Note: A Summary is provided on page 18. It is advised to go through the summary before delving into the details of the paper

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.

1. Introduction

Large image-text foundation models, such as CLIP [46], have demonstrated excellent zero-shot performance and improved robustness [15] across a wide range of downstream tasks [29]. However, deploying these models on mobile devices is challenging due to their large size and high latency.

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 computationally expensive, hindering rapid development and exploration of efficient architecture design. On the other hand, standard

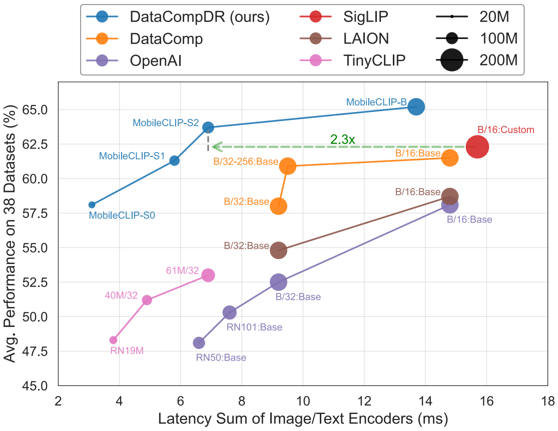


Figure 1. **MobileCLIP models are fast and accurate.** Comparison of publicly available CLIP models with MobileCLIP trained on our DataCompDR dataset. Latency is measured on iPhone12 Pro Max.

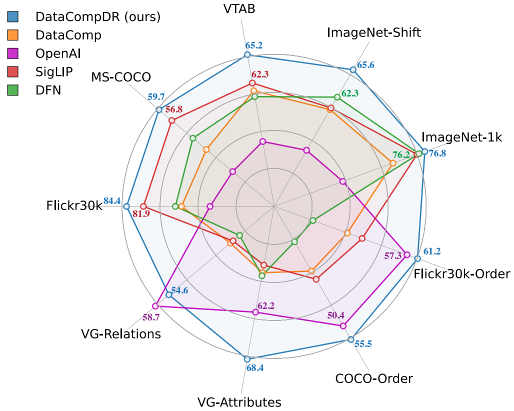


Figure 2. **DataCompDR dataset improves all metrics.** Zero-shot performance of CLIP models with ViT-B/16 image encoder.

Why do we want to design a lightweight version anyway?
What are the advantages of designing a lightweight model?

*Equal contribution.

multi-modal contrastive learning [46] 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, 20, 60] and convolutional token mixing [61]. 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 [46], 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.

Contributions

- 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, 36, 54, 70], unimodal self-supervision [34, 42], fine-grained image-text alignment [71], contrastive learning in image-text-label space [68], and pairwise Sigmoid loss [76]. Recently, CLIPA [33] proposed training at multi-resolutions for cost-effective CLIP training. These methods are complementary to our proposed method and can be exploited for further improvements.

CLIP training dataset is often comprising noisy image-text pairs obtained at web-scale. Since the original CLIP model [46], several works have demonstrated improved results on large-scale and filtered datasets [16, 18, 50, 51, 76]. 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 [31, 44, 69]. Our proposed reinforced multi-modal dataset also benefits from synthetically generated captions, which we show are crucial for improved learning efficiency.

Previous works explored extending unimodal knowledge distillation [25] to vision-language models. DIME-FM [55] proposes using in-domain unimodal data for distillation with a focus on zero-shot classification. TinyCLIP [67] 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 [30, 63, 64]. Our proposed multi-modal reinforced training also includes cross-modal affinity mimicking [67] toward targets that are added to our reinforced datasets. Further, we extend unimodal model ensembling [32, 45] to multimodal setup, and store targets obtained from an ensemble of CLIP models.

Offline knowledge distillation methods [14, 53, 75] 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, 22, 26, 27, 40, 47, 49, 60], transformer based [12, 39, 58] and convolution-transformer hybrids like [21, 35, 37, 43, 52, 61]. Similarly there are transformer based [62] and convolution-transformer hybrids like [19, 66] for text encoding. There have been works like [67], 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 [21, 37, 43, 52]. The optimizations introduced in [3, 67] 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. The proposed training strategy results in significant improvement in learning efficiency, i.e., reaching to certain target performance with less training budget and fewer samples.

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 [73] model and generate multiple synthetic captions $\boldsymbol{x}_{\text{syn}}^{(i,s)}$ for each image $\boldsymbol{x}_{\text{img}}^{(i)}$ (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. 2a) a combination of both real and synthetic captions is crucial to obtain best zero-shot retrieval and classification performance.

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

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

where $\boldsymbol{a}^{(i,j)}$ are the augmentation parameters that are sufficient to reproduce $\hat{\boldsymbol{x}}_{\text{img}}^{(i,j)}$ from $\boldsymbol{x}_{\text{img}}^{(i)}$ (see Fig. 3a). Ablations on the number and different kinds of augmentations used per image are provided in Tabs. 3a and 11, respectively.

Ensemble teacher. Model ensembling is a widely used technique for creating a stronger model from a set of independently trained ones [32, 45]. 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 $\hat{\boldsymbol{x}}_{\text{img}}^{(i,j)}$ and synthetic captions $\boldsymbol{x}_{\text{syn}}^{(i,s)}$ obtaining d_k-dimensional vectors $\psi_{\text{img}}^{(i,j,k)}$ and $\psi_{\text{syn}}^{(i,s,k)}$ for the k-th teacher model. We also compute the teacher embeddings $\psi_{\text{txt}}^{(i,k)}$ of the ground-truth captions $\boldsymbol{x}_{\text{txt}}^{(i)}$ (see Fig. 3b).

Reinforced dataset. We store the image augmentation parameters $\boldsymbol{a}^{(i,j)}$, synthetic captions $\boldsymbol{x}_{\text{syn}}^{(i,s)}$, feature embeddings $\psi_{\text{img}}^{(i,j,k)}$, $\psi_{\text{syn}}^{(i,s,k)}$ and $\psi_{\text{txt}}^{(i,k)}$ of the CLIP teachers as additional knowledge in the dataset along with the original image $\boldsymbol{x}_{\text{img}}^{(i)}$ and caption $\boldsymbol{x}_{\text{txt}}^{(i)}$ (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 $\Psi_{\text{img}}^{(k)}, \Psi_{\text{txt}}^{(k)} \in \mathcal{R}^{b \times d_k}$ 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 [46] loss $\mathcal{L}_{\text{CLIP}}(\mathcal{B})$ and a knowledge distillation loss $\mathcal{L}_{\text{Distill}}(\mathcal{B})$:

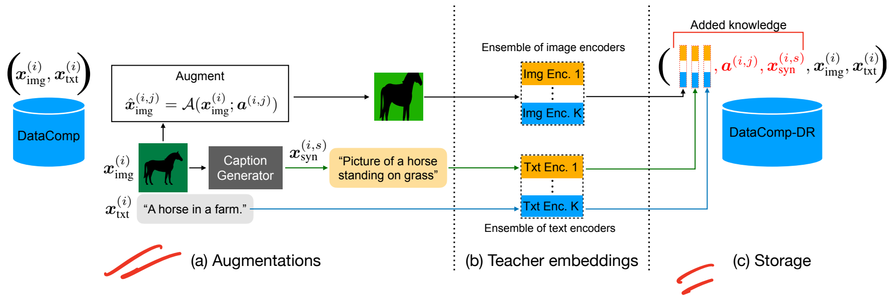
$$\boxed{\begin{aligned}
\mathcal{L}_{\text{Total}}(\mathcal{B}) &= \lambda \mathcal{L}_{\text{CLIP}}(\mathcal{B}) + (1-\lambda)\mathcal{L}_{\text{Distill}}(\mathcal{B}), \qquad (2)\\
\mathcal{L}_{\text{Distill}}(\mathcal{B}) &= \frac{1}{2}\mathcal{L}_{\text{Distill}}^{\text{I2T}}(\mathcal{B}) + \frac{1}{2}\mathcal{L}_{\text{Distill}}^{\text{T2I}}(\mathcal{B}),\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

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

Can you think of a better loss function than KL divergence?

Can you think of a reason why they used BN even though it has become out of fashion?

If you can answer this without looking up, you have learned the hard lessons! Good job 👏



$(\boldsymbol{x}_{\text{img}}^{(i)}, \boldsymbol{x}_{\text{txt}}^{(i)})$
DataComp
Augment
$\hat{\boldsymbol{x}}_{\text{img}}^{(i,j)} = \mathcal{A}(\boldsymbol{x}_{\text{img}}^{(i)}; \boldsymbol{a}^{(i,j)})$
$\boldsymbol{x}_{\text{img}}^{(i)}$
Caption Generator
$\boldsymbol{x}_{\text{syn}}^{(i,s)}$
"Picture of a horse standing on grass"
$\boldsymbol{x}_{\text{txt}}^{(i)}$
"A horse in a farm."
Ensemble of image encoders
Img Enc. 1
Img Enc. K
Txt Enc. 1
Txt Enc. K
Ensemble of text encoders
Added knowledge
$\boldsymbol{a}^{(i,j)}, \boldsymbol{x}_{\text{syn}}^{(i,s)}, \boldsymbol{x}_{\text{img}}^{(i)}, \boldsymbol{x}_{\text{txt}}^{(i)}$
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.

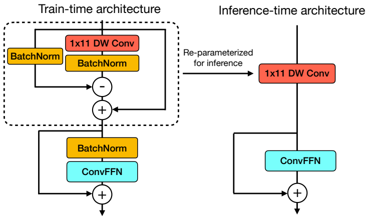
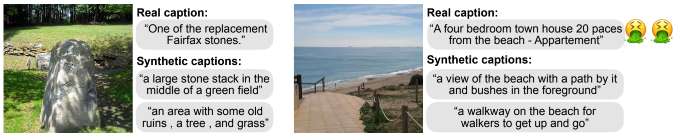


Figure 4. Architecture of convolutional and reparameterizable blocks, called Text-RepMixer used in MobileCLIP's text encoder - MCt.



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.

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 Tab. 3d). For every sample, we read the image $\boldsymbol{x}_{\text{img}}^{(i)}$ and the corresponding ground-truth caption $\boldsymbol{x}_{\text{txt}}^{(i)}$ from the dataset. Then, we randomly load one of stored augmentation parameters $\boldsymbol{a}^{(i,j)}$ and reproduce the augmented image $\hat{\boldsymbol{x}}_{\text{img}}^{(i,j)}$. We also randomly load one of synthetic captions $\boldsymbol{x}_{\text{syn}}^{(i,s)}$. 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 [46] 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 [66] 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. We design variants of MobileCLIP using both transformer and hybrid text encoders.

Neither conv nor attention alone are enough for efficient architectures

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 [61]. 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 [19], 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. For all depthwise 1-D convolutions, we use a kernel size of 11. This was chosen based on accuracy-latency tradeoff, full ablation can be found in Appendix E. 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. 4b).

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 [61] architecture with certain key differences explained below.

In FastViT, an MLP expansion ratio of 4.0 is used for FFN blocks. Recent works like [38, 67] 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 Tab. 4a. MCi0 has similar stage configuration as [60]. 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 [60]. We find that this design has a minimal impact on latency, but a good improvement in capacity of the model as reflected in the downstream task performance. When trained from scratch on ImageNet dataset for image classification task, MCi2 attains the same top-1 accuracy of 84.5% as FastViT [61] (previous state-of-the-art hybrid vision transformer), while being 15% smaller and 14.3% faster. More details can be found in Appendix A.

5. Experiments

In this section, we present our experimental setup, ablations on our proposed method and fast MobileCLIP architectures, 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 [48], ImageNet-A [24], ImageNet-O [24], ImageNet-R [23], 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 [72] 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 [74]. 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 Appendix A).

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 1. **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 [28], 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.

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 [46]. 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) [57] 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 [60].

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 ($\sim$20 epochs). Table 1 summarizes the analysis of our training.

Strong image augmentations. In contrast to uni-modal supervised and self-supervised methods for vision where strong

$\mathcal{B} \in$	$\{\mathcal{B}_{\text{real}}\}$	$\{\mathcal{B}_{\text{syn}}\}$	$\{\mathcal{B}_{\text{real}}$ 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 2. **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.

augmentations are used [13, 59], CLIP training recipes [46] often use light image augmentations to avoid image-text misalignment. However, several works [2, 14, 45] demonstrated the efficacy of strong augmentations in a distillation setup. In Tab. 1 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 Appendix B.

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. 1 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. 2a, we observe a similar trend for CLIP training with distillation loss only ($\lambda = 1$). In Tab. 2b, 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 [31, 69] while distillation works focus on zero-shot classification [55]. 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. 1). We also observe that the most accurate models are not the best teachers. See Appendix C 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. 3a and 3b. 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. 14.

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

Variant	$\{C_1, C_2, C_3, C_4\}$	$\{L_1, L_2, L_3, L_4\}$
MCi0	$\{64, 128, 256, 512\}$	$\{2, 6, 10, 2\}$
MCi1	$\{64, 128, 256, 512\}$	$\{4, 12, 20, 4\}$
MCi2	$\{80, 160, 320, 640\}$	$\{4, 12, 24, 4\}$

(a) Configurations of MCi.

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

(b) Effect of the number of self-attention layers in MCt.

Table 4. **Ablation on architecture.** 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. 3d 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. In general, the storage size of datasets depends on the file format and the tradeoff between load time and the compression rate. 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. 3c. We provide analysis on additional size reductions in Appendix D 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. 3a and 3b.

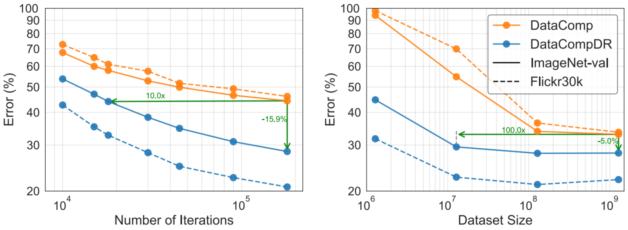
Hybrid text encoder. We ablate over the number of Text-RepMixer blocks that can effectively replace self-attention layers with negligible impact on zero-shot performance. For this ablation, we choose a 6-layer purely convolutional text encoder and systematically introduce self-attention layers in the middle. From Tab. 4b, 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. 5, 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 [42](42.8→45.0%). In comparison to CLIPA [33] 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).

Name	Dataset	Seen Samples	Latency (ms) (img+txt)	Zero-shot IN-val
CLIP-B/16 [42, 46]	CC-12M [4]	0.39B	11.5 + 3.3	36.5
CLIP-B/16 [42, 46]	YFCC-15M [56]	0.37B		37.6
SLIP-B/16 [42]	CC-12M [4]	0.39B		40.7
SLIP-B/16 [42]	YFCC-15M [56]	0.37B		42.8
CLIP-B/16	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, 46]	YFCC-15M [56]	0.49B	5.9 + 3.3	32.8
SLIP-B/32 [7, 42]				34.3
FILIP-B/32 [7, 71]				39.5
DeCLIP-B/32 [34]				43.2
DeFILIP-B/32 [7]				45.0
RILS-B/16 [70]	LAION-20M [50]	0.5B	11.5 + 3.3	45.0
TinyCLIP-8M/16 [67]	YFCC-15M [56]	0.75B	**2.0 + 0.6**	41.1
SLIP-B/16 [42]	YFCC-15M [56]	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 [67]	YFCC-15M [56]	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 [42]	YFCC-15M [56]	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 [33]	LAION-400M [50]	2.69B†	11.5 + 3.3	63.2

Table 5. **Small-scale CLIP training.** † 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.

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.

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. 6, we compare with methods with large scale training. MobileCLIP-S0, trained on DataCompDR-1B significantly outperforms recent works like TinyCLIP [67], and has similar performance as a ViT-B/32 model trained on

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 [46]	-	ViT-L/14 ViT-L/14	Base Base	(-)	(-)	80.1	69.6	74.5	92.3	46.7	66.5	67.3
TinyCLIP-RN19M [67]	LAION-400M [50]	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 [67]	LAION-400M [50]	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 [67]	LAION-400M [50]	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 [46]	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 [67]	LAION-400M [50]	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 [67]	LAION-400M [50] YFCC-15M [56]	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 [46]	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 [46]	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 [51]	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 [31]	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 [46]	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 [51]	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 [54]	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 [76]	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**

Table 6. **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 [28]. Models are grouped by their total latency in increasing order and by performance within each group. “Base” refers to standard CLIP Transformer-based [62] 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 [54] 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[51] dataset. All other models, including ours are trained from scratch.

DataComp [18] while being 2.8× smaller and 3× faster. 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 [76] 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 [74]. We compare our MobileCLIP-B trained on DataCompDR-1B with all the publicly available ViT-B/16:Base models in Tab. 7. 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 [76], 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 [46]	68.3	**58.7**	62.2	50.4	57.3
CLIP	LAION-2B [51]	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 [76]	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 7. **Performance on ARO benchmark.** All the models use ViT-B/16 as image encoder and the standard CLIP text encoder. For VG Rel. and VG Attr. datasets, Macro Accuracy is reported and for Flickr30k-Order and COCO-Order recall@1 is reported. For more details please refer to [74].

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 pre-training to recognize long-tail visual concepts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3558–3568, 2021. 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. 2

[7] Yufeng Cui, Lichen Zhao, Feng Liang, Yangguang Li, and Jing Shao. Democratizing contrastive language-image pre-training: 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, 3

[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

[19] 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

[20] 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

[21] 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. 3

[22] 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

[23] 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

[24] 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

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

[26] 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

[27] 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

[28] 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, 2, 3, 4

[29] 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

[30] 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

[31] 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, 5

[32] 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

[33] Xianhang Li, Zeyu Wang, and Cihang Xie. An inverse scaling law for clip training. *arXiv preprint arXiv:2305.07017*, 2023. 2, 7, 5

[34] 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

[35] 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. 3

[36] 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

[37] 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. 3, 1

[38] 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

[39] 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. 3

[40] 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

[41] 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. 2

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

[43] 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. 3, 1

[44] 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

[45] 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

[46] 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

[47] 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, 1

[48] 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

[49] 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

[50] 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

[51] 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

[52] 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. 3, 1

[53] 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

[54] 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

[55] 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, 5

[56] 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

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

[58] 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. 3, 1

[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. PMLR, 2021. 6

[60] 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, 5

[61] 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, 3, 4, 5, 1

[62] 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. 3, 8

[63] 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

[64] 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

[65] Ross Wightman. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019. 1

[66] 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

[67] 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, 5

[68] 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

[69] 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

[70] 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

[71] 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

[72] 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

[73] 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

[74] 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

[75] 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

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

[77] 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. 2

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

Supplementary Material

A. Experimental Setup

Additional details of our training and evaluation are provided here. We train all models at 224 resolution.

Table 10 summarizes the hyperparameters we used to train MobileCLIP-B on DataCompDR-1B. For other variants of MobileCLIP (S0, S1, and S2) we use the same hyperparameters except using $\lambda = 1.0$. For experiments on DataCompDR-12M we use global batch size of 8192.

For our ensemble distillation ablations in Appendix C, we use 32 total A100 GPUs but we use the same global batch size of 8192 as our other ablations. We also use a smaller uniformly sampled DataComp-8M for ablations in Appendices B and C that results in a slightly lower performance than DataCompDR-12M used for the rest of ablations.

For ImageNet-1k experiments in Sec. 4.2, we follow the training recipe prescribed in [37, 58], i.e. the models are trained for 300 epochs using AdamW optimizer with weight decay of 0.05 and peak learning rate 10^{-3} for a total batch size of 1024. The number of warmup epochs is set to 5 and cosine schedule is used to decay the learning rate. The teacher model for distillation is RegNetY-16GF [47] Our implementation uses Timm library [65] and all the models were trained on single machine with 8×NVIDIA A100 GPUs. The hyperparameters for the three variants of MCi are detailed in Tab. 8. The performance of MCi variants is detailed in Tab. 9 and compared against recent state-of-art efficient architectures. MCi obtains the best trade-off amongst recent efficient architectures as seen in Fig. 7.

Hyperparameter	Training
	MCi0, MCi1, MCi2
Stochastic depth rate	[0.0, 0.05, 0.15]
Input resolution	256×256
Data augmentation	RandAugment
Mixup α	0.8
CutMix α	1.0
Random erase prob.	0.25
Label smoothing	0.1
Train epochs	300
Warmup epochs	5
Batch size	1024
Optimizer	AdamW
Peak learning rate	1e-3
LR. decay schedule	cosine
Weight decay rate	0.05
Gradient clipping	✗
EMA decay rate	0.9995

Table 8. Training hyperparameters for ImageNet-1k experiments.

Model	Eval Image Size	Param (M)	FLOPs (G)	Mobile Latency (ms)	Top-1 Acc. (%)
MobileViG-M [43]	224	14.0	1.5	1.4	80.6
SwiftFormer-L1 [52]	224	12.1	1.6	1.5	80.9
EfficientFormerV2-S2 [37]	224	12.6	1.3	1.6	81.6
FastViT-SA12 [61]	256	11.5	1.9	1.5	81.9
MCi0 (ours)	256	11.8	2.4	1.5	82.2
MobileViG-B [43]	224	26.7	2.8	2.3	82.6
SwiftFormer-L3 [52]	224	28.5	4.0	2.6	83.0
EfficientFormerV2-L [37]	224	26.1	2.6	2.6	83.3
FastViT-SA24 [61]	256	21.5	3.8	2.4	83.4
MCi1 (ours)	256	21.9	4.7	2.5	83.8
FastViT-MA36 [61]	256	43.9	7.8	4.3	84.5
MCi2 (ours)	256	36.3	7.8	3.6	84.5

Table 9. Comparison of MCi variants with recent state-of-the-art models on ImageNet classification task.

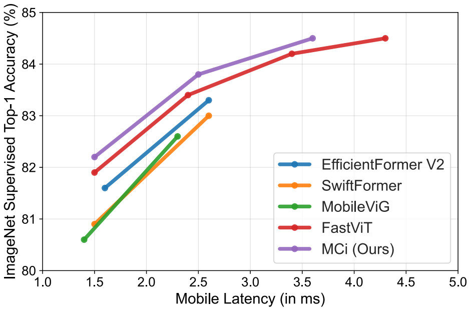


Figure 7. Top-1 Accuracy on ImageNet v/s latency plot of MCi variants and recent state-of-the-art architectures.

B. Image Augmentation

In this section we provide a detailed ablation on the effect of image augmentations. The training setup is the same as training with DataCompDR-12M presented in Sec. 5.2, except we used an 8M subset for this ablation. In Tab. 11 we show classification and retrieval performance of a ViT-B/16 based CLIP model trained with our final loss as in Eq. (3) ($\lambda = 1$) and different image augmentations. Note that we feed the same augmented image to both teacher and student models. First, we consider `RandomResizedCrop` (RRC) with three magnitudes (0.08, 0.4, 0.9) determining the lower bound of random area of the crop (smaller lower bound means stronger augmentation). We observe that strong RRC results in significant accuracy improvement both for classification and retrieval metrics. While using strong RRC augmentation is standard for supervised training, for CLIP training the widely used recipe [46] includes weak RRC (lower-bound for scale= 0.9).

Hyperparameter	Value MobileCLIP-B, S0, S1, S2
Input resolution	224^2, 256^2, 256^2, 256^2
Context length	77
Data augmentation	RandAugment
Random resize crop scale	[0.08, 1.0]
Random resized crop ratio	[0.75, 1.33]
RangeAugment target value	(40, 20)
Train iterations	200k
Warmup iterations	2k
Global batch size	65536
Optimizer	AdamW
AdamW beta1	0.9
AdamW beta2	0.95
Max learning rate	1e-3
Min learning rate	1e-6
LR. decay schedule	cosine
Weight decay rate	0.2
Gradient clipping	✗
Mixed precision	BFloat16
EMA decay rate	0.9995
CLIP loss weight	0.25
KD loss weight	0.75
GT caption weight	1.0
Synth. caption weight	1.0
Synth. teacher	coca_ViT-L-14
Teacher 1	openai-ViT-L-14
Teacher 2	datacomp_xl_s13b_b90k-ViT-L-14
Teacher resolution	224×224

Table 10. Training hyperparameters for our CLIP experiments on DataCompDR.

We further utilize `RangeAugment` [41] to automatically adjust Brightness, Contrast, and Noise. We use PSNR metric with target range [20, 40] and a Cosine curriculum. Since in `RangeAugment` individual augmentation magnitudes are adjusted dynamically during training, they cannot be stored as part of the dataset reinforcement process. Hence, we only apply it to images fed to the student model. We show that if the same augmentation is applied to both student and teacher (not feasible for our dataset reinforcement approach) further improvement can be obtained (56.6% vs 55.9% on ImageNet-val).

Finally, we consider `RandomHorizontalFlip`, `RandomErasing` [77], and `RandAugment` [6], and find that only `RandAugment` is beneficial in our setup. Our reinforced datasets include parameters of RRC and `RandAugment` and during training time we apply `RangeAugment` to images fed to the student model.

C. CLIP Ensembles

In this section we provide a detailed ablation on CLIP ensembles. First, we show that we can construct more accurate zero-shot models by ensembling pretrained individual CLIP models. For inference, we concatenate normalized embeddings of each modality followed by a re-normalization. In Tab. 12 we show performance of some CLIP ensemble models that we picked from OpenCLIP [28]. We also include performance of individual models. Evidently, ensembling results in improved performance. For example, an ensemble of two pretrained `ViT-L-14`-based CLIP models from `datacomp_xl_s13b_b90k` and `openai` results in average performance of 67.3%, while each individual model has 66.3% and 61.7% performance, respectively. Further, ensembling can be a more parameter efficient approach to obtain a stronger model. For instance, the ensemble of two `ViT-L-14`-based CLIP models has less parameters than the one with `ViT-bigG-14` image encoder, but comes with the same ImageNet-val performance (80.1%). In general, given a set of pretrained CLIP models (e.g., as in OpenCLIP [28]) with this approach we can push state-of-the-art and obtain stronger zero-shot performance. Here, we show and ensemble of four CLIP models can reach up to 81.7% zero-shot classification performance on ImageNet-val, while individual models' performance is not more than 80.1%. As stronger individual models become publicly available, one can create stronger ensembles with this approach.

Image Augmentations	Zero-shot CLS		Flickr30k Ret.		COCO Ret.		Avg Perf.
	IN-val	IN-shift	I2T	T2I	I2T	T2I	on 38
RandomResizedCrop: `0.9-1.0` Student-RangeAugment [41]	51.0	40.1	54.2	68.5	30.5	45.3	45.9
RandomResizedCrop: `0.4-1.0` Student-RangeAugment	55.0	43.9	60.4	76.0	34.1	48.4	48.9
RandomResizedCrop: `0.08-1.0` Student-RangeAugment	55.9	44.6	58.8	76.1	34.2	49.0	49.6
RandomResizedCrop: `0.08-1.0`	56.4	44.6	59.8	74.6	34.4	49.3	49.1
RandomResizedCrop: `0.08-1.0` Student&Teacher-RangeAugment	56.6	44.9	60.2	74.0	34.9	50.5	50.8
RandomResizedCrop: `0.08-1.0` Student-RangeAugment RandomHorizontalFlip: `p=0.5`	55.9	44.7	59.4	75.9	34.4	49.2	48.8
RandomResizedCrop: `0.08-1.0` Student-RangeAugment RandomErasing [77]: `p=0.25`	55.8	44.5	59.4	75.3	34.5	49.7	49.1
RandomResizedCrop: `0.08-1.0` Student-RangeAugment RandAugment [6]	56.6	45.4	60.9	78.3	35.0	51.0	50.2

Table 11. Ablation on different augmentations for distillation. We highlight our choice with blue.

In this work, we are interested in creating a strong ensemble model to be used as a teacher in the context of distillation. In Tab. 13 we show performance of a ViT-B/16 CLIP model trained with different CLIP models as teacher (both individual models and ensembles). Training setup is the same as that of in Sec. 5.2, except we use a uniformly sampled 8M subset. Similar to standard distillation for classification task [25], we observe that more accurate CLIP models are not necessarily better teachers. We picked the ensemble of two `ViT-L-14`-based CLIP models as the teacher model (highlighted in blue) in our dataset reinforcement process.

Teacher	Teacher	Teacher	Zero-shot CLS		Flickr30k Ret.		COCO Ret.		Avg Perf.
Models(s)	Pre-taining(s)	Resolution(s)	IN-val	IN-shift	I2T	T2I	I2T	T2I	on 38
ViT-bigG-14	laion2b_s39b_b160k	224	80.1	69.1	79.6	92.9	51.4	67.4	66.7
EVA01-g-14-plus	merged2b_s11b_b114k	224	79.3	69.3	79.0	91.7	50.3	68.2	66.2
ViT-L-14	datacomp_xl_s13b_b90k	224	79.2	67.9	73.4	89.0	45.7	63.3	66.3
ViT-L-14	openai	224	75.5	64.9	65.0	85.2	36.5	56.3	61.7
ViT-L-14-336	openai	336	76.6	67.1	66.9	87.7	37.1	57.9	62.8
ViT-L-14	datacomp_xl_s13b_b90k	224	80.1	69.6	74.5	92.3	46.7	66.5	67.3
ViT-L-14	openai	224							
ViT-L-14	datacomp_xl_s13b_b90k	224	80.5	70.6	75.8	91.8	47.0	67.0	67.8
ViT-L-14-336	openai	336							
EVA01-g-14-plus	merged2b_s11b_b114k	224	81.1	70.9	78.1	93.8	50.2	69.7	68.5
ViT-L-14	datacomp_xl_s13b_b90k	224							
ViT-L-14	openai	224							
EVA01-g-14-plus	merged2b_s11b_b114k	224	81.2	71.6	78.8	93.7	50.2	69.9	68.9
ViT-L-14	datacomp_xl_s13b_b90k	224							
ViT-L-14-336	openai	336							
convnext_xxlarge	laion2b_s34b_b82k_augreg_soup	256	81.5	71.7	79.0	94.5	50.5	69.5	68.7
ViT-L-14	datacomp_xl_s13b_b90k	224							
ViT-L-14-336	openai	336							
ViT-bigG-14	laion2b_s39b_b160k	224	81.6	71.7	79.9	94.6	52.4	71.3	69.4
EVA01-g-14-plus	merged2b_s11b_b114k	224							
ViT-L-14	datacomp_xl_s13b_b90k	224							
ViT-L-14	openai	224							
EVA01-g-14-plus	merged2b_s11b_b114k	224	81.7	72.1	80.0	95.0	52.0	70.8	69.3
ViT-L-14-336	openai	336							
ViT-L-14	datacomp_xl_s13b_b90k	224							
convnext_xxlarge	laion2b_s34b_b82k_augreg_soup	256							
ViT-L-14	openai	224	78.2	68.9	73.4	89.7	42.0	63.5	65.5
ViT-L-14-336	openai	336							
RN50x64	openai	384							
RN50x16	openai	448							

Table 12. Zero-shot evaluation of (ensemble of) clip models. Each group of rows corresponds to an ensemble teacher. All models are taken from OpenCLIP [28] on Aug-2023. We highlight our choice with blue .

D. Ablations on Lossy Compressions

In Tab. 3c we presented the storage sizes for DataCompDR-12M and DataCompDR-1B with BFloat16 compression of the embeddings. In this section, we further analyze the storage reduction by i) reducing the number of augmentations, and ii) lossy compression of embeddings.

We report the total storage size for 12.8k samples of DataCompDR in Tab. 14. The storage size for DataCompDR-12M can be easily deduced by multiplying the numbers by 1000 (TBs instead of GBs) and by 10^5 for DataCompDR-1B.

Table 15 shows the accuracy of training with BFloat16 embeddings achieves accuracies within the standard deviation of the training on DataComp-12M.

E. Hybrid Text Encoder

In this section, we ablate over kernel dimensions for our hybrid text encoder. For this ablation, we use a 6-layered fully convolutional text encoder and systematically increase the kernel size. We use ViT-B/16 as the image encoder for these runs. These models were trained on DataCompDR-12M for 30k iterations. From Tab. 16, we notice that zero-shot IN-val performance does improve with increased kernel size, but it is significantly more expensive to run the model on mobile device. For zero-shot IN-val performance improvement of 1.1%, the model is 4.5× slower. From Tab. 16, kernel size of 11 obtains the best accuracy-latency trade-off.

F. Extended Results

In this section we provide extended zero-shot results of our proposed family of CLIP models: MobileCLIP-S0, MobileCLIP-S1, MobileCLIP-S2, and MobileCLIP-B. Zero-shot classification and retrieval results are provided in Tab. 17. We also include additional results from related works where only partial evaluation is available.

Teacher Models(s)	Teacher Pre-taining(s)	Teacher Resolution(s)	Zero-shot CLS		Flickr30k Ret.		COCO Ret.		Avg Perf.
			IN-val	IN-shift	I2T	T2I	I2T	T2I	on 38
ViT-bigG-14	laion2b_s39b_b160k	224	53.4	42.6	59.6	76.2	35.8	52.1	47.8
EVA01-g-14-plus	merged2b_s11b_b114k	224	54.5	43.3	59.6	74.6	35.4	50.8	47.7
ViT-L-14	datacomp_xl_s13b_b90k	224	54.0	43.4	58.9	74.3	34.3	50.1	48.3
ViT-L-14	openai	224	54.4	42.7	54.5	69.1	29.7	44.6	47.2
ViT-L-14-336	openai	336	54.2	43.3	53.6	68.7	30.1	44.3	47.2
ViT-L-14	datacomp_xl_s13b_b90k	224	56.3	44.8	59.2	74.5	34.4	49.9	49.6
ViT-L-14	openai	224							
ViT-L-14	datacomp_xl_s13b_b90k	224	55.9	44.6	58.8	76.1	34.2	49.0	49.6
ViT-L-14-336	openai	336							
EVA01-g-14-plus	merged2b_s11b_b114k	224	56.2	45.0	59.6	76.9	35.7	51.5	49.4
ViT-L-14	datacomp_xl_s13b_b90k	224							
ViT-L-14	openai	224							
EVA01-g-14-plus	merged2b_s11b_b114k	224	56.0	44.5	60.1	76.5	35.3	50.6	49.5
ViT-L-14	datacomp_xl_s13b_b90k	224							
ViT-L-14-336	openai	336							
convnext_xxlarge	laion2b_s34b_b82k_augreg_soup	256	55.8	44.4	59.4	75.1	35.0	49.5	50.1
ViT-L-14	datacomp_xl_s13b_b90k	224							
ViT-L-14-336	openai	336							
ViT-bigG-14	laion2b_s39b_b160k	224	56.3	44.6	60.8	76.2	35.8	51.4	49.2
EVA01-g-14-plus	merged2b_s11b_b114k	224							
ViT-L-14	datacomp_xl_s13b_b90k	224							
ViT-L-14	openai	224							
EVA01-g-14-plus	merged2b_s11b_b114k	224	55.9	44.6	60.4	75.1	35.6	52.3	49.4
ViT-L-14-336	openai	336							
ViT-L-14	datacomp_xl_s13b_b90k	224							
convnext_xxlarge	laion2b_s34b_b82k_augreg_soup	256							
ViT-L-14	openai	224	56.4	44.6	57.9	72.0	31.7	47.0	48.6
ViT-L-14-336	openai	336							
RN50x64	openai	384							
RN50x16	openai	448							

Table 13. Ablation on using different (ensemble of) teacher models in our multi-modal distillation. Each group of rows demonstrate an ensemble teacher. Student architecture is fixed to ViT-B/16 for image encoder and base 12-layer Transformer for text encoder (MobileCLIP-B setup). For this ablation, we use an 8M subset of DataComp and train all experiments for 20k iterations with global batch size of 8k. All models are imported from OpenCLIP [28] on Aug-2023. We highlight our choice with blue .

Image	Text	Syn.	Aug. Params	Text Emb.	Image Emb.	BFloat16	Sparsity	Size (GBs)
✓	✓	✗	✗	✗	✗	✗	✗	0.9
✓	✓	✓	✓	✗	✗	✗	✗	0.9
✓	✓	✓	✓	5+1	30	✗	✗	3.3
✓	✓	✓	✓	5+1	30	✓	✗	1.9
✓	✓	✓	✓	5+1	30	✗	50%	1.8
✓	✓	✓	✓	5+1	30	✓	50%	1.3
✓	✓	✓	✓	5+1	10	✗	✗	1.9
✓	✓	✓	✓	5+1	10	✓	✗	1.4
✓	✓	✓	✓	5	5	✗	✗	1.5
✓	✓	✓	✓	5	5	✓	✗	1.2
✓	✓	✓	✓	2	2	✗	✗	1.1
✓	✓	✓	✓	2	2	✓	✗	1.0

Table 14. Total storage for 12.8k samples stored in individual Pickle Gzip files. Storage for 12.8M and 1.28B samples are approximately the same numbers in TBs and 100 TBs.

Num. Aug.	1	2	5	10	15	20	25	30
w/o BFloat16	60.63	63.27	64.81	64.74	64.49	64.92	64.78	64.74
w/ BFloat16	-	-	64.32	64.88	64.57	64.81	65.13	64.91

Table 15. Effect of BFloat16 and the number of augmentations on ImageNet-val zero-shot Accuracy. We train on DataCompDR-12M for approximately 30 epochs.

Kernel Size	3	11	31
Num Params. (M)	38.2	38.3	38.4
Latency (ms)	1.0	1.2	5.4
IN-val	56.3	57.9	59.0

Table 16. Ablation on kernel size for text encoder. We train for 30k iterations. We highlight our choice with blue

Name	ImageNet Shifts CLS							Flickr30k Retrieval						COCO Retrieval					
	val	A	R	O	S	V2	Obj	T→I			I→T			T→I			I→T		
								@1	@5	@10	@1	@5	@10	@1	@5	@10	@1	@5	@10
MobileCLIP-B	76.8	58.7	89.6	41.4	64.5	69.8	69.4	91.4	99.1	99.9	77.3	94.4	96.7	68.8	88.3	92.9	50.6	74.9	82.9
MobileCLIP-S2	74.4	49.3	87.0	46.9	62.2	66.8	66.6	90.3	98.9	99.6	73.4	92.3	95.6	63.4	85.1	91.4	45.4	70.1	79.0
MobileCLIP-S1	72.6	40.3	84.7	50.5	60.3	64.9	63.4	89.2	98.0	99.5	71.0	91.3	95.3	62.2	84.3	90.1	44.0	68.9	77.7
MobileCLIP-S0	67.8	26.5	78.6	53.8	55.5	59.9	55.9	85.9	97.1	98.8	67.7	88.8	93.3	58.7	81.1	88.2	40.4	66.0	75.9
DIME-FM-B/32 [55]	66.5	32.2	69.8	(-)	46.5	58.9	43.2	(-)	(-)	(-)	(-)	(-)	(-)	(-)	(-)	(-)	(-)	(-)	(-)
VeCLIP-B/16 [31]	64.6	(-)	(-)	(-)	(-)	57.7	(-)	91.1	98.5	99.7	76.3	93.5	96.4	67.2	87.3	92.7	48.4	73.3	81.8
TinyCLIP-63M/32 [67]	64.5	22.8	74.1	(-)	50.8	55.7	31.2	84.9	(-)	(-)	66.0	(-)	(-)	56.9	(-)	(-)	38.5	(-)	(-)
CLIPA-B/16 [33]	63.2	26.8	73.2	(-)	48.7	55.6	44.3	75.3	(-)	(-)	58.3	(-)	(-)	53.1	(-)	(-)	35.2	(-)	(-)

Table 17. Extended zero-shot evaluations. We also include additional results from related works where the full DataComp [18] evaluation was not accessible. Numbers are read from the corresponding papers. For each method we picked their best model up to ViT-B/16 size. Please see Tab. 6 for additional details including runtime benchmarking. Models are sorted by their zero-shot classification performance on ImageNet-val. Here our MobileCLIP-S1 is fully trained with 13B seen samples.

Summary

Abstract and Introduction

- Over the last few years, we have seen incredible progress in contrastive pretraining of image-text models. These models have demonstrated excellent zero-shot performance and improved robustness on a range of downstream tasks.
- One bottleneck remains. Most image-text pre-trained models utilize transformer-based encoders with significant memory and latency overhead, making them unsuitable for deploying on mobile and edge devices. To this end, the authors propose 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.
- MobileCLIP sets a new state-of-the-art latency-accuracy tradeoff for zero-shot classificatio n and retrieval tasks on several datasets.
- Main contributions made in this paper:
    - A new family of mobile-friendly CLIP models known as MobileCLIP
    - Multi-modal reinforced training
    - Two variants of reinforced datasets: DataCompDR-12M, and DataCompDR-1B

Multi-Modal Reinforced Training

- Leverages knowledge transfer from an image captioning model and a strong encoder of pre-trained CLIP models for training the target model.
- Two main components:
    - Leveraging the knowledge of an image captioning model to generate synthetic captions
    - Knowledge distillation of image-text alignments from an ensemble of strong pre-trained CLIP models
- The additional knowledge (synthetic captions and teacher embeddings) in the dataset, thereby avoiding any training time computational overhead such as evaluation of captioning or the ensemble teacher model.

- Synthetic Caption
    - Used the CoCa model to generate multiple synthetic captions for each image in the filtered DataComp dataset
    - A good property of synthetic captions is that they are generally less noisy compared to the web-crawled datasets used for language-image pretraining.

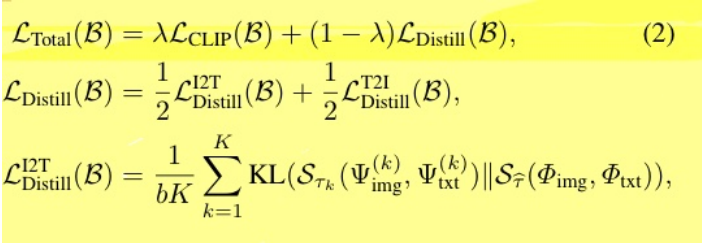
- Image Augmentations
    - For each image, the authors generate multiple augmented versions using a parameterized augmentation function A
    - Augmentations include RandomResizedCrop, RangeAugment, RandomHorizontalFlip, RandomErasing, and RandAugment. Of the last three, RandAugment performs better than the others
    - Performance peaks after 5 image augmentations and 2 synthetic captions

- Ensemble Teacher
    - Ensemble of K CLIP models

    Each model is used to compute the feature embeddings for the images (real + augmented), and the captions (real + synthetic)

- Reinforced dataset
    - Store image augmentation parameters, synthetic captions, and feature embeddings computed by the models in the ensemble in addition to the original images and the corresponding captions.
    - Reinforcement is a one-time cost but is necessary for efficient training and fast experiments.

- Training
    - Loss function
        - Distill the affinity matrix between image-text pairs from multiple image-text teacher encoders into a student image-text encoder.
        - Two components:
            - Standard CLIP loss
            - Knowledge distillation loss
        - Distillation loss is the KL divergence between the similarity scores produced by the K models in the teacher ensemble and the scores produced by the student model.



$$\mathcal{L}_{\text{Total}}(\mathcal{B}) = \lambda \mathcal{L}_{\text{CLIP}}(\mathcal{B}) + (1-\lambda)\mathcal{L}_{\text{Distill}}(\mathcal{B}), \qquad (2)$$

$$\mathcal{L}_{\text{Distill}}(\mathcal{B}) = \frac{1}{2}\mathcal{L}^{\text{I2T}}_{\text{Distill}}(\mathcal{B}) + \frac{1}{2}\mathcal{L}^{\text{T2I}}_{\text{Distill}}(\mathcal{B}),$$

$$\mathcal{L}^{\text{I2T}}_{\text{Distill}}(\mathcal{B}) = \frac{1}{bK}\sum_{k=1}^{K} \text{KL}(\mathcal{S}_{\tau_k}(\Psi^{(k)}_{\text{img}}, \Psi^{(k)}_{\text{txt}}) \| \mathcal{S}_{\hat{\tau}}(\Phi_{\text{img}}, \Phi_{\text{txt}})),$$

- Architecture
    - Text Encoder
        - Hybrid text encoder - consists of self-attention, and 1D convolutional layers, and is named as Text-RepMixer.
        - MobileCLIP variants use both transformer-based text encoder and hybrid text encoder.
        - Text-RepMixer uses BatchNorm and 1D Convolutions. For feed-forward network (FFN) blocks, linear layers are augmented with an additional 1-D convolution of similar kernel dimensions as the token mixer. These are named ConvFFN blocks.
        - For the depthwise conv, a kernel size of 11 is used.

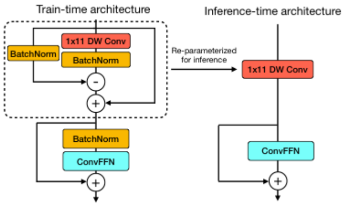


Figure 4. Architecture of convolutional and reparameterizable blocks, called Text-RepMixer used in MobileCLIP's text encoder - MCt.

- Image Encoder
    - Inspired by FastViT
    - Lowered the expansion ratio used in FastViTs from 4 to 3, and increased the depth
    - Different variants of MobileCLIP have different depths and widths. When trained from scratch on the ImageNet dataset for the image classification task, MCi2 attains the same top-1 accuracy of 84.5% as FastViT [61] (previous state-of-the-art hybrid vision transformer) while being 15% smaller