MobileCLIP

> 原文地址：2311.17049

MobileCLIP：基于多模态强化训练的快速图文模型

帕万・库马尔・阿纳索萨卢・瓦苏 、*哈迪・普兰萨里* 、法尔塔什・法赫里 *、拉维泰贾・维穆拉帕利、翁塞尔・图泽尔（苹果公司）电子邮箱：
{panasosaluvasu,mpouransari,fartash,r_vemulapalli,otuzel}@apple.com

摘要

CLIP 等图文基础模型的对比预训练，在各类下游任务中展现出优异的零样本性能和更强的鲁棒性。然而，这些模型采用基于 Transformer 的大型编码器，存在显著的内存和延迟开销，难以在移动设备上部署。本文提出**MobileCLIP**—— 一套专为运行时性能优化的高效图文模型，同时提出一种新颖高效的训练方法，即**多模态强化训练**。该训练方法利用图像描述生成模型和 CLIP 强集成编码器的知识迁移，提升高效模型的准确率；并通过将额外知识存储在强化数据集中，避免了训练时的计算开销。在多个数据集的零样本分类和检索任务中，MobileCLIP 实现了当前最优的延迟 - 准确率权衡：其中 MobileCLIP-S2 变体相较于此前基于 ViT-B/16 的最优 CLIP 模型，速度提升 2.3 倍，准确率同时更高。此外，我们基于 ViT-B/16 图像骨干网络训练 CLIP 模型，验证了多模态强化训练的有效性 —— 在 38 个评估基准上，其平均性能较此前最优结果提升 2.9%。研究还表明，与非强化 CLIP 训练相比，该方法的学习效率提升了 10 倍至 1000 倍。相关代码和模型已开源，地址为：https://github.com/apple/ml-mobileclip

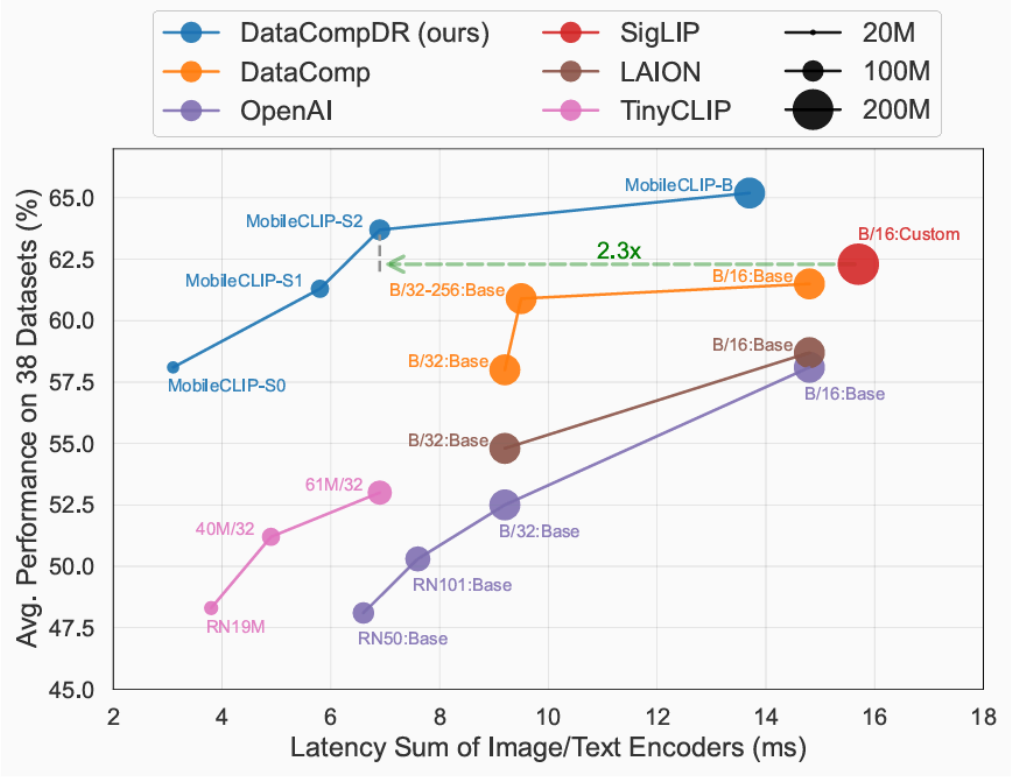


Figure 1. **MobileCLIP models are fast and accurate.** Comparison of publicly available CLIP models with MobileCLIP trained on our DataCompDR dataset. Latency is measured on iPhone12 Pro Max.

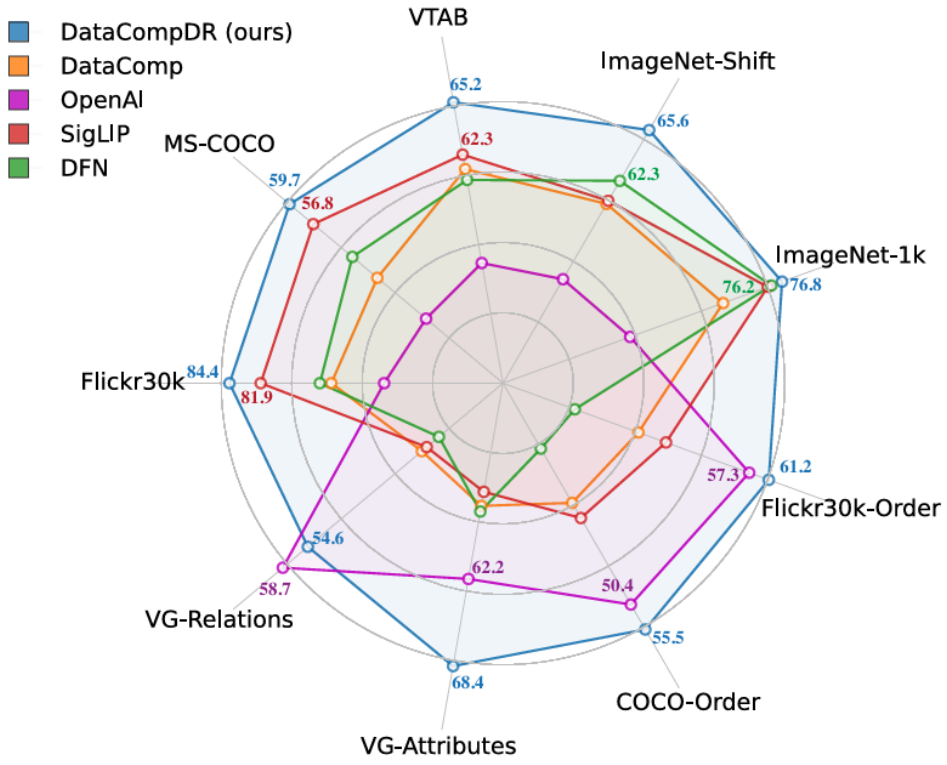


Figure 2. **DataCompDR dataset improves all metrics.** Zero-shot performance of CLIP models with ViT-B/16 image encoder.

1. 引言

CLIP 等大型图文基础模型 [47] 在各类下游任务中 [30] 展现出优异的零样本性能和更强的鲁棒性 [15]，但由于模型规模大、延迟高，在移动设备上部署面临挑战。

我们的目标是设计一套适用于移动设备的**对齐图文编码器**，实现这一目标需解决两大核心挑战：

1. 运行时性能（如延迟）与架构准确率存在权衡关系，需快速且全面地分析不同架构设计；但 CLIP 模型的大规模训练计算成本高昂，阻碍了高效架构设计的快速开发与探索；而小规模下的标准多模态对比学习 [47] 准确率较差，无法为架构设计提供有效指导信号。
2. 小型架构的容量有限，导致准确率欠佳，需通过更优的训练方法提升性能。

为应对这些挑战，我们基于数据集强化方法 [14] 提出一种新颖的训练框架：① 用额外信息对数据集进行一次强化；② 将强化数据集用于多次实验。在固定计算预算下，基于强化数据集的训练准确率高于原始数据集。我们提出多模态版本的数据集强化方法，用于训练高效 CLIP 模型：具体而言，通过为图文数据集 DataComp [18] 添加合成描述和 CLIP 强预训练集成模型的嵌入向量，得到强化数据集**DataCompDR**。我们还设计了两种 DataCompDR 变体：适用于高效模型设计快速迭代的 DataCompDR-12M，以及用于大规模训练最优性能的 DataCompDR-1B。

实验表明，基于 DataCompDR 的训练较标准 CLIP 训练，学习效率显著提升。例如，使用单节点 8×A100 GPU，在 DataCompDR-12M 上从零训练基于 ViT-B/16 [12] 的 CLIP 模型，约 1 天即可在 ImageNet-val [8] 上实现 61.7% 的零样本分类准确率；而基于 DataCompDR-1B 的训练在多个指标上刷新当前最优（图 2），且训练计算开销仅为此前工作的一部分。

借助 DataCompDR，我们探索架构设计空间，提出一套适用于移动设备的对齐图文编码器家族 **MobileCLIP**，其延迟 - 准确率权衡性能优于此前工作（图 1）。我们通过多种架构设计技术构建高效图像与文本编码器，包括结构重参数化 [9-11,21,61] 和卷积令牌混合 [62]。MobileCLIP 包含 S0、

S1、S2、B 四种变体，覆盖不同规模和延迟，适配各类移动应用场景：其中最快的 MobileCLIP-S0 变体，较标准 OpenAI ViT-B/16 CLIP 模型 [47] 速度提升约 5 倍、规模缩小 3 倍，且平均准确率持平。

本文主要贡献如下：

- 设计适用于移动设备的 CLIP 模型家族 MobileCLIP：其变体采用混合 CNN-Transformer 架构，在图像和文本编码器中引入结构重参数化，实现规模与延迟的优化。
- 提出**多模态强化训练**策略：通过图像描述生成预训练模型和 CLIP 强集成模型的知识迁移，提升学习效率；并发布两种强化数据集变体 DataCompDR-12M 与 DataCompDR-1B，实验验证其学习效率较 DataComp 提升 10 倍至 1000 倍。
- MobileCLIP 家族在零样本任务中实现当前最优的延迟 - 准确率权衡，其中基于 ViT-B/16 的 MobileCLIP 模型同样刷新该架构的性能上限。

2. 相关工作

CLIP 的高效学习

可通过优化训练目标提升 CLIP 的学习效率，例如图像掩码 [17,37,55,71]、单模态自监督 [35,43]、细粒度图文对齐 [72]、图文标签空间对比学习 [69]、成对 Sigmoid 损失 [77] 等；CLIPA [34] 通过多分辨率训练降低训练成本，这些方法与本文方法具有互补性。

CLIP 训练数据集多为网页级噪声图文对，自原始 CLIP 模型 [47] 以来，已有研究通过大规模过滤数据集 [16,18,51,52,77] 提升性能。除数据收集与过滤外，近期研究表明，将预训练描述生成模型生成的视觉增强型合成描述与真实描述结合，可提升 CLIP 模型质量 [32,45,70]。本文提出的强化多模态数据集同样受益于合成描述，实验验证其对提升学习效率至关重要。

此前工作如 DIME-FM [56] 将单模态蒸馏 [26] 扩展至零样本分类场景；TinyCLIP [68] 通过跨模态相似度模仿和权重继承训练紧凑型 CLIP 模型；多模态蒸馏也被用于特定任务的融合型视觉 - 语言学生模型训练 [31,64,65]。本文的多模态强化训练同样包含跨模态相似度模仿 [68]，并将单模态模型集成 [33,46] 扩展至多模态场景，同时存储 CLIP 集成模型的目标输出。

近期提出的离线知识蒸馏方法 [14,54,76] 可降低运行大型教师模型的训练时开销，本文将数据集强化策略 [14] 扩展至 CLIP 的多模态场景，所提出的强化多模态数据集在不增加训练计算开销的前提下，显著提升模型准确率。

CLIP 的高效架构

近年来，针对资源受限设备视觉任务的高效架构层出不穷，可大致分为纯卷积架构 [11,23,27,28,41,48,50,61]、基于 Transformer 的架构 [12,40,59]，以及卷积 - Transformer 混合架构 [22,36,38,44,53,62]；文本编码领域则有基于 Transformer 的架构 [63] 和卷积 - Transformer 混合架构 [20,67]。此外，TinyCLIP [68] 通过剪枝 ViT 架构得到更小更快的 CLIP 模型，PuMer [3] 通过减少图文令牌数量提升视觉 - 语言模型推理速度，但这些模型规模仍较大，难以在移动设备上高效部署。本文提出改进的卷积 - Transformer 混合架构，同时适配视觉和文本模态，性能优于近期最优架构 [22,38,44,53]；且 [3,68] 中的优化方法可进一步提升 MobileCLIP 的效率。

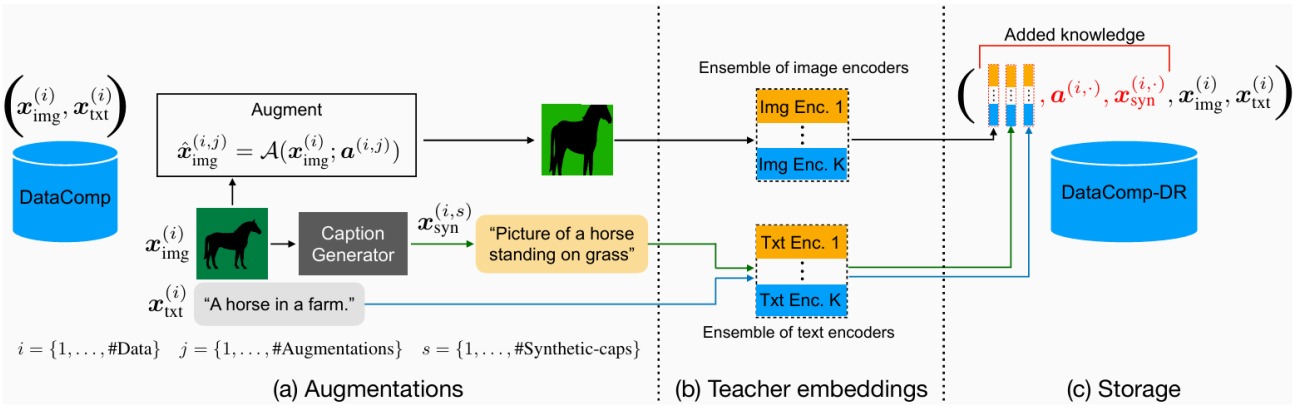


Figure 3. Illustration of multi-modal dataset reinforcement with one image augmentation and one synthetic caption. In practice, we use multiple image augmentations and synthetic captions.

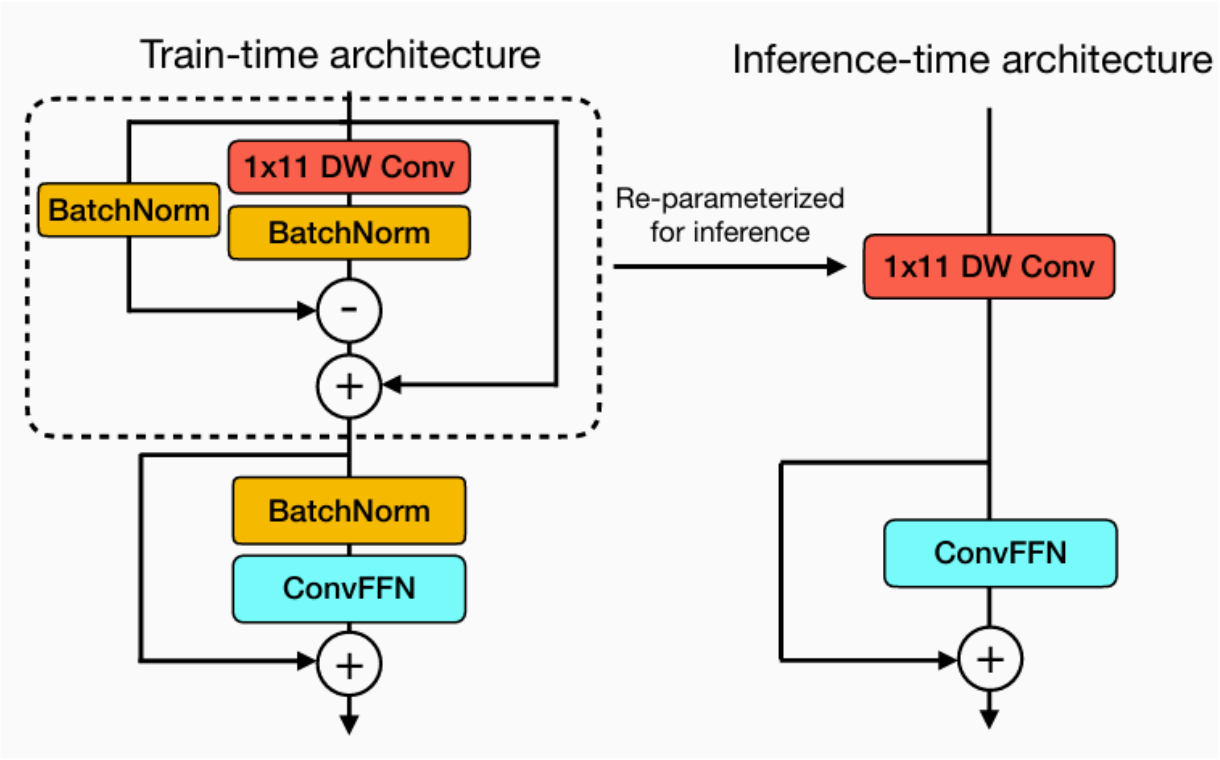


Figure 4. Architecture of convolutional and reparameterizable blocks, called Text-RepMixer used in MobileCLIP's text encoder MCt.

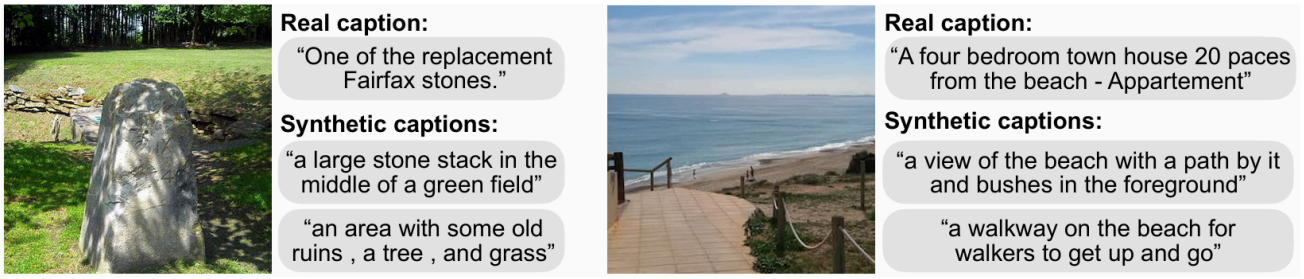


Figure 5. Real vs synthetic captions.

3. 多模态强化训练

多模态强化训练利用图像描述生成模型和 CLIP 强预训练集成模型的知识迁移，训练目标模型。该框架包含两大核心组件：① 借助合成描述利用图像描述生成模型的知识；② 从 CLIP 强预训练集成模型中蒸馏图文对齐知识。我们遵循 [14] 的数据集强化策略，将额外知识（合成描述、教师模型嵌入）存储在数据集中（见图 3），避免训练时运行描述生成模型或集成教师模型带来的额外计算开销。

3.1 数据集强化

合成描述

CLIP 训练所用的图文数据集多源自网页，存在固有噪声。DataComp [18]、数据过滤网络 [16] 等工作通过严格过滤提升网页数据集质量，但即使经过过滤，描述的视觉信息量仍可能不足。为增强描述的视觉表达能力，我们使用主流 CoCa [74] 模型，为每张图像$x_{img}^{(i)}$生成多个合成描述$x_{syn}^{(i,s)}$（见图 3a）。第 5.1 节将分析每张图像生成的合成描述数量对性能的影响，图 5 展示了 CoCa 模型生成的合成描述示例。实验表明，真实描述虽更具体但噪声较多，而真实描述与合成描述的结合，是实现最优零样本检索与分类性能的关键（表 3a）。

图像增强

对每张图像$x_{img}^{(i)}$，采用参数化增强函数 $\mathcal{A}$ 生成多个增强图像 $\hat{x}_{img}^{(i,j)}$，公式如下：

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

其中 $a^{(i,j)}$ 为增强参数，可基于原始图像 $x_{img}^{(i)}$ 复现增强图像 $\hat{x}_{img}^{(i,j)}$（见图 3a）。第 5.1 节将通过表 4a 和表 13，分析每张图像的增强次数与增强类型对性能的影响。

集成教师模型

模型集成是通过多个独立训练模型构建更强模型的常用技术 [33,46]。本文将其扩展至多模态场景，采用K个 CLIP 模型组成强教师模型（第 5.1 节将分析教师模型选择对性能的影响）。对增强图像 $\hat{x}_{img}^{(i,j)}$ 和合成描述 $x_{syn}^{(i,s)}$，计算每个教师模型的特征嵌入，得到第k个教师模型的 d_k 维向量 $\psi_{img}^{(i,j,k)}$ 和 $\psi_{syn}^{(i,s,k)}$；同时计算真实描述 $x_{txt}^{(i)}$ 的教师嵌入 $\psi_{txt}^{(i,k)}$（见图 3b）。（三个特征嵌入：图像的、合成描述的、真实描述的）

强化数据集

我们将图像增强参数 $a^{(i,j)}$ 、合成描述 $x_{syn}^{(i,s)}$ ，以及 CLIP 教师模型的特征嵌入 $\psi_{img}^{(i,j,k)}$ 、$\psi_{syn}^{(i,s,k)}$ 、$\psi_{txt}^{(i,k)}$ 作为额外知识，与原始图像 $x_{img}^{(i)}$ 、原始描述 $x_{txt}^{(i)}$ 一同存储在数据集中（见图 3c）（七种数据元素）。注意，数据集强化是一次性成本，可通过多次高效模型训练与实验分摊。 *（一次性成本 指的是这笔投入只需要支付一次。 一旦你完成了一个高质量、大规模、清洗好的数据集，这个“资产”就形成了。前期在数据上的一次性高昂投入，可以被后续无数次的模型训练和实验所“分摊”。你训练模型的次数越多，单次训练所分摊到的数据成本就越低。）*

3.2 训练

损失函数

直观而言，损失函数的核心是将多个图文教师编码器的图文对相似度矩阵，蒸馏到学生图文编码器中。设 $\mathcal{B}$ 为包含b个（图像，文本）对的批次，$\Psi_{img}^{(k)}$ 、$\Psi_{txt}^{(k)} \in \mathbb{R}^{b\times d_k}$ 分别为批次 $\mathcal{B}$ 中第k个教师模型的 d_k 维图像、文本嵌入矩阵；Φ_{img} 、$\Phi_{txt} \in \mathbb{R}^{b\times d}$ 分别为目标模型的图像、文本嵌入矩阵。对矩阵U和V，定义 $S_\tau(U,V) \in \mathbb{R}^{b\times b}$ 为相似度矩阵，由 $UV^\top/\tau$ 经行级 Softmax 运算得到（ τ 为温度参数）。

训练损失包含两部分：标准 CLIP 损失 $\mathcal{L}_{CLIP}(\mathcal{B})$ [47] 和知识蒸馏损失 $\mathcal{L}_{Distill}(\mathcal{B})$ ，总损失公式如下：

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

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

$$\mathcal{L}_{Distill}^{I2T}(\mathcal{B}) = \frac{1}{bK}\sum_{k=1}^{K} KL\left(S_{\tau_k}(\Psi_{img}^{(k)}, \Psi_{txt}^{(k)}) \parallel S_{\hat{\tau}}(\Phi_{img}, \Phi_{txt})\right)$$

其中，KL表示 KL 散度（Kullback-Leibler 散度）； $\mathcal{L}_{Distill}^{T2I}$ 通过交换 $\mathcal{L}_{Distill}^{I2T}$ 中的文本与图像嵌入项计算；λ 为权衡参数。

高效训练

基于强化数据集的训练仅需修改数据加载器和损失函数，以利用数据集中存储的额外知识，训练成本与标准 CLIP 训练持平（见表 4d）。具体流程如下：

1. 从数据集中读取图像 $x_{img}^{(i)}$ 和对应的真实描述 $x_{txt}^{(i)}$ ；
2. 随机加载一个存储的增强参数 $a^{(i,j)}$ ，复现增强图像 $\hat{x}_{img}^{(i,j)}$ ； 同时随机加载一个合成描述 $x_{syn}^{(i,s)}$ ；
3. 读取K个教师模型对应的存储嵌入 $\psi_{img}^{(i,j,k)}$ 、$\psi_{syn}^{(i,s,k)}$ 、$\psi_{txt}^{(i,k)}$ ；
4. 构建两个数据批次：B_{real}（增强图像 - 真实描述对）和 B_{syn}（增强图像 - 合成描述对），并分别对两个批次计算式 (2) 的训练损失；
5. 最终损失为两个批次的总损失之和，公式如下：

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

需注意，由于计算蒸馏损失所需的教师嵌入已作为数据集的一部分存储，学生模型仅需一次前向传播即可计算总损失，无需额外的教师相关计算。 *（无需额外的教师相关计算： 因为教师模型的输出已经预先计算好并存储在数据集里了，所以在训练循环中，当需要计算蒸馏损失时，程序只需要直*

接从硬盘或内存中读取预先存好的“教师嵌入”即可，完全不需要在运行时再次调用那个庞大、耗时的教师模型。）

4. 架构设计

4.1 文本编码器

CLIP [47] 采用视觉 Transformer 与含自注意力层的经典 Transformer 结合，实现文本编码。该模型虽有效，但移动部署更需小型高效模型。近期研究 [67] 表明，卷积在文本编码中可实现与 Transformer 相当的性能，但本文发现纯卷积架构的性能显著低于 Transformer。因此，我们提出**混合文本编码器**，结合 1D 卷积与自注意力层。

混合文本编码器的核心是**Text-RepMixer**—— 一种卷积令牌混合模块，可解耦训练时与推理时架构，灵感源自结构重参数化卷积令牌混合（RepMixer）[62]。推理时，跳跃连接被重参数化（架构见图 4）。在前馈网络（FFN）块中，我们为线性层添加深度可分离 1D 卷积（卷积核尺寸与令牌混合模块相近），得到 ConvFFN 块。该结构与 [20] 中的卷积块类似，核心差异在于引入批归一化（batchnorm），且可与后续深度可分离 1D 卷积层折叠，实现高效推理。附录 F 将详细讨论 Text-RepMixer 的设计选择。

为寻找混合文本编码器的最优设计，我们从纯卷积文本编码器出发，逐步用自注意力层替换卷积块（见表 5）。表 1 对比了本文文本编码器与 CLIP 基础文本编码器的性能：与 ViT-S/16 等高效骨干网络结合时，本文模型规模更小、速度更快，且准确率与更大的基础文本编码器持平。

4.2 图像编码器

近期研究表明，混合视觉 Transformer 在视觉表征学习中表现优异。本文基于 FastViT [62] 架构，提出改进的混合视觉 Transformer**MCi**，核心差异如下：

FastViT 的 FFN 块采用 4.0 倍的 MLP 扩展比，而近期研究 [39,68] 发现 FFN 块线性层存在大量冗余。为提升参数效率，我们将扩展比降至 3.0，同时增加架构深度，确保图像编码器参数数量不变。三种 MCi 变体的阶段配置见附录 A：MCi0 与 [61] 的阶段配置相近；MCi1 为 MCi0 的加深版本；MCi2 为 MCi1 的加宽版本。各变体的阶段计算占比与 [61] 相近，实验表明该设计对延迟影响极小，但能显著提升模型容量，进而改善下游任务性能（附录 B）。

表 1 对比了 MCi 编码器与同规模 FastViT-MA36 编码器（均作为 CLIP 图像编码器）的性能：在 DataCompDR-12M 上训练 30k 迭代（约 0.24B 样本）后，本文 MCi 模型的零样本 IN-val 准确率显著更高，且速度提升 16.3%。

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.

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.

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

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.

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.

5. 实验

5.0 实验设置与评估指标

评估指标

采用 DataComp [18] 的评估基准，具体包括：

- 零样本分类：ImageNet 验证集 [8] 及其分布偏移数据集（ImageNetV2 [49]、ImageNet-A [25]、ImageNet-O [25]、ImageNetR [24]、ObjectNet [1]），后者平均性能记为 IN-Shift；
- 零样本图文检索：MSCOCO [5] 和 Flickr30k [73] 数据集的 Recall@1；
- 38 个 DataComp 评估数据集的平均性能；
- Attribute, Relation and Order（ARO）基准 [75]：Visual Genome Relation、Visual Genome Attributes、Flickr30k-Order、COCO-Order 数据集。

文中 “IN-val” 指零样本 ImageNet 验证集准确率；“Flickr30k” 指零样本图文检索与文图检索的平均 Recall@1；所有指标均为无微调结果。

训练设置

实验分为消融实验与大规模实验两类：

- 消融实验：基于 1280 万图文对数据集训练，全局批次大小为 8192，使用 8×NVIDIA-A100-80GB GPU，训练 30-45k 迭代；
- 大规模实验：全局批次大小为 65536，使用 256×A100 GPU，训练 200k 迭代；所有模型均从零训练（细节见附录 B）。

数据集

训练基于 DataComp 数据集 [18] 的图像 - 文本对，采用 12.8 亿样本的 Bestpool 过滤子集（性能最优），记为 DataComp-1B；为快速实验，构建 1280 万样本的固定均匀采样子集，记为 DataComp-12M（DataComp [18] 未研究该子集，但实验表明其性能优于同规模 DataComp-medium 的 Bestpool 子集）。

DataCompDR（强化版 DataComp）：采用多模态数据集强化策略对 DataComp 进行强化，得到 DataCompDR-1B（强化 DataComp-1B）和 DataCompDR-12M（强化 DataComp-12M）。强化为一次性过程，成本可通过多次架构实验分摊：

- 用 OpenCLIP [29] 的 coca_ViT-L-14 模型，为每张图像生成 5 个合成描述；
- 采用强随机图像增强（DataCompDR-1B 为 10 次 / 图像，DataCompDR-12M 为 30 次 / 图像）；
- 计算两个强教师模型（OpenCLIP 中预训练权重为 datacomp_xl_s13b_b90k 和 openai 的 ViT-L-14）的嵌入：包括增强图像、真实描述、合成描述的嵌入，嵌入向量为 2×768 维向量的 1536 维拼接；
- 采用无损压缩和 BFloat16 格式存储所有强化信息。

第 5.1 节将分析上述设计选择的合理性。DataCompDR 中的一个 “样本” 指 “随机增强图像 + 真实描述 + 随机合成描述” 的三元组。

MobileCLIP 架构

MobileCLIP 架构由 MCi（图像编码器）与 MCt（文本编码器）配对构成，具体包括 3 个小型变体和 1 个标准变体：

- MobileCLIP-S0（MCi0:MCt）、MobileCLIP-S1（MCi1:Base）、MobileCLIP-S2（MCi2:Base），其中 “Base” 为 12 层 Transformer，与 ViT-B/16 CLIP [47] 的文本编码器结构一致；
- MobileCLIP-B：采用 ViT-B/16:Base 配对，为本文训练的标准 CLIP 模型。

延迟基准测试

延迟测试采用各方法对应的输入尺寸：在 iPhone 设备上，通过 Core ML Tools（v7.0）[58] 导出模型，在搭载 iOS 17.0.3 的 iPhone 12 Pro Max 上运行；所有模型的批次大小均设为 1，测试流程遵循 [61]。

5.1 消融实验

消融实验基于 ViT-B/16:Base 编码器，在 DataComp-12M 上训练 30k 迭代，全局批次大小为 8k（约 20 轮），分析训练与架构各组件的影响（表 2 为消融实验总结）。

强图像增强

与单模态监督 / 自监督方法常用的强增强 [13,60] 不同，CLIP 训练通常采用弱图像增强 [47]，以避免图文错位。但近期研究 [2,14,46] 表明，蒸馏场景下强增强可提升性能。表 2 显示，强图像增强使蒸馏性能显著提升：IN-val 准确率 + 4.8%，Flickr30k 检索率 + 4.4%。附录 C 将详细分析图像增强的影响。

合成描述

与图像增强类似，合成描述（或描述增强）可进一步提升 CLIP 模型性能，尤其在图文检索任务中。表 2 显示，标准 CLIP 训练（$\lambda = 0$）中，结合真实描述与合成描述的批次，可使 IN-val 准确率 + 7.4%，Flickr30k 检索率 + 27.5%；表 3a 显示，仅使用蒸馏损失的 CLIP 训练（$\lambda = 1$）也呈现类似趋势。表 3b 分析了 λ 的影响：$\lambda = 1.0$ 时 IN-val 性能最优，$\lambda = 0.7$ 时 Flickr30k 性能最优。此前利用合成描述的工作多聚焦检索性能提升 [32,70]，而蒸馏工作多聚焦零样本分类 [56]；在大规模实验中，MobileCLIP-B 采用 $\lambda = 0.75$ 平衡两者，小型变体采用 $\lambda = 1.0$ 。

集成教师模型

表 2 显示，将 CLIP 强集成模型作为多模态强化训练的教师，是实现 IN-val 准确率 + 2.4% 的关键。实验还发现，准确率最高的模型未必是最优教师，附录 D 将全面分析不同教师模型的性能。

图像增强次数与合成描述数量

我们为每张图像生成多个增强版本和合成描述，并与教师嵌入一同高效存储，表 4a 和表 4b 分析了其数量对性能的影响。实验基于 DataCompDR-12M 训练 45k 迭代（约 30 轮），增强次数最多 30 次，合成描述最多 5 个。结果表明：5 次增强、2 个合成描述后，性能基本饱和，说明模型可多次复用单个增强 / 合成描述，直至充分学习其中包含的知识。若需降低成本，可减少增强次数与合成描述数量；为追求最优性能，DataCompDR-12M 和 DataCompDR-1B 分别采用 10 次 / 30 次增强、5 个合成描述。

训练时间与存储开销

强化训练的核心优势是训练时间与非强化训练差异极小。表 4d 对比了标准 CLIP 训练、在线蒸馏训练、描述生成训练的耗时：在单节点 8×A100-80GB GPU 上，DataCompDR-12M 单轮训练仅需 1.3 小时；无数据集强化时，训练速度慢 16 倍；仅部分强化（添加合成描述）时，训练速度慢 3 倍。

表 4c 对比了强化数据集与原始 DataComp 数据集的存储需求：每个图文对单独存储为一个文件，采用 Pickle 格式 + Gzip 压缩，图像 - 文本嵌入采用 BFloat16 格式。1280 万样本的 DataCompDR-12M 需 1.9TB 存储，12.8 亿样本的 DataCompDR-1B 需 140TB 存储。附录 E 将分析进一步压缩存储的方法，并验证 BFloat16 格式对准确率无影响。基于表 4a 和表 4b 的消融结果，推荐 DataCompDR-12M 采用 5 次增强 / 5 个合成描述（30 轮训练），DataCompDR-1B 采用 2 次增强 / 2 个合成描述（10 轮训练），以最小化存储开销。

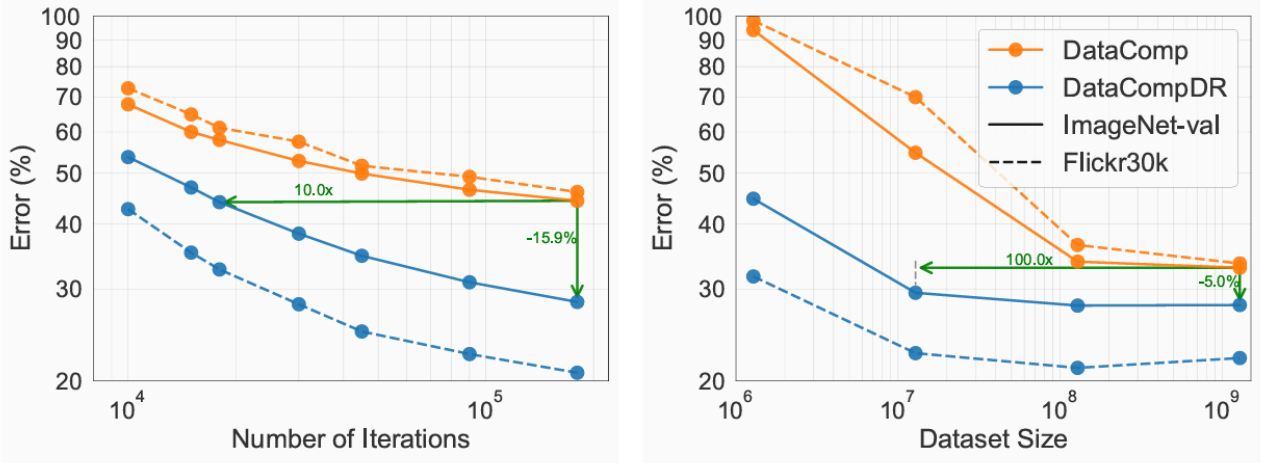
混合文本编码器

消融实验分析了 Text-RepMixer 块替换自注意力层的效果（零样本性能损失最小）。实验采用 6 层纯卷积文本编码器，在中间层逐步引入自注意力层。表 5 显示：即使引入 1 个自注意力层，零样本性能也显著提升；最优权衡为 2 个 Text-RepMixer 块 + 4 个自注意力层 —— 该变体（MCt）与纯 Transformer 变体准确率持平，但规模缩小 5%，速度提升 15.8%。

5.2 小规模训练场景

表 6 对比了基于 1200 万 - 2000 万样本数据集的训练方法（适用于架构搜索等快速探索场景）。结果表明：

- 基于 DataCompDR-12M 训练的 MobileCLIP-B，仅使用 3.7 亿样本，性能显著优于所有训练样本量 4 倍以上的方法；
- MobileCLIP-B 的样本量缩放性能优异（准确率从 65.3% 提升至 71.7%），远超 SLIP [43]（从 42.8% 提升至 45.0%）；
- 与 CLIPA [34]（通过多分辨率训练提升效率）相比，DataCompDR-12M 训练更高效：CLIPA 需 26.9 亿多分辨率样本（等效于 0.5 亿 224×224 样本），IN-val 准确率仅 63.2%，而 MobileCLIP-B 仅用 0.37 亿样本，准确率达 65.3%；
- 与 TinyCLIP 相比：TinyCLIP-39M/16 延迟更高、准确率更低；TinyCLIP-8M/16 与 MobileCLIP-S0 延迟相近（2.6ms vs 3.1ms），但准确率显著更低（41.1% vs 59.1%）。



(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 学习效率

已知知识蒸馏的长期训练可持续提升分类模型性能 [2]，图 6a 显示，强化训练同样受益于长期训练：在 DataComp-1B 的 1200 万子集上训练 120 轮后，ImageNet-val 零样本准确率达 71.7%，而非强化训练最高仅 55.7%。

图 6b 分析了数据集规模对性能的影响：采用 DataComp-1B 的子集（128 万 - 12.8 亿样本），所有实验均训练 20k 迭代（全局批次大小 65k，等效于 12.8 亿样本子集训练 1 轮）。结果表明：DataCompDR 训练仅需 128 万样本，IN-val 准确率即可达 55.2%，而 DataComp-1B 训练仅约 6%；该设置下，DataCompDR 的样本效率提升超 100 倍，Flickr30k 检索任务的样本效率甚至提升 1000 倍。

5.4 与当前最优方法的对比

表 7 对比了大规模训练方法的性能，结果表明：

- MobileCLIP-S0 基于 DataCompDR-1B 训练，性能显著优于 TinyCLIP [68] 等近期工作，与 DataComp [18] 训练的 ViT-B/32 模型准确率相近，但规模缩小 2.8 倍，速度提升 3 倍；
- MobileCLIP-S2 的 38 个数据集平均性能较 DataComp [18] 训练的 ViT-B/32-256 模型（训练时长 2.6 倍）提升 2.8%，检索性能显著更优，且规模缩小 1.5 倍，速度提升 1.4 倍；
- MobileCLIP-B 的 38 个数据集平均性能较 SigLIP-B/16 [77]（WebLI 数据集训练时长 3 倍）提升 2.9%，检索性能更优，且规模缩小 26.3%。

5.5 检索性能分析

在最新 ARO 基准 [75] 上评估模型性能，表 8 对比了 MobileCLIP-B（基于 DataCompDR-1B 训练）与所有公开 ViT-B/16:Base 模型的性能。实验表明，仅针对零样本分类或检索优化（基于网页级噪声数据集），会降低模型对自然场景的组合理解能力；而 DataCompDR 在保证零样本分类与检索性能的同时，显著提升模型在 ARO 基准上的表现：与 SigLIP [77] 相比，MobileCLIP-B 在 Visual Genome Relation 和 Attributes 数据集上的准确率分别提升 19.5% 和 12.4%，在 Flickr30k-Order 和 COCO-Order 数据集上的 Recall@1 分别提升 69.7% 和 50.3%。

6. 结论

本文提出适用于端侧 CLIP 推理（低延迟、小尺寸）的对齐图文骨干网络 MobileCLIP，同时提出 DataCompDR—— 通过图像描述生成预训练模型和 CLIP 强集成模型知识强化的 DataComp 数据集。实验验证，基于 DataCompDR 的训练学习效率提升 10 倍至 1000 倍；MobileCLIP 模型在延迟 - 准确率权衡上实现当前最优，且在鲁棒性和 ARO 基准性能上表现更优。

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					63.3	48.5	58.8	78.9	30.4	50.1	52.5
LAION-B/32	LAION-2B [52]	32B	ViT-B/32	Base	86.2 + 63.4	5.9 + 3.3	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		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	ViT-B/16	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 Appendix I.

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].