-15M [57]	0.37B		37.6
MobileCLIP-B	CC-12M [4]	0.37B	10.4 + 3.3	38.1
SLIP-B/16 [43]	CC-12M [4]	0.39B	11.5 + 3.3	40.7
SLIP-B/16 [43]	YFCC-15M [57]	0.37B		42.8
MobileCLIP-B	DataComp-12M [18]	0.37B	10.4 + 3.3	50.1
MobileCLIP-B	DataCompDR-12M	0.37B	10.4 + 3.3	**65.3**
CLIP-B/32 [7, 47]	YFCC-15M [57]	0.49B	5.9 + 3.3	32.8
SLIP-B/32 [7, 43]				34.3
FILIP-B/32 [7, 72]				39.5
DeCLIP-B/32 [35]				43.2
DeFILIP-B/32 [7]				45.0
RILS-B/16 [71]	LAION-20M [51]	0.5B	11.5 + 3.3	45.0
TinyCLIP-8M/16 [68]	YFCC-15M [57]	0.75B	**2.0 + 0.6**	41.1
SLIP-B/16 [43]	YFCC-15M [57]	0.75B	11.5 + 3.3	44.1
CLIP-B/16	DataComp-12M [18]	0.74B	10.4 + 3.3	53.5
MobileCLIP-S0	DataCompDR-12M	0.74B	**1.5 + 1.6**	**59.1**
TinyCLIP-39M/16 [68]	YFCC-15M [57]	0.75B	5.2 + 1.9	63.5
MobileCLIP-S2	DataCompDR-12M	0.74B	**3.6 + 3.3**	**64.6**
MobileCLIP-B	DataCompDR-12M	0.74B	10.4 + 3.3	**69.1**
SLIP-B/16 [43]	YFCC-15M [57]	1.5B	11.5 + 3.3	45.0
CLIP-B/16	DataComp-12M [18]	1.48B	10.4 + 3.3	55.7
MobileCLIP-B	DataCompDR-12M	1.48B	10.4 + 3.3	**71.7**
CLIPA-B/16 [34]	LAION-400M [51]	2.69B†	11.5 + 3.3	63.2

Table 6. **Small-scale CLIP training.** MobileCLIP-B notation refers to our re-implementation of ViT-B/16 image encoder and standard Base text encoder. † refers to multi-resolutions. Models are grouped based on the number of samples seen.



(a) Iteration Scaling

(b) Data Scaling

Figure 6. **Learning efficiency up to 1000×.** Training on DataCompDR is 10× more iteration efficient and 100× more data efficient on ImageNet-val and 18× and 1000× more efficient on Flickr30k compared with non-reinforced training.

We also demonstrate scaling with dataset size in Fig. 6b, where we deploy subsets of DataComp-1B from 1.28M to all 1.28B samples. For all experiments we train for 20k iterations with global batch size of 65k (equivalent to one epoch training on 1.28B subset). Training on DataCompDR reaches above 55.2% accuracy with 1.28M samples while training on DataComp-1B gets only to ∼6% accuracy. In this setup, we observe more than 100× data efficiency using DataCompDR. Moreover, we observe 1000× data efficiency for performance on Flickr30k.

5.4. Comparison with State-of-the-art

In Tab. 7, we compare with methods with large scale training. MobileCLIP-S0, trained on DataCompDR-1B significantly outperforms recent works like TinyCLIP [68], and has similar performance as a ViT-B/32 model trained on DataComp [18] while being 2.8× smaller and 3× faster.

Name	Dataset	Seen Samples	Image Encoder	Text Encoder	Params (M) (img+txt)	Latency (ms) (img+txt)	Zero-shot CLS		Flickr30k Ret.		COCO Ret.		Avg. Perf. on 38
							IN-val	IN-shift	T→I	I→T	T→I	I→T	
Ensemble Teacher	DataComp-1B [18] OpenAI-400M [47]	-	ViT-L/14 ViT-L/14	Base Base	(-)	(-)	80.1	69.6	74.5	92.3	46.7	66.5	67.3
TinyCLIP-RN19M [68]	LAION-400M [51]	15.2B	ResNet-19M	Custom	18.6 + 44.8	1.9 + 1.9	56.3	43.6	58.0	75.4	30.9	47.8	48.3
TinyCLIP-RN30M [68]	LAION-400M [51]	15.2B	ResNet-30M	Custom	29.6 + 54.2	2.6 + 2.6	59.1	45.7	61.5	80.1	33.8	51.6	50.2
TinyCLIP-40M/32 [68]	LAION-400M [51]	15.2B	ViT-40M/32	Custom	39.7 + 44.5	3.0 + 1.9	59.8	46.5	59.1	76.1	33.5	48.7	51.2
MobileCLIP-S0	DataCompDR-1B	13B	MCi0	MCt	11.4 + 42.4	1.5 + 1.6	**67.8**	**55.1**	**67.7**	**85.9**	**40.4**	**58.7**	**58.1**
OpenAI-RN50	OpenAI-400M [47]	13B	ResNet-50	Base	38.3 + 63.4	3.3 + 3.3	59.8	45.1	57.4	80.0	28.5	48.8	48.1
TinyCLIP-61M/32 [68]	LAION-400M [51]	15.2B	ViT-61M/32	Custom	61.4 + 54.0	4.3 + 2.6	62.4	48.7	62.6	78.7	36.5	52.8	53.0
TinyCLIP-63M/32 [68]	LAION-400M [51] YFCC-15M [57]	15.8B	ViT-63M/32	Custom	(-)	(-)	64.5	(-)	66.0	84.9	38.5	56.9	(-)
MobileCLIP-S1	DataCompDR-1B	13B	MCi1	Base	21.5 + 63.4	2.5 + 3.3	**72.6**	**60.7**	**71.0**	**89.2**	**44.0**	**62.2**	**61.3**
OpenAI-RN101	OpenAI-400M [47]	13B	ResNet-101	Base	56.3 + 63.4	4.3 + 3.3	62.3	48.5	58.0	79.0	30.7	49.8	50.3
OpenAI-B/32	OpenAI-400M [47]	13B	ViT-B/32	Base	86.2 + 63.4	5.9 + 3.3	63.3	48.5	58.8	78.9	30.4	50.1	52.5
LAION-B/32	LAION-2B [52]	32B					65.7	51.9	66.4	84.4	39.1	56.2	54.8
DataComp-B/32	DataComp-1B [18]	13B					69.2	55.2	61.1	79.0	37.1	53.5	58.0
DataComp-B/32-256	DataComp-1B [18]	34B	ViT-B/32-256	Base	86.2 + 63.4	6.2 + 3.3	72.8	58.7	64.9	84.8	39.9	57.9	60.9
MobileCLIP-S2	DataCompDR-1B	13B	MCi2	Base	35.7 + 63.4	3.6 + 3.3	**74.4**	**63.1**	**73.4**	**90.3**	**45.4**	**63.4**	**63.7**
VeCLIP-B/16 [32]	WIT-200M	6.4B	ViT-B/16	Base	86.2 + 63.4	11.5 + 3.3	64.6	(-)	76.3	91.1	48.4	67.2	(-)
OpenAI-B/16	WIT-400M [47]	13B		Base	86.2 + 63.4	11.5 + 3.3	68.3	55.9	67.7	85.9	40.4	58.7	58.1
LAION-B/16	LAION-2B [52]	34B		Base	86.2 + 63.4	11.5 + 3.3	70.2	56.6	69.8	86.3	42.3	59.4	58.7
EVA02-B/16	Merged-2B [55]	8B		Base	86.2 + 63.4	(-)	74.7	59.6	71.5	86.0	42.2	58.7	58.9
DFN-B/16	DFN-2B [16]	13B		Base	86.2 + 63.4	11.5 + 3.3	76.2	62.3	69.1	85.4	43.4	60.4	60.9
DataComp-B/16	DataComp-1B [18]	13B		Base	86.2 + 63.4	11.5 + 3.3	73.5	60.8	69.8	86.3	42.3	59.4	61.5
SigLIP-B/16 [77]	Webli-1B	40B		Custom	92.9 + 110.3	9.9 + 5.8	76.0	61.0	74.7	89.1	47.8	65.7	62.3
MobileCLIP-B	DataCompDR-1B	13B		Base	86.3 + 63.4	10.4 + 3.3	76.8	65.6	**77.3**	91.4	**50.6**	**68.8**	65.2
MobileCLIP-B (LT)	DataCompDR-1B	39B		Base	86.3 + 63.4	10.4 + 3.3	**77.2**	**66.1**	76.9	**92.3**	50.0	**68.7**	**65.8**

Table 7. **MobileCLIP family of models has the best average performance at various latencies.** Retrieval performances are reported @1. Last column shows average performance on 38 datasets as in OpenCLIP [29]. Models are grouped by their total latency in increasing order and by performance within each group. "Base" refers to standard CLIP Transformer-based [63] text encoder with 12 layers, and "Custom" stands for customized text encoder used in the respective method. For TinyCLIP-63M/32 and EVA02-B/16, we were unable to reliably benchmark models. *Note*: EVA02-B/16 [55] uses MIM pretrained weights for its vision encoder and OpenCLIP-B pretrained weights for its text encoder. TinyCLIP models use advanced weight initialization methods utilizing OpenCLIP models trained on LAION-2B[52] dataset. All other models, including ours are trained from scratch. "(LT)" refers to longer training schedule, described in detail in Appx. I.

MobileCLIP-S2 obtains 2.8% better average performance on 38 datasets and significantly better retrieval performance when compared to ViT-B/32-256 model trained 2.6× longer on DataComp [18]. MobileCLIP-S2 is 1.5× smaller and 1.4× faster than ViT-B/32-256 model. MobileCLIP-B obtains 2.9% better average performance on 38 datasets and better retrieval performance while being 26.3% smaller than SigLIP-B/16 [77] model, which is trained approximately 3× longer on WebLI dataset.

5.5. Retrieval Performance Analysis

We evaluate our models on the recent Attribute, Relation and Order (ARO) benchmark [75]. We compare our MobileCLIP-B trained on DataCompDR-1B with all the publicly available ViT-B/16:Base models in Tab. 8. Optimizing solely for zero-shot classification or retrieval using noisy webscale datasets can degrade the compositional understanding of natural scenes. DataCompDR largely improves the models performance on ARO benchmark while obtaining good performance on zero-shot classification and retrieval tasks. Compared to the recent SigLIP method [77], MobileCLIP-B obtains 19.5% and 12.4% better accuracy on Visual Genome Relation and Attributes datasets and achieves improved recall@1 on Flickr30k-Order and COCO-Order datasets by 69.7% and 50.3%, respectively.

Method	Dataset	IN-val zero-shot	VG Rel.	VG Attr.	COCO Order	Flickr30k Order
CLIP	OpenAI-400M [47]	68.3	**58.7**	62.2	50.4	57.3
CLIP	LAION-2B [52]	70.2	39.7	62.3	31.0	37.5
CLIP	DataComp-1B [18]	73.5	35.9	57.0	29.6	35.2
SigLIP [77]	Webli-1B	76.0	35.1	56.0	32.7	40.7
CLIP	DFN-2B [16]	76.2	33.1	57.4	18.5	22.5
MobileCLIP-B	DataCompDR-1B	**76.8**	54.6	**68.4**	**55.5**	**61.2**

Table 8. **Performance on ARO benchmark.** All the models use ViT-B/16 as image encoder and the Base text encoder. For VG Rel. and VG Attr. datasets, Macro Acc. is reported and for Flickr30k-Order and COCO-Order recall@1 is reported following [75].

6. Conclusion

In this work we introduced MobileCLIP aligned image-text backbones, designed for on-device CLIP inference (low latency and size). We also introduced DataCompDR, a reinforcement of DataComp with knowledge from a pre-trained image captioning model and an ensemble of strong CLIP models. We demonstrated 10×-1000× learning efficiency with our reinforced dataset. MobileCLIP models trained on DataCompDR obtain state-of-the-art latency-accuracy tradeoff when compared to previous works. MobileCLIP models also exhibit better robustness and improved performance on Attribute, Relation and Order (ARO) benchmark.

References

[1] Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. *Advances in neural information processing systems*, 32, 2019. 5

[2] Lucas Beyer, Xiaohua Zhai, Amélie Royer, Larisa Markeeva, Rohan Anil, and Alexander Kolesnikov. Knowledge distillation: A good teacher is patient and consistent. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10925–10934, 2022. 6, 7

[3] Qingqing Cao, Bhargavi Paranjape, and Hannaneh Hajishirzi. PuMer: Pruning and merging tokens for efficient vision language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023. 3

[4] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12m: Pushing web-scale image-text pretraining to recognize long-tail visual concepts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3558–3568, 2021. 7

[5] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015. 5

[6] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020. 14

[7] Yufeng Cui, Lichen Zhao, Feng Liang, Yangguang Li, and Jing Shao. Democratizing contrastive language-image pretraining: A clip benchmark of data, model, and supervision. *arXiv preprint arXiv:2203.05796*, 2022. 7

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 2, 5

[9] Xiaohan Ding, Yuchen Guo, Guiguang Ding, and Jungong Han. Acnet: Strengthening the kernel skeletons for powerful cnn via asymmetric convolution blocks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1911–1920, 2019. 2

[10] Xiaohan Ding, Xiangyu Zhang, Jungong Han, and Guiguang Ding. Diverse branch block: Building a convolution as an inception-like unit. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10886–10895, 2021.

[11] Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. Repvgg: Making vgg-style convnets great again. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 2

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 6

[14] Fartash Faghri, Hadi Pouransari, Sachin Mehta, Mehrdad Farajtabar, Ali Farhadi, Mohammad Rastegari, and Oncel Tuzel. Reinforce data, multiply impact: Improved model accuracy and robustness with dataset reinforcement. *arXiv preprint arXiv:2303.08983*, 2023. 2, 3, 6

[15] Alex Fang, Gabriel Ilharco, Mitchell Wortsman, Yuhao Wan, Vaishaal Shankar, Achal Dave, and Ludwig Schmidt. Data determines distributional robustness in contrastive language image pre-training (clip). In *International Conference on Machine Learning*, pages 6216–6234. PMLR, 2022. 1

[16] Alex Fang, Albin Madappally Jose, Amit Jain, Ludwig Schmidt, Alexander Toshev, and Vaishaal Shankar. Data filtering networks. *arXiv preprint arXiv:2309.17425*, 2023. 2, 3, 8

[17] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19358–19369, 2023. 2

[18] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, et al. Datacomp: In search of the next generation of multimodal datasets. *arXiv preprint arXiv:2304.14108*, 2023. 2, 3, 5, 7, 8, 18

[19] Saurabh Garg, Mehrdad Farajtabar, Hadi Pouransari, Raviteja Vemulapalli, Sachin Mehta, Oncel Tuzel, Vaishaal Shankar, and Fartash Faghri. Tic-clip: Continual training of clip models. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024. 15

[20] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. Conformer: Convolution-augmented transformer for speech recognition. In *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, pages 5036–5040. ISCA, 2020. 3, 4

[21] Shuxuan Guo, Jose M Alvarez, and Mathieu Salzmann. Expandnets: Linear over-parameterization to train compact convolutional networks. *Advances in Neural Information Processing Systems*, 33:1298–1310, 2020. 2

[22] Ali Hatamizadeh, Greg Heinrich, Hongxu Yin, Andrew Tao, Jose M Alvarez, Jan Kautz, and Pavlo Molchanov. Fastervit: Fast vision transformers with hierarchical attention. *arXiv preprint arXiv:2306.06189*, 2023. 2, 3

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 2

[24] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8349, 2021. 5

[25] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15262–15271, 2021. 5

[26] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 2, 15

[27] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1314–1324, 2019. 2

[28] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 2

[29] 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. If you use this software, please cite it as below. 5, 8, 14, 15, 16, 17, 18

[30] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR, 2021. 1

[31] Huafeng Kuang, Jie Wu, Xiawu Zheng, Ming Li, Xuefeng Xiao, Rui Wang, Min Zheng, and Rongrong Ji. Dlip: Distilling language-image pre-training. *arXiv preprint arXiv:2308.12956*, 2023. 2

[32] Zhengfeng Lai, Haotian Zhang, Wentao Wu, Haoping Bai, Aleksei Timofeev, Xianzhi Du, Zhe Gan, Jiulong Shan, Chen-Nee Chuah, Yinfei Yang, et al. From scarcity to efficiency: Improving clip training via visual-enriched captions. *arXiv preprint arXiv:2310.07699*, 2023. 2, 6, 8, 18

[33] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017. 2, 3

[34] Xianhang Li, Zeyu Wang, and Cihang Xie. An inverse scaling law for clip training. *arXiv preprint arXiv:2305.07017*, 2023. 2, 7, 18

[35] 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. *arXiv preprint arXiv:2110.05208*, 2021. 2, 7

[36] Yanyu Li, Geng Yuan, Yang Wen, Ju Hu, Georgios Evangelidis, Sergey Tulyakov, Yanzhi Wang, and Jian Ren. Efficientformer: Vision transformers at mobilenet speed. *Advances in Neural Information Processing Systems*, 2022. 2

[37] Yanghao Li, Haoqi Fan, Ronghang Hu, Christoph Feichtenhofer, and Kaiming He. Scaling language-image pre-training via masking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23390–23400, 2023. 2

[38] Yanyu Li, Ju Hu, Yang Wen, Georgios Evangelidis, Kamyar Salahi, Yanzhi Wang, Sergey Tulyakov, and Jian Ren. Rethinking vision transformers for mobilenet size and speed. In *Proceedings of the IEEE international conference on computer vision*, 2023. 2, 3, 13

[39] Xinyu Liu, Houwen Peng, Ningxin Zheng, Yuqing Yang, Han Hu, and Yixuan Yuan. Efficientvit: Memory efficient vision transformer with cascaded group attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 4

[40] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021. 2

[41] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *arXiv preprint arXiv:2201.03545*, 2022. 2

[42] Sachin Mehta, Saeid Naderiparizi, Fartash Faghri, Maxwell Horton, Lailin Chen, Ali Farhadi, Oncel Tuzel, and Mohammad Rastegari. Rangeaugment: Efficient online augmentation with range learning. *arXiv preprint arXiv:2212.10553*, 2022. 14

[43] Norman Mu, Alexander Kirillov, David Wagner, and Saining Xie. Slip: Self-supervision meets language-image pretraining. In *European Conference on Computer Vision*, pages 529–544. Springer, 2022. 2, 7

[44] Mustafa Munir, William Avery, and Radu Marculescu. Mobilevig: Graph-based sparse attention for mobile vision applications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 2210–2218, 2023. 2, 3, 13

[45] Thao Nguyen, Samir Yitzhak Gadre, Gabriel Ilharco, Sewoong Oh, and Ludwig Schmidt. Improving multimodal datasets with image captioning. *arXiv preprint arXiv:2307.10350*, 2023. 2

[46] Hadi Pouransari, Mojan Javaheripi, Vinay Sharma, and Oncel Tuzel. Extracurricular learning: Knowledge transfer beyond empirical distribution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3032–3042, 2021. 2, 3, 6

[47] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1, 2, 3, 4, 5, 6, 7, 8, 14

[48] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10428–10436, 2020. 2, 13

[49] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, pages 5389–5400. PMLR, 2019. 5

[50] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018. 2

[51] 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. *arXiv preprint arXiv:2111.02114*, 2021. 2, 7, 8

[52] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022. 2, 8

[53] Abdelrahman Shaker, Muhammad Maaz, Hanoona Rasheed, Salman Khan, Ming-Hsuan Yang, and Fahad Shahbaz Khan. Swiftformer: Efficient additive attention for transformer-based real-time mobile vision applications. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 2, 3, 13

[54] Zhiqiang Shen and Eric Xing. A fast knowledge distillation framework for visual recognition. In *European Conference on Computer Vision*, pages 673–690. Springer, 2022. 2

[55] Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*, 2023. 2, 8

[56] Ximeng Sun, Pengchuan Zhang, Peizhao Zhang, Hardik Shah, Kate Saenko, and Xide Xia. Dime-fm: Distilling multimodal and efficient foundation models. *arXiv preprint arXiv:2303.18232*, 2023. 2, 6, 18

[57] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016. 7, 8

[58] Core ML Tools. Use Core ML Tools to convert models from third-party libraries to Core ML. https://coremltools.readme.io/docs, 2017. 5

[59] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357, 2021. 2, 13

[60] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021. 6

[61] Pavan Kumar Anasosalu Vasu, James Gabriel, Jeff Zhu, Oncel Tuzel, and Anurag Ranjan. Mobileone: An improved one millisecond mobile backbone. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7907–7917, 2023. 2, 4, 5

[62] Pavan Kumar Anasosalu Vasu, James Gabriel, Jeff Zhu, Oncel Tuzel, and Anurag Ranjan. Fastvit: A fast hybrid vision transformer using structural reparameterization. *arXiv preprint arXiv:2303.14189*, 2023. 2, 4, 13, 15

[63] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2, 8

[64] Zhecan Wang, Noel Codella, Yen-Chun Chen, Luowei Zhou, Xiyang Dai, Bin Xiao, Jianwei Yang, Haoxuan You, Kai-Wei Chang, Shih-fu Chang, et al. Multimodal adaptive distillation for leveraging unimodal encoders for vision-language tasks. *arXiv preprint arXiv:2204.10496*, 2022. 2

[65] Zhecan Wang, Noel Codella, Yen-Chun Chen, Luowei Zhou, Jianwei Yang, Xiyang Dai, Bin Xiao, Haoxuan You, Shih-Fu Chang, and Lu Yuan. Clip-td: Clip targeted distillation for vision-language tasks. *arXiv preprint arXiv:2201.05729*, 2022. 2

[66] Ross Wightman. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019. 13

[67] Felix Wu, Angela Fan, Alexei Baevski, Yann Dauphin, and Michael Auli. Pay less attention with lightweight and dynamic convolutions. In *International Conference on Learning Representations*, 2019. 3, 4

[68] Kan Wu, Houwen Peng, Zhenghong Zhou, Bin Xiao, Mengchen Liu, Lu Yuan, Hong Xuan, Michael Valenzuela, Xi Stephen Chen, Xinggang Wang, et al. Tinyclip: Clip distillation via affinity mimicking and weight inheritance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21970–21980, 2023. 2, 3, 4, 7, 8, 18

[69] Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Bin Xiao, Ce Liu, Lu Yuan, and Jianfeng Gao. Unified contrastive learning in image-text-label space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19163–19173, 2022. 2

[70] Kaicheng Yang, Jiankang Deng, Xiang An, Jiawei Li, Ziyong Feng, Jia Guo, Jing Yang, and Tongliang Liu. Alip: Adaptive language-image pre-training with synthetic caption. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2922–2931, 2023. 2, 6

[71] Shusheng Yang, Yixiao Ge, Kun Yi, Dian Li, Ying Shan, Xiaohu Qie, and Xinggang Wang. Rils: Masked visual reconstruction in language semantic space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23304–23314, 2023. 2, 7

[72] Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. Filip: Fine-grained interactive language-image pre-training. *arXiv preprint arXiv:2111.07783*, 2021. 2, 7

[73] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014. 5

[74] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive

captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022. 3

[75] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? In *The Eleventh International Conference on Learning Representations*, 2023. 5, 8

[76] Sangdoo Yun, Seong Joon Oh, Byeongho Heo, Dongyoon Han, Junsuk Choe, and Sanghyuk Chun. Re-labeling imagenet: from single to multi-labels, from global to localized labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2340–2350, 2021. 2

[77] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. *arXiv preprint arXiv:2303.15343*, 2023. 2, 8

[78] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, pages 13001–13008, 2020. 14