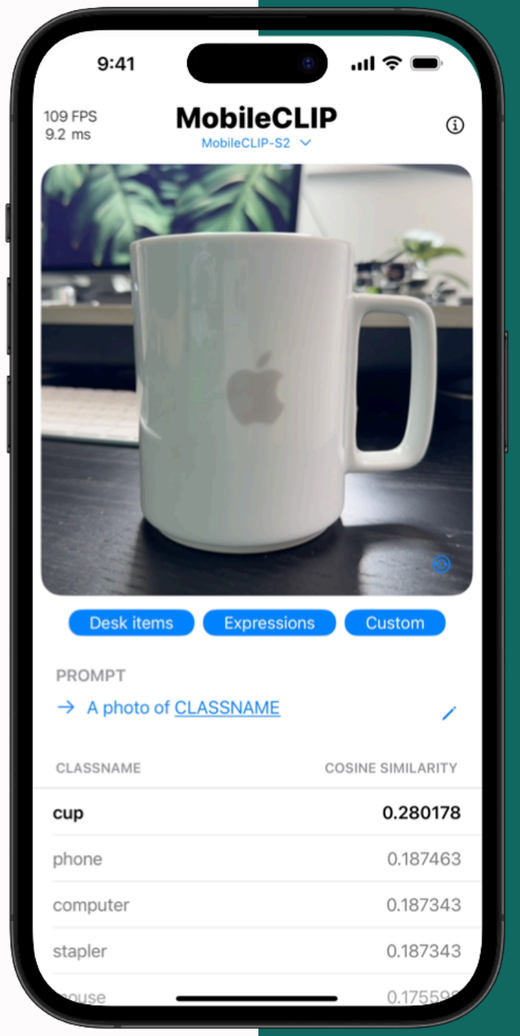
Emanuele Poiana
Ettore Saggiorato

Advanced Computer Vision

MobileCLIP

arXiv:2311.17049



Overview

Objective

Design a new family of aligned image-text encoders suitable for **mobile devices.**

01

Small

‼ Needs to run on edge devices.

02

Low Latency

‼ Needs to be fast.

03

Reduced capability of smaller models

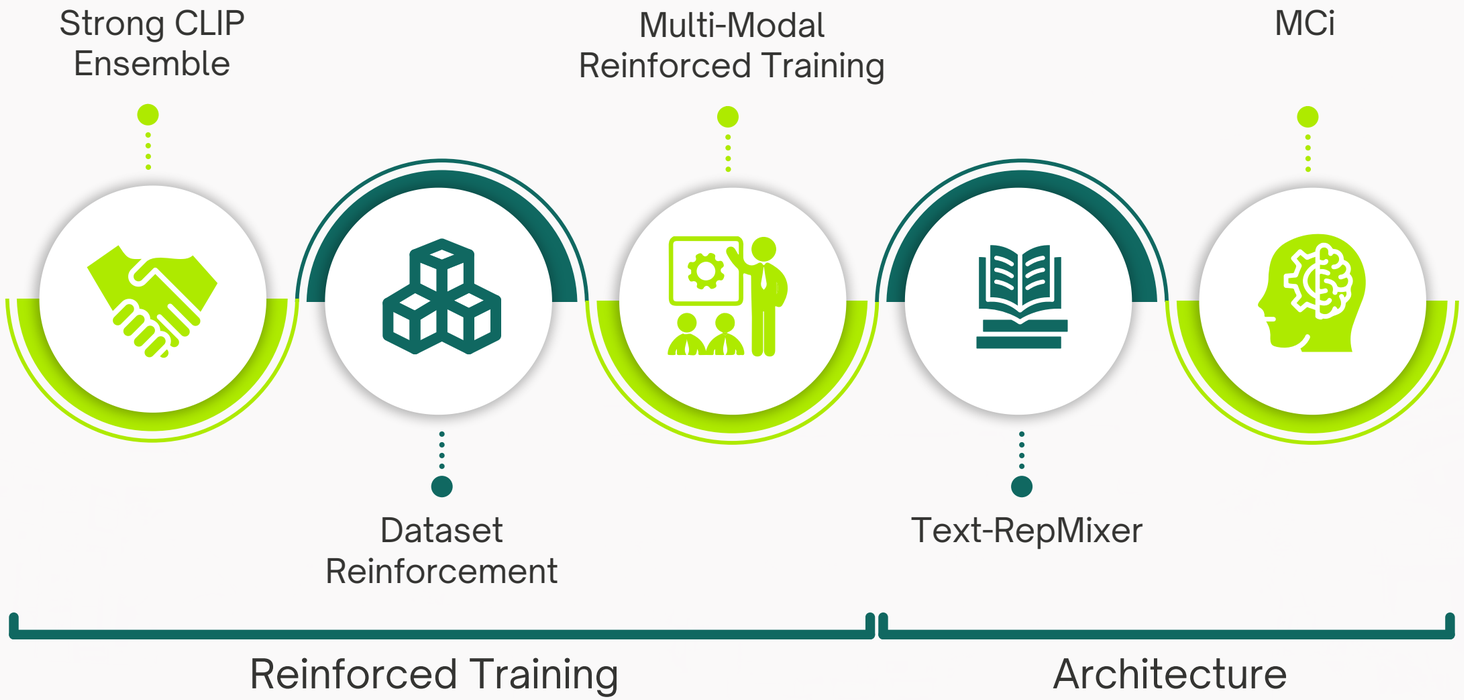
⚠ Can be improved with a better training method.

04

Training is expensive

⚠ Rapid development and exploration of efficient architecture designs is hard. Need for better training efficency.

Summary


Strong CLIP Ensemble
Multi-Modal Reinforced Training
MCi
Dataset Reinforcement
Text-RepMixer
Reinforced Training
Architecture

Strong CLIP Ensemble

Motivation

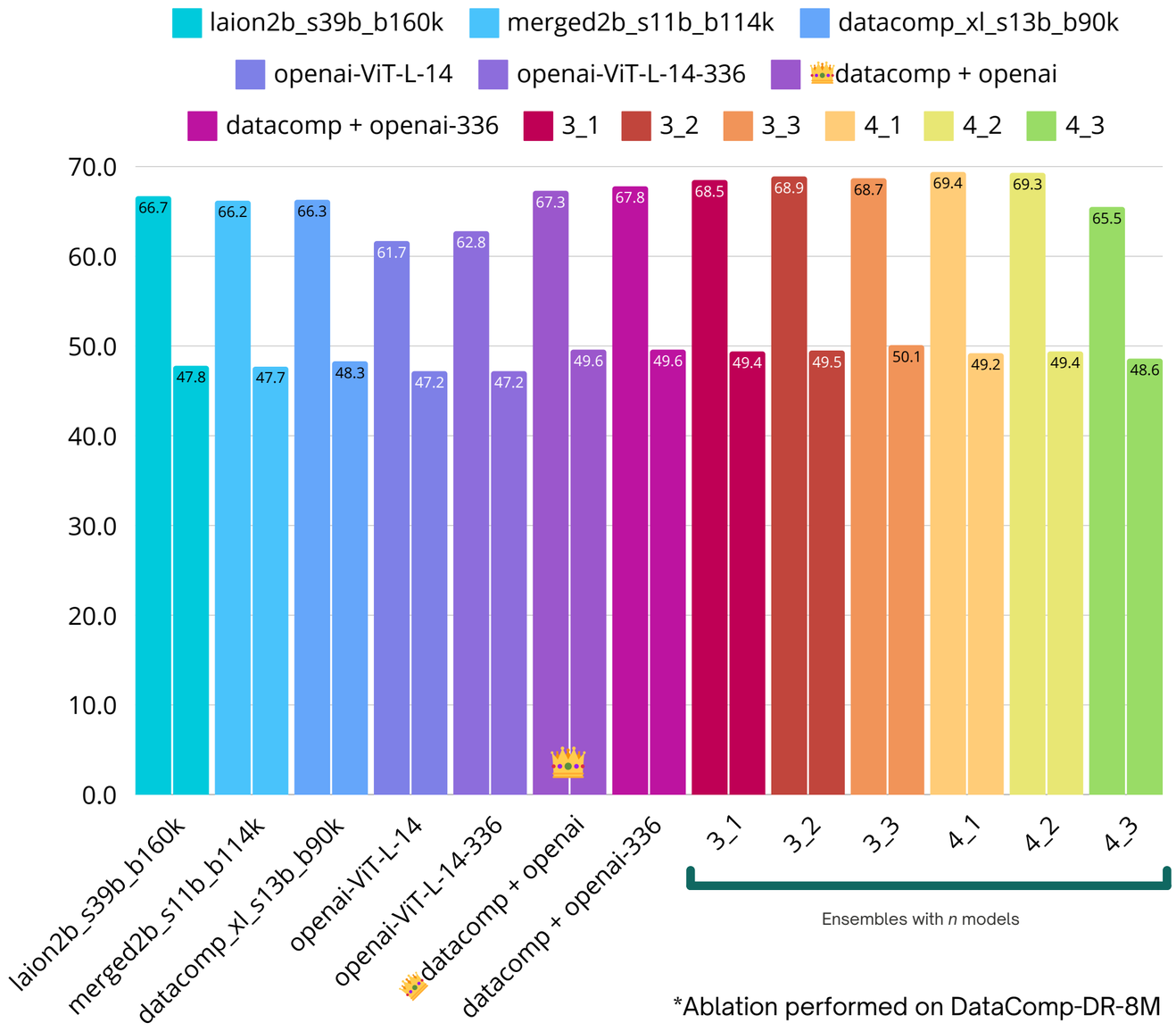
To train MobileCLIP in a knowledge distillation scenario we need a good teacher. Here comes the CLIP ensemble.

They have made ablation studies which observe that more accurate CLIP models are not necessarily better teachers.

Ensemble

Based on ViT-L-14.

- openai-ViT-L-14
- datacomp_xl_s13b_b90k-ViT-L-14



👑: selected ensemble

*Ablation performed on DataComp-DR-8M

Multi-Modal Reinforced Training

1

Reinforced Dataset

Expand DataComp to improve training efficiency.

2

Efficient Training

Fundamental to reduce costs and time between experiments.

3

Loss Function

Manages CLIP and knowledge distillation losses.

Multi-Modal Reinforced Training

1. Dataset Reinforcement

Real caption:

"One of the replacement Fairfax stones"

Synthetic caption:

"a large ston 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 caption:

"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"

Idea

- Reinforce a dataset once (improve accuracy)
- Store and use it for multiple experiments
- Base dataset: **Data-Comp**

Innovation

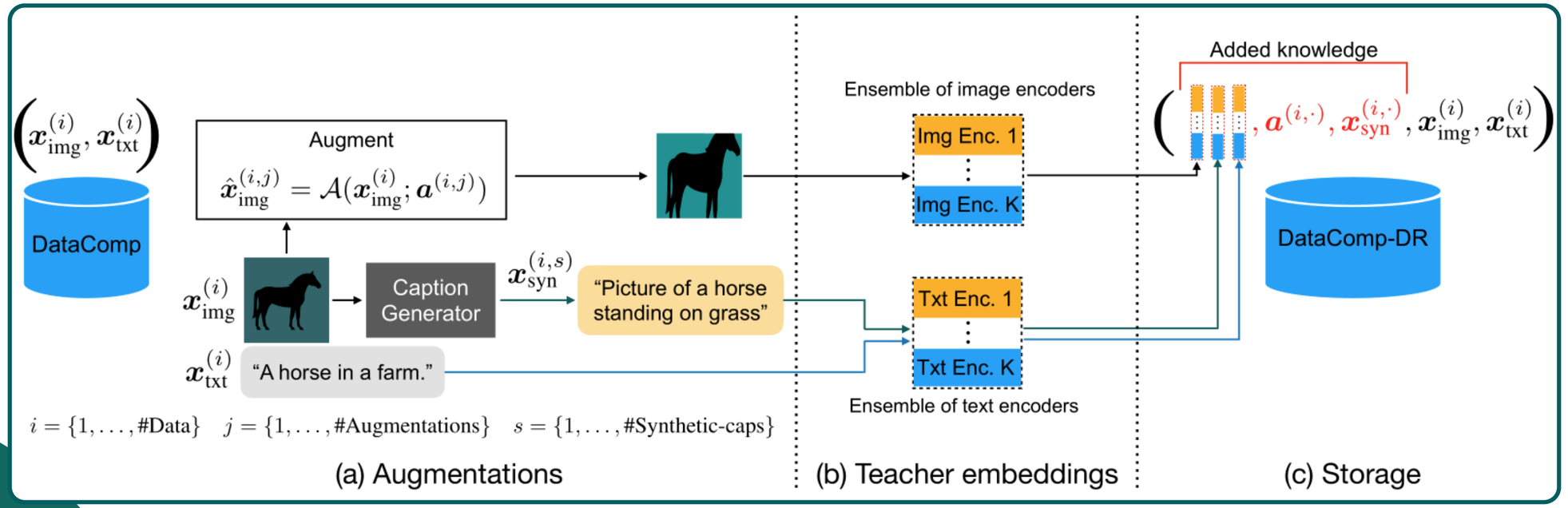
- Image Augmentation (**RRC, RA**)
- Synthetic captions (**CoCa**)
- Store embeddings from a **strong ensemble** to avoid adding computational burden

Generated

- Data-CompDR-12M
- Data-CompDR-1B

Multi-Modal Reinforced Training

1. Dataset Reinforcement



Multi-Modal Reinforced Training

2. Efficient Training ⚡

- Adds **no training time overhead**
- As seen before **Data-CompDR** improves accuracy

1

From DataComp-DR

Load:
- image $x_{img}^{(i)}$
- ground-truth caption $x_{txt}^{(i)}$

Randomly load:
- aug. param. $a_{img}^{(i,j)}$
- syn. caption $x_{syn}^{(i,s)}$

2

Reproduce

Synthetic image $\hat{x}_{img}^{(i)}$ from the augmentation parameters.

3

From DataComp-DR

Read:
- img. embedding $\psi_{img}^{(i,j,k)}$
- txt. embedding $\psi_{txt}^{(i,k)}$
- syn. txt. emb. $\psi_{syn}^{(i,j,k)}$

for *K* teacher models.

4

Constuct data batches

Two data batches:

$\mathcal{B}_{real}$ real image, real caption

$\mathcal{B}_{syn}$ aug. image, syn. caption

$$\mathcal{L}_{Final} = \sum_{\mathcal{B} \in \{\mathcal{B}_{real}, \mathcal{B}_{syn}\}} \mathcal{L}_{Total}(\mathcal{B})$$

Multi-Modal Reinforced Training

3. Loss 🤦

- Standard CLIP loss
- *Distill:* distillation loss parametrized by λ (*0.7-1.0*) which discriminate between the two losses

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

Ablation on the loss. The tradeoff between IN-val and Flickr30k is controlled by the synthetic sampling and loss coefficient. Trained for 30k iterations.

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

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

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

Where:

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

Architecture

1

Text Encoder

FastViT on steroids

2

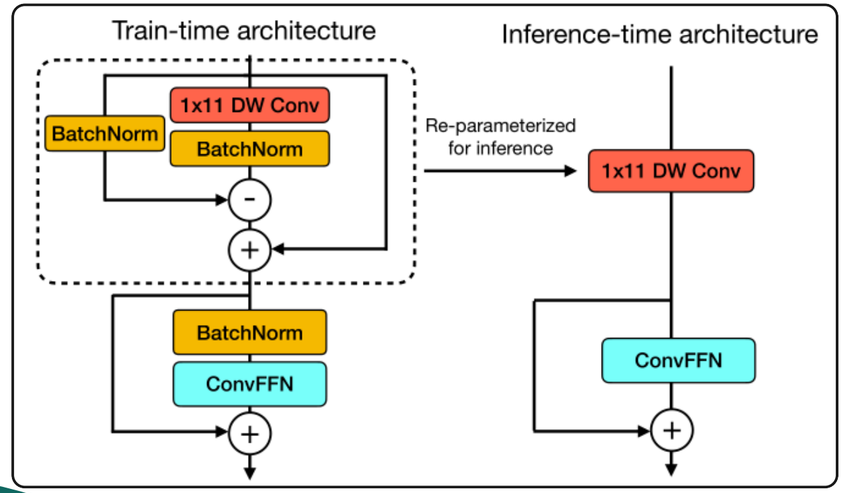
Image Encoder

Faster, deeper FastViT

Architecture

1. Text Encoder

Text-RepMixer



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

Idea

- Use 1D convolutions
- Use self-attention layers
- Inspired by RepMixer for efficiency
- Based on CLIP's text encoder
- Decouple train/inference architectures

Innovation

- Replace convolution with self-attention
- Smaller then FastViT's
- Reparameterization

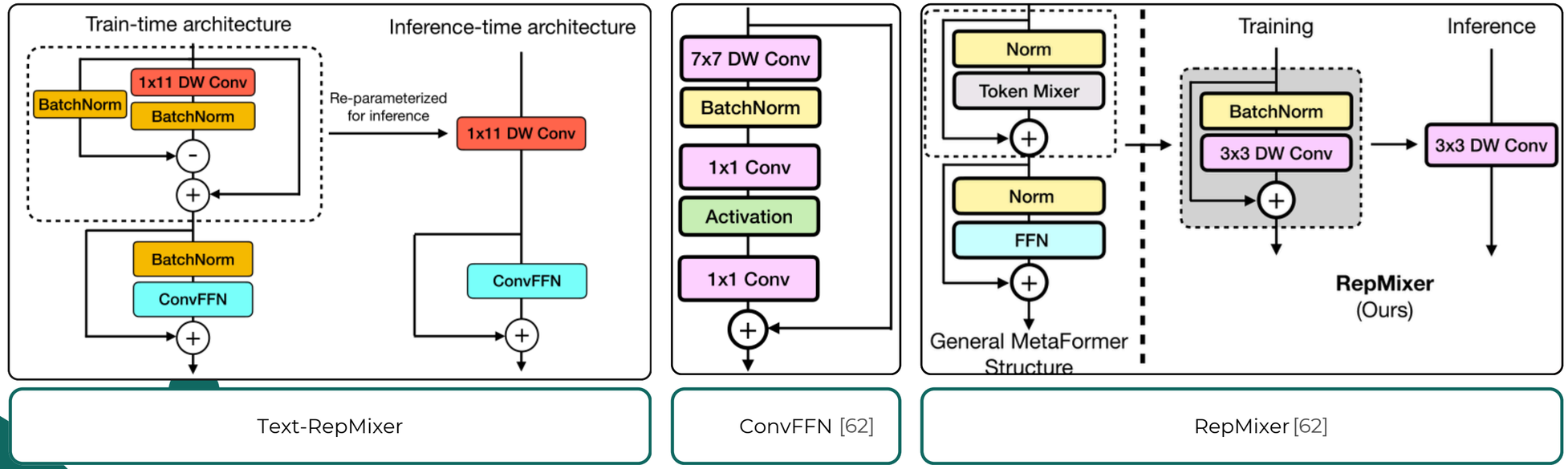
Results

Text Enc.	Latency	0-shot IN-Val
Base	3.3	53.4
MCt (MobileClip)	**1.6**	**53.6**

Architecture

1. Text Encoder

Text-RepMixer



Text-RepMixer | ConvFFN [62] | RepMixer [62]

[62]: P. Vasu, J. Gabriel, J. Zhu, O. Tuzel, and A. Ranjan. Fastvit. arXiv:2303.14189, 2023.

Architecture

2. Image Encoder

MCi

Idea

- Based on FastViT
- Improve the parameter efficiency by lowering the expansion ratio + increasing depth

Results

Image Enc.	Latency	0-shot IN-Val
FastViT-MA36	4.3	58.9
MCi2 (MobileClip)	**3.6**	**60.0**

Results

1

Reinforced Dataset

More efficient training at no additional cost

2

Text Encoder

FastViT on steroid

3

Small Scale

Faster, more accurate

4

Efficiency

So fast, much wow

5

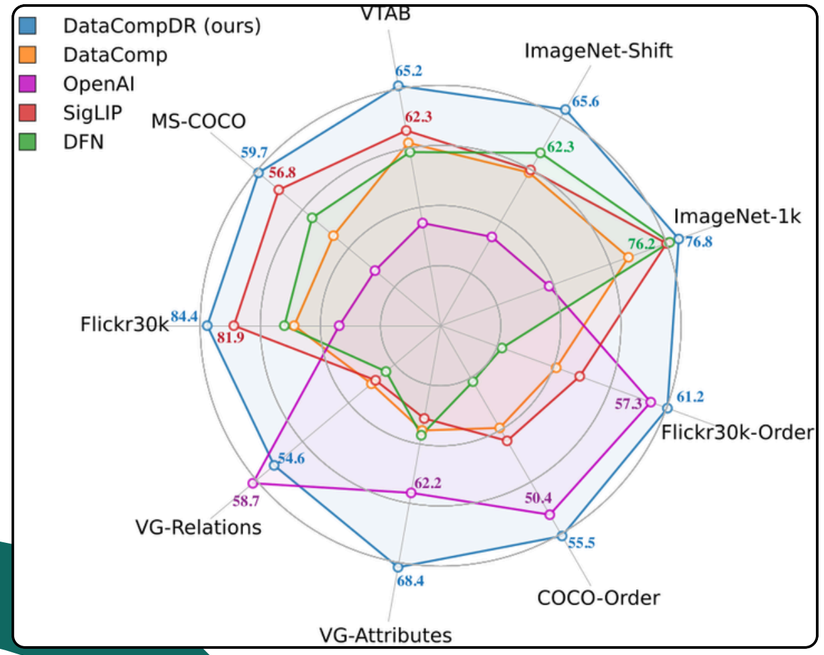
SOTA

Performs pretty well

Results

DataComp-DR

in a nutshell



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

Image Enc.	Dataset	# Image Enc. Params (M)	Latency (ms) (img+txt)	0-shot IN-val	Δ
MobileNetv3-L	DataComp-12M DataCompDR-12M (Ours)	4.9	1.1 + 3.3	34.1 **44.7**	↑+10.6
ViT-T/16	DataComp-12M DataCompDR-12M (Ours)	5.6	3.0 + 3.3	32.9 **44.1**	↑+11.2
ResNet-50	DataComp-12M DataCompDR-12M (Ours)	24.6	2.6 + 3.3	40.4 **51.9**	↑+11.5
FastViT-MA36	DataComp-12M DataCompDR-12M (Ours)	43.5	4.3 + 3.3	45.2 **58.9**	↑+13.7

DataCompDR-12M vs. DataComp-12M. All the models are trained for 30k iterations (0.24B seen samples).

Results

DataComp-DR 🚀

in a nutshell

Accuracy Improvement

On different architectures on DataComp and DataComp-DR

Training Time

Using DataComp and DataComp-DR, but better efficiency. Bonus: plug and play setup.

Dataset	Size (TBs)	Time (hours)
DataComp-12M	0.9	1.3
DataCompDr-12M	1.9	1.3
DataComp-1B	90	-
DataCompDR-1B	140	-

Trained on 8xA100.

Results

Hybrid Text Encoder ⚡

in a nutshell

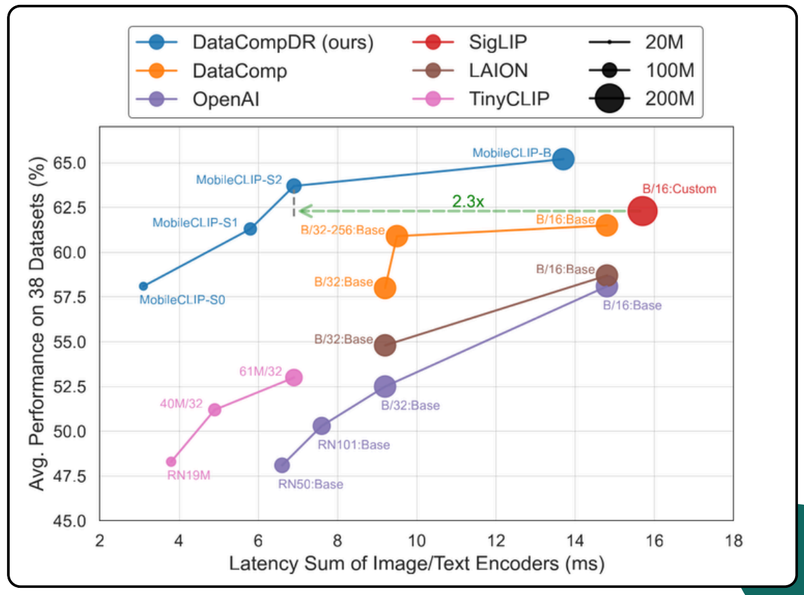
Smaller*

With

- 2 blocks of Text-RepMixer
- 4 blocks of self-attention layers

Faster*

*wrt pure transformer variant



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

Ablation study. Effect of the number of self-attention layers. Trained for 30k iterations.

Results

Small Scale Regime

in a nutshell

Comparison with ~12-20M samples

MobileClip is more accurate

Compared with SLIP [43] and CLIPA [34] while using less samples -> better scaling

MobileClip is faster

Compared with TinyCLIP [68]: it has lower latency and higher accuracy.

Name	Dataset	Seen Samples	Latency (ms) (img+txt)	Zero-shot IN-val
CLIP-B/16 [43, 47]	CC-12M [4]	0.39B	11.5 + 3.3	36.5
CLIP-B/16 [43, 47]	YFCC-15M [57]	0.37B		37.6
MobileCLIP-B	CC-12M [4]	0.37B	10.4 + 3.3	38.1
SLIP-B/16 [43]	CC-12M [4]	0.39B	11.5 + 3.3	40.7
SLIP-B/16 [43]	YFCC-15M [57]	0.37B		42.8
MobileCLIP-B	DataComp-12M [18]	0.37B	10.4 + 3.3	50.1
MobileCLIP-B	DataCompDR-12M	0.37B	10.4 + 3.3	**65.3**
CLIP-B/32 [7, 47]				32.8
SLIP-B/32 [7, 43]				34.3
FILIP-B/32 [7, 72]	YFCC-15M [57]	0.49B	5.9 + 3.3	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

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.

[43]: N. Mu, A. Kirillov, D. Wagner, S. Xie - SLIP: Self-supervision meets Language-Image Pre-training - arxiv:2112.12750 - 2021
[34]: X. Li, Z. Wang, C. Xie - An Inverse Scaling Law for CLIP Training - 2305.07017 - 2023
[68]: Kan Wu et al. - TinyCLIP: CLIP Distillation via Affinity Mimicking and Weight Inheritance - arxiv:2309.12314 - 2023

Results

Learning Efficiency

in a nutshell

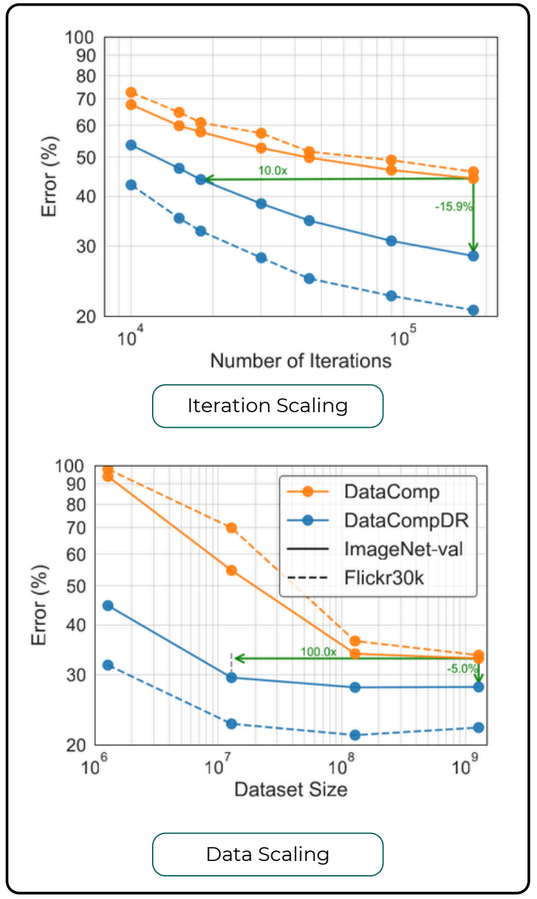
Longer Training

Training for 120 Epochs, with 12M samples on DataComp-1B the reinforced training strategy outperforms the non-reinforced training (71.7% vs 55.7%).

Efficient Scaling

Subsets of DataComp-1B from 1.28M to all 1.28B samples

- With 1.28M samples
  - DataComp-DR: 55.2% acc
  - DataComp: ~6% acc.
- 100× data efficiency



Training on DataCom-DR 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.

Results

Comparison with SOTA 💪

in a nutshell

Outperforms TinyCLIP

By ~6% in acc.(also they say that have not truly test TinyClip in a good way)

Performs similar to ViT-B/32

Trained on DataComp, while being

- 2.8x smaller
- 3x faster

Outperforms SigLIP-B/16 on 38 datasets

- MobileCLIP has 2.9% better avg. acc. While being 26.3% smaller.
- SigLIP-B/16 was trained for ~3x longer on WebLI dataset.

Better performance on ARO benchmark

- Attribute, Relation and Order benchmark
- Better performances and robustness

Method	Dataset	IN-val zero-shot	VG Rel.	VG Attr.	COCO Order	Flickr30k Order
CLIP	OpenAI-400M [47]	68.3	**58.7**	62.2	<u>50.4</u>	<u>57.3</u>
CLIP	LAION-2B [52]	70.2	39.7	<u>62.3</u>	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]	<u>76.2</u>	33.1	57.4	18.5	22.5
MobileCLIP-B	DataCompDR-1B	**76.8**	<u>54.6</u>	**68.4**	**55.5**	**61.2**

Performance on ARO benchmark. All the models use ViT-B/16 as image encoder and the Base text encoder.

Improvements !?💀

1

Reinforced Dataset

Caption evaluation and filtering

2

Sigmoid Self-Att.

Higher efficiency, similar acc.

3

TinyCLIP on MobileCLIP

Loss Integration (not possible)

4

PuMER

Token Pruning and Merging in VILT

Training

Inference

Possible improvements

Caption Generation

Idea

- Generated captions may be equal, or too similar
- Regenerate these captions

Method

- Calculate "*similarity score*" with:
  - CLIPScore [2] (image-text alignment)
  - BERTScore [3] (text similarity + fluency)
  - Diversity penalty (prevent repeated capt.)

Expected Results

Little accuracy and efficiency gain.

	MSCOCO				NoCaps			
					Valid		Test	
	B@4	M	C	S	C	S	C	S
CLIP-ViL [73]	40.2	29.7	134.2	23.8	-	-	-	-
BLIP [37]	40.4	-	136.7	-	113.2	14.8	-	-
VinVL[27]	41.0	31.1	140.9	25.4	105.1	14.4	103.7	14.4
SimVLM [16]	40.6	33.7	143.3	**25.4**	112.2	-	110.3	14.5
LEMON [80]	**41.5**	30.8	139.1	24.1	117.3	15.0	114.3	14.9
LEMON$_{SCST}$ [80]†	42.6	31.4	145.5	25.5	-	-	-	-
OFA [17]†	43.5	31.9	149.6	26.1	-	-	-	-
CoCa	40.9	**33.9**	**143.6**	24.7	**122.4**	**15.5**	**120.6**	**15.5**

Image captioning results on MSCOCO and NoCaps. **CoCa** was finetuned only on MSCOCO. [1]

Bad Ideas

- Finetune CoCa (RL) on DataComp
  - expensive
  - gains may not justify the expense
- FLEUR [4]: LLM based
  - performance vs cost
  - CLIPScore: 20ms/img, FLEUR: 700ms/img

[1]: J. Yu, Z. Wang, V. Vasudevan and L. Yeung, M. Seyedhosseini, Y. Wu - CoCa: Contrastive Captioners are Image-Text Foundation Models - TSLR - arxiv:2205.01917, 2022
[2]: J Hessel, A. Holtzman, M. Forbes, R. Le Bras, Y. Choi - CLIPScore: A Reference-free Evaluation Metric for Image Captioning- EMNLP 2021 - arxiv:2104.08718, 2022
[3]: T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, Y. Artzi - BERTScore: Evaluating Text Generation with BERT - ICLR2020 - arxiv:1904.09675, 2020
[4]: L. Yebin, P. Imseong, K. Myungjoo, FLEUR: An Explainable Reference-Free Evaluation Metric for Image Captioning Using a Large Multimodal Model, ACL, 10.18653/v1/2024.acl-long.205, 2024

Possible improvements

Sigmoid Self-Attention

Idea

- Substitute softmax self-attention with sigmoid self-attention

Method

- Close to be plug-and-play

Difficulties

- Initial norms stabilization
  - “*The central problem with naïve sigmoid attention is that of large initial attention norms*”
- Under research

Expected Results

- Faster training and inference on dedicated hardware (GPUs)
  - it has a *~17% i*nference kernel speed-up over FLASHATTENTION2 [2] on H100 GPUs, *~10%* on A100
- Lower latency (higher throughput)
- Higher memory efficiency
- Similar model performance (acc.) wrt softmax self-attention

[1]: J. Ramapuram, F. Danieli, E. Dhekane, F. Weers, D. Busbridge, P. Ablin, T. Likhomanenko, J. Digani, Z. Gu, A. Shidani, R. Webb - Theory, Analysis, and Best Practices for Sigmoid Self-Attention- arxiv:2409.04431, 2025
[2]: T. Dao - FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning - ICLR - arxiv:2307.08691 - 2023

Possible improvements

Model Training ✗

Idea

1. Integrate TinyCLIP weight inheritance
2. Distill MobileCLIP with TinyCLIP strategy

Problems

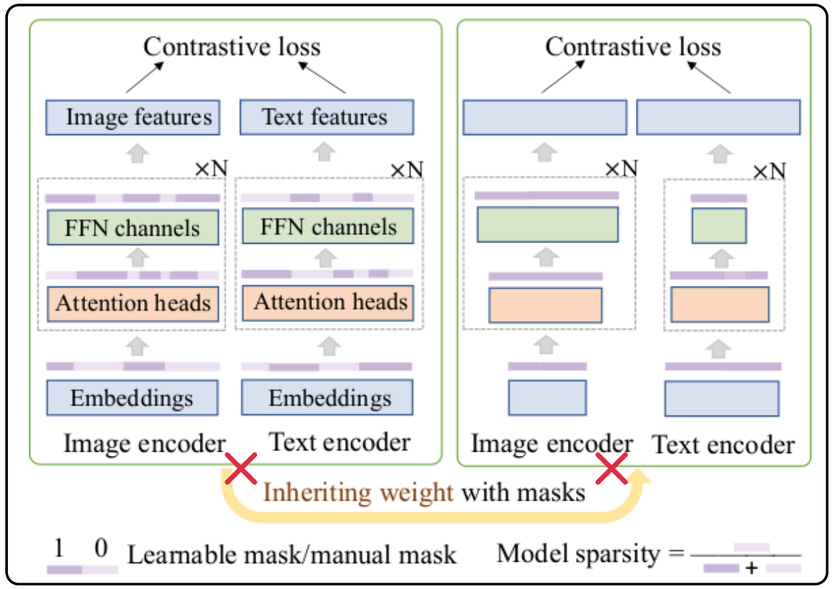
- Integration
  - MobileCLIP doesn't have the teacher weights
  - Straightforward integration of TinyCLIP method is not directly applicable
- Distillation
  - Reducing the parm. even further leads to lower performances
  - Possible to do but we need to train another model where the strong assumption of TinyCLIP is re-utilize the weights of SA and FFN layers, which is what MobileCLIP is already reducing

Expected Results

- Deterioration of the model generalization.
- Possibly wasting resources on more distillation

$$L = L_{distill} + L_{sparsity}$$

$$L_{sparsity} = \lambda * (p - q) + \beta * (p - q)^2$$



✗Integrate TinyCLIP is not feasible with DataComp-DR

[1]: J. Ramapuram, F. Danieli, E. Dhekane, F. Weers, D. Busbridge, P. Ablin, T. Likhomanenko, J. Digani, Z. Gu, A. Shidani, R. Webb - Theory, Analysis, and Best Practices for Sigmoid Self-Attention- arxiv:2409.04431, 2025
[2]: T. Dao - FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning - arxiv:2307.08691 - 2023

Possible improvements

Model Inference

Ideas

- As suggested in the paper, an integration of PuMer might improve performance without compromising performance
- PatchRanking is a powerful plug in for reducing the token number in CLIP, joint with the MAM like in PuMer

Main Problems

- PuMer is thought for VILT where they have a shared token mixing mechanism in the cross-modal encoder
- The hybrid transformer-CNN architectures that are deployed present already few SA layers

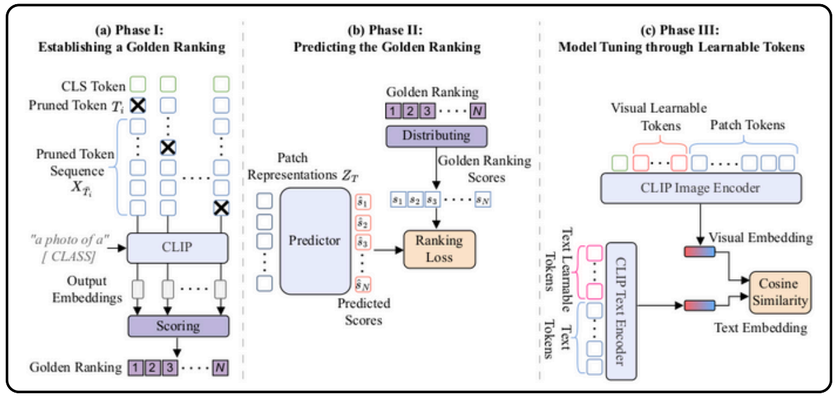
Expected Results

- **Image Encoder**
  - not much of a difference for the cost that could have (only 1 SA layer)
- **Text Encoder**
  - might be possible to do more (6 SA layers)
- Training the Patch Ranking predictor might not result in a considerable latency reduction/accuracy tradeoff

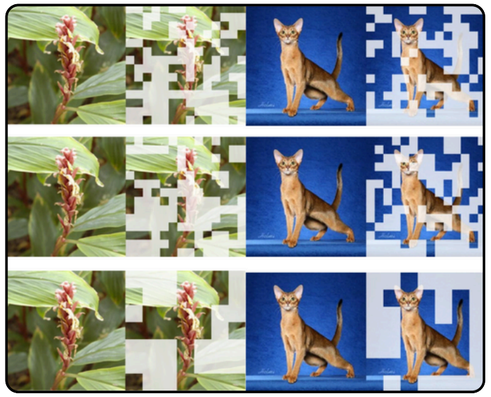
[1]: Q. Cao, B. Paranjape, H. Hajishirzi - PuMer: Pruning and Merging Tokens for Efficient Vision Language Models - ACL 2023 Main Conference - arxiv:2305.17530 - 2023
[2]: C. Wu, J. Lin, Y. H. Hu, P. Morgado - Patch Ranking: Efficient CLIP by Learning to Rank Local Patches - WACV 2025 - arxiv:2409.14607 - 2024

Possible improvements

Model Inference



Patch Ranking: MobileCLIP architecture doesn't allow Phase 3

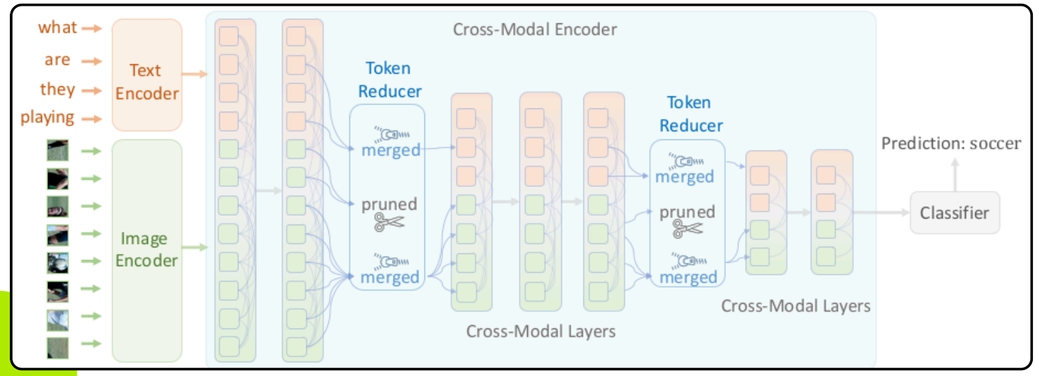


Golden Ranking differences

Label Driven Score

Maximum Confidence Score

Feature Preservation Score



PuMer implementation - in CLIP the Cross-Modal Encoder is missing

[1]: Q. Cao, B. Paranjape, H. Hajishirzi - PuMer: Pruning and Merging Tokens for Efficient Vision Language Models - ACL 2023 Main Conference - arxiv:2305.17530 - 2023
[2]: C. Wu, J. Lin, Y. H. Hu, P. Morgado - Patch Ranking: Efficient CLIP by Learning to Rank Local Patches - WACV 2025 - arxiv:2409.14607 - 2024

Improvements

Summary

1

Reinforced Dataset

Caption evaluation and filtering

- Small perf. and acc. improvement

2

Sigmoid Self-Att.

- Under research
- Faster inference/train
- Similar acc.

3

TinyCLIP Integration

- MobileCLIP distillation
- May result in too small models - underfit

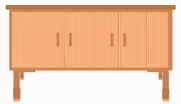
4

PuMER

- Custom token pruning and merging, PuMer inspired
- Lower latency
- Need to train patch ranking predictor

Training

Inference



+

Emanuele Poiana
Ettore Saggiorato

Thanks for your attention

Q&A?

MobileCLIP

arXiv:2311.17049