MoPE-CLIP: Structured Pruning for Efficient Vision-Language Models with Module-wise Pruning Error Metric

Supplementary Material

A. Related work

Vision-Language Pre-trained Models. Benefiting from the efficiency of contrastive learning, vision-language pre-trained models like [5, 10, 14, 18, 26, 39, 58, 59, 64] have achieved advanced capability across downstream tasks. Such dual-stream models have efficient inference speed on multi-modal tasks like retrieval, as the image/text features can be computed offline [9, 57]. However, these models are often pre-trained with millions or billions of image-text pairs from scratch, which is computationally expensive [7, 45, 46]. Later works [25, 27, 35] propose to use more complex objectives to reduce the amount of pre-training data. Others [21, 56] intend to reduce the influence of noisy and unmatched image-text pairs. However, these methods lead to less competitive retrieval performance. In this work, we show that we can prune the original pre-trained CLIP to a desired size and significantly lift up the performance of the pruned model in a data-efficient way, i.e., with several magnitudes fewer pertaining data than the original CLIP.

Pruning of Transformer-based Models. Various methods have been proposed to compress uni-modal vision and language transformer models [6, 24, 29, 31, 50, 52, 54, 63]. Among them, structured pruning methods remove unimportant structured components (e.g., attention heads, FFN neurons, and Transformer layers) in the network. Depending on how the pruned components are determined, pruning methods could be divided into two categories: search-based and metric-based methods. Search-based methods [4, 19, 48] usually apply masks on the structured components and need a searching process to determine their importance. On the other hand, metric-based methods apply various metrics to determine module importance and result in a single-shot pruning process. Widely used metrics include the magnitude of weight [15, 16, 62, 65] and the variant in loss [31, 33, 34]. Some researchers [12, 40] explore different strategies for pruning BERT layers, such as "every other", "bottom or top dropping" and "search on valid" like CNN Oracle Filter Pruning [1, 32]. Notably, the "every other" strategy has been proven effective [12, 40], with DynaBERT [17] implementing it to create dynamic depth networks. Additionally, pruning is often used in combination with knowledge distillation, which transfers knowledge from the original unpruned teacher model to the smaller pruned model with different kinds of knowledge [41, 47, 54].

In contrast to the extensive research on compressing uni-modal Transformer-based models, compression of multi-modal models remains under-explored. Our experiments show that directly using widely-used metrics [15, 31] or "every other" strategy [12, 40] for VLP pruning leads to unsatisfactory performance, indicating the demand for exploring more accurate metrics to measure module importance of VLP models across multi-modal tasks. Recently, EfficientVLM [51] proposes to distill the VLP model in the pre-training stage and then prune attention heads during the task-specific fine-tuning stage, but the distillation stage proved not optimal in our experiments. Another work Upop [45] uses a unified and progressive search-based pruning method on vision-language models, but the search process is expensive and is hard to apply to the pre-training stage. TinyCLIP [53] proposes a multi-stage pruning and distillation method for pre-training small OpenCLIP models [7]. However, the design of the multi-stage is complex and the final performance relies on the huge pre-training dataset LAION400M [42]. In this work, we propose a simple but effective metric called MoPE, which serves as a general importance measure of various compressible components like attention heads, FFN neurons, and Transformer layers. Based on MoPE metric, we design a unified pruning framework applied to both the pre-training and fine-tuning stages, resulting in state-of-the-art MoPE-CLIP models.

B. Implementation Details

B.1. Detailed Experimental Settings

Here we describe detailed setups. For all experiments, we use the same random seed (e.g., 42). All pre-training or fine-tuning processes utilize 8x Nvidia V100 GPUs.

Details for Evaluation Benchmarks. For retrieval tasks, we split the MSCOCO [28] and Flickr30K [38] datasets following [20]. For classification tasks, we adopt 11 downstream datasets following [60, 61], including CIFAR10, CIFAR100 [23], Caltech101 [13], Flowers102 [36], Oxford Pets [37], DTD [8], Stanford Cars [22], FGVC Aircraft [30], SUN397 [55], Food101 [3] and ImageNet [11].

Details for Fine-tuning Stage Compression Table B1 summarizes the hyperparameters for fine-tuning CLIP-ViT-L/14 and distilling CLIP-VIT-B/32. During the distilling process, we first fix the model and train the linear layer for 5 epochs with a learning rate of 1e-5 to learn a better mapping function. Table B2 lists the detailed retraining setups for MagnCLIP, DynaCLIP, MoPE-CLIP, and SE-CLIP in

Config	Fine-tuning	Distilling
Optimizer	AdamW, $\beta = (0.9, 0.98)$	
LR schedule	CosineLRScheduler	
Weight decay	3e-4	
Warmup ratio	0.1	
Init LR	3e-6	1e-6
Batch size	256	1024
Training epochs	12	15
Distillation	N/A	$\mathcal{L}_{sim} + \mathcal{L}_{feat}$

Table B1. Experimental setup for fine-tuning CLIP-VIT-L/14 or distilling CLIP-ViT-B/32.

Downstream Task	Image-to-text	Text-to-image
Optimizer	AdamW, $\beta = (0.9, 0.98)$	
LR schedule	CosineLRScheduler	
Weight decay	3e-4	
Warmup ratio	0.1	
Init LR	2e-5	8e-5
Batch size	256	1024
Training epochs	20	10

Table B2. Experimental setup for retraining MagnCLIP, DynaCLIP, MoPE-CLIP and SE-CLIP across TR and IR tasks.

Config	Pre-training	Further Fine-tuning
Optimizer	AdamW, $\beta = (0.9, 0.98)$	
LR schedule	CosineLRScheduler	
Weight decay	3e-4	
Warmup ratio	0.02	0.1
Init LR	5e-5	4e-5
Batch size	512	512
Training epochs	20	15

Table B3. Experimental setup for pre-training DynaCLIP and MoPE-CLIP and further fine-tuning on downstream tasks.

image-to-text retrieval (TR) and text-to-image retrieval (IR) tasks The text encoders of these models are fixed for the TR task, while image encoders are frozen for the IR task. For SE-CLIP, we add a linear layer to align feature space, and the hidden distillation loss is excluded due to the unmatched number of image patches.

Details for Pre-training Stage Compression We list the detailed setup for pretraining stage compression in Table B3. MoPE-CLIP adopts the Recall Mean on MSCOCO validation dataset as the specific MoPE metric. DynaCLIP and MoPE-CLIP share the same hyperparameters.

B.2. Main Algorithm

We illustrate the computation process of the MoPE metric in Algorithm 1, and our unified pruning framework resulting in MoPE-CLIP in Algorithm 2.

Algorithm 1 Module-wise Pruning Error Metric

Input: CLIP model $\boldsymbol{f_\varphi}$, Module $\boldsymbol{\theta}$, Dataset $\boldsymbol{\mathcal{D}}$
Output: Importance of $\boldsymbol{\theta}$

procedure MoPE ($\boldsymbol{f_\varphi, \theta, \mathcal{D}}$):
  Compute the full CLIP Performance on $\boldsymbol{\mathcal{D}}$: $\mathcal{Z}\left[f_\varphi\right]$
  Compute the CLIP$_{\boldsymbol{\theta=0}}$ Performance on $\boldsymbol{\mathcal{D}}$: $\mathcal{Z}\left[f_{\varphi-\theta}\right]$
  Compute the MoPE$_{\boldsymbol{\theta}} = \mathcal{Z}\left[f_\varphi\right] - \mathcal{Z}\left[f_{\varphi-\theta}\right]$
  return MPWE$_{\boldsymbol{\theta}}$
end procedure

Algorithm 2 MoPE-CLIP: Pruning with MoPE Metric

Input: CLIP model f_φ, Validation Set $\mathcal{D}_{val}$, Training Set $\mathcal{D}_{train}$
Output: MoPE-CLIP model

Partition the Attention Heads in $N \times L$ modules
for l in 1, ..., L **do**
  for head h in 1, ..., N **do**
    ▷ *run in parallel*
    MoPE$_{\boldsymbol{h}}$ ← MoPE(f_φ, h, $\mathcal{D}_{val}$)
    Update $\mathcal{C}_{head}$
  end for
end for
CLIP f'_φ ← Rewire Neurons in FFN by gradient
Partition the FFN Neurons in N groups
for group n in 1, ..., N **do**
  ▷ *run in parallel*
  MoPE$_{\boldsymbol{n}}$ ← MoPE(f'_φ, n, $\mathcal{D}_{val}$)
  Update $\mathcal{C}_{neuron}$
end for
if Compression in fine-tuning stage **then**
  MoPE-CLIPw f_{Cw} ← Prune the CLIP in width and retrain on $\mathcal{D}_{train}$
  for layer l in 1, ..., L **do**
    ▷ *run in parallel*
    MoPE$_{\boldsymbol{l}}$ ← MoPE(f_{Cw}, l, $\mathcal{D}_{val}$)
    Update $\mathcal{C}_{layer}$
  end for
  MoPE-CLIP ← Prune the MPEE-CLIPw in depth and retrain on D_{train}
else if Compression in pretraining stage **then**
  for layer l in 1, ..., L **do**
    ▷ *run in parallel*
    MoPE$_{\boldsymbol{l}}$ ← MoPE(f_φ, l, $\mathcal{D}_{val}$)
    Update $\mathcal{C}_{layer}$
  end for
  MoPE-CLIP ← Prune the CLIP in width and depth and retrain on D_{train}
end if
return the MoPE-CLIP

C. More Experimental Results

C.1. Detailed Comparison with Baselines

We provide a more detailed comparison with DynaCLIP$_V$, MagnCLIP$_V$, and UPop in the following.

Pruning Ratios. To further evaluate our MoPE-CLIP performance under different model sizes. We test six pruning ratios with the model performance plotted in Fig. C1. MoPE-CLIP consistently stands above all other baselines (i.e., DynaCLIP and MagnCLIP) across different pruning ratios. The gap becomes even larger for higher sparsities.

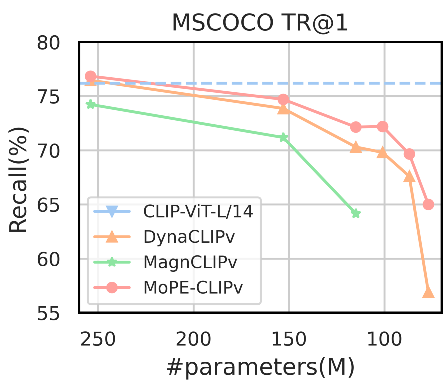

Figure C1. Comparsion of different pruning ratios.

Model	Params	MSCOCO (5K test set) TR @1	IR @1
UPop-Teacher	856M	71.5	56.8
UPop-CLIP	280M ↓ 67%	56.1 ↓ 21%	41.1 ↓ 27%
Upop-CLIP (+KD)	280M ↓ 67%	58.6 ↓ 18%	44.3 ↓ 22%
MoPE-Teacher	390M	76.2	58.8
MoPE-CLIP	**122M** ↓ 69%	**70.7** ↓ 7%	**54.7** ↓ 7%

Table C4. UPop and MoPE-CLIP on MSCOCO.

Relative Comparison with UPop. We further compare Upop with Knowledge Distillation in Tab. C4. MoPE-CLIP is superior to Upop (+KD) both on the relative performance drop and absolute task score, given a comparable relative decrease (69% vs 67%) in the number of parameters. Moreover, MoPE-CLIP's advantage is notable, as compressing smaller original model sizes is more challenging.

C.2. Fine-tuning Stage Compression on Flickr30K

To demonstrate the robustness of the MoPE metric across different data distributions, we further evaluate MoPE-CLIP on Flickr30K Dataset during fine-tuning stage compression.

Results for Image-to-text Retrieval. Following the setting in Section 4.1, we compress the vision encoder of fine-tuned CLIP-ViT-L14 (FT-L14) for image-to-text retrieval. We mainly compare the fine-tuned performance of our MoPE-CLIP$_V$ with fine-tuned CLIP-ViT-B/32 (FT-B32), DynaCLIP$_V$, and UPop-CLIP [45] on the Flickr30K dataset. In particular, we compute the loss gradient and MoPE metric (TR Mean) in Flickr30K [38] validation dataset for DynaCLIP$_V$ and MoPE-CLIP$_V$. The results are presented in Table C5. We could observe that once depth pruning is added to DynaCLIP$_V$, the TR@1 drops from 89.6% to 84.5%, while the MoPE-CLIP$_V$ with 87M vision encoder maintains competitive retrieval and surpasses the FT-B32. In addition, our MoPE-CLIP$_V$ with 115M vision encoder termed an entire model of 234M parameters outperforms the UPop-CLIP with 280M parameters by 8.2% TR@1. These results indicate the superiority of the MoPE metric across different downstream datasets.

Approach	Vision Encoder Wdith	Depth	Parmas	Flickr30K (1K test set) TR@1	TR@5	TR@10
Teacher Model	1024	24	304M	96.3	99.8	100.0
CLIP-ViT-B/32	768	12	88M	87.7	97.7	99.3
DynaCLIP$_V$ [17]	512	24	153M	**92.7**	99.4	99.8
	384	24	115M	89.6	98.5	99.4
	384	18	87M	84.5	97.3	98.5
UPop-CLIP [45]	N/A	N/A	474M‡	**93.2**	99.4	99.8
	N/A	N/A	280M‡	82.9	95.7	97.8
MoPE-CLIP$_V$	512	24	153M	**92.7**	**99.5**	**99.9**
	384	24	115M	**91.1**	**98.9**	**99.7**
	384	18	87M	**88.5**	**98.5**	**99.6**

Table C5. Image-to-text retrieval results on the Flickr30K dataset. The Params labeled as ‡ denote the parameters of the entire model.

Approach	Text Encoder Width	Depth	Params	Flickr30K (1K test set) IR @1	IR @5	IR @10
Teacher Model	768	12	85M	84.7	97.4	99.0
CLIP-ViT-B/32	512	12	38M	74.7	93.4	96.9
DynaCLIP$_T$ [17]	384	12	42M	84.1	97.1	98.7
	192	12	21M	80.3	95.7	98.0
MoPE-CLIP$_T$	384	12	42M	**85.1**	**97.4**	**99.1**
	192	12	21M	**83.5**	**97.2**	**98.8**

Table C6. Text-to-image retrieval results on the Flickr30K dataset. Pruning is applied in the width direction.

Results for Text-to-image Retrieval. We compress the text encoder of fine-tuned CLIP-ViT-L/14 for text-to-image retrieval. The pruning and retraining remain the same as the setting on the MSCOCO dataset and the results are illustrated in Table C6. The MoPE-CLIP$_T$ exhibits significant performance on the Flickr30K dataset. Even at a 4x compression ratio, the MoPE-CLIP$_T$ surpasses the FT-B32 by 8.8% IR@1 and DynaCLIP$_T$ by 3.2% IR@1. These superior results demonstrate that our MoPE-CLIP$_T$ provides a powerful text encoder for the text-to-image retrieval task.

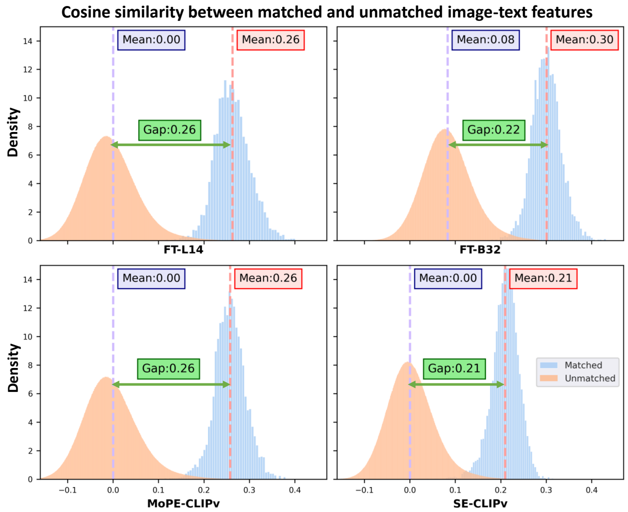

Figure C2. Histograms of cosine similarities between matched and unmatched image-text features. The green box represents the similarity gap. MoPE-CLIP$_V$ preserves a similar space to FT-L14.

C.3. Further Discussion.

Similarity matrix indicates pruning is the best architecture. We compare and analyze the similarity matrix of three architectures discussed in Section 2 since it directly influences retrieval performance. In particular, we sample 5k image-text pairs from the MSCOCO [28] validation dataset and calculate the similarities between matched image-text features and unmatched pairs, as done in previous works [49, 66]. Following [2], we suppose that the retrieval performance is more influenced by the similarity gap between matched and unmatched features. We compare the MoPE-CLIP$_V$ with fine-tuned CLIP-ViT-L/14 (FT-L14), fine-tuned CLIP-ViT-B/32 (FT-B32) and SE-CLIP$_V$. From Figure C2, we observe that FT-L14 has a larger gap between two similarities compared with FT-B32, reflecting its powerful performance. The pruned MoPE-CLIP$_V$ shows a similar distribution and gap to FT-L14, while the SE-CLIP$_V$ even closes the gap, indicating the performance difference among these models. Therefore, MoPE-CLIP$_V$, which preserves a similarity space like FT-L14, emerges as the best compact model architecture.

Grad-CAM demonstrates MoPE-CLIP preserves more important heads. To better understand the effect of our MoPE metric, we use Grad-CAM [44] to visualize the regions focused by DynaCLIP$_V$ and MoPE-CLIP$_V$. In detail, we select the model with a 115M vision encoder and compute the Grad-CAM using self-attention maps averaged over all attention heads in the last layer of the vision encoder. The gradients are acquired by contrastive loss $\mathcal{L}_{cont}$. From Figure C3, we could observe that the average attention map of MoPE-CLIP$_V$ is similar to original model (FT-L14), but the DynaCLIP$_V$ misses some important regions,

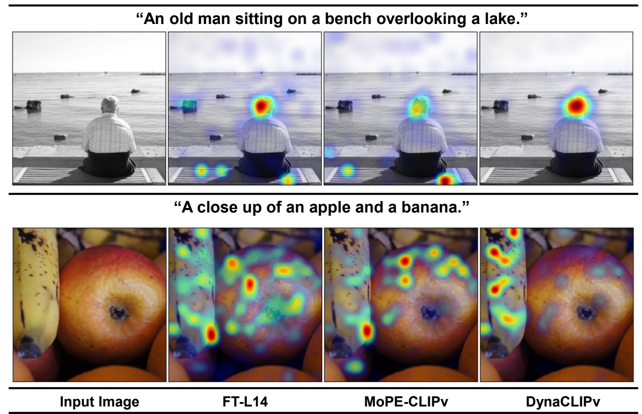

Figure C3. Grad-CAM visualization on the self-attention maps corresponding to the caption input.

Figure C4. Grad-CAM visualization of the last layer self-attention maps for original FT-L14's vision encoder. Red box denotes preserved heads based on MoPE-CLIP$_V$. Yellow box denotes preserved heads based on DynaCLIP$_V$. Orange box denotes the head is preserved by two models simultaneously.

like the "bench" in the top line and the "apple" in the bottom line. Furthermore, We visualize the Gram-CAM of each head of the FT-L14 model and identify the preserved heads by DynaCLIP$_V$ or MoPE-CLIP$_V$. As shown in Figure C4, MoPE-CLIP$_V$ preserves heads 3, 4, and 15, which correspond to the crucial region of "sitting on the horse." Conversely, DynaCLIP$_V$ prunes these heads, leading to their exclusion. This observation proves the precision of the MoPE metric in identifying and preserving vital information.

Method	Vision Enocder Width	Vision Enocder Depth	Text Encoder Width	Text Encoder Depth	Params(M) Vision + Text	MSCOCO (5K test set) TR @1	TR @5	TR @10	IR @1	IR @5	IR @10	Flickr30K (1K test set) TR @1	TR @5	TR @10	IR @1	IR @5	IR @10
Pre-trained on WIT-400M																	
CLIP-ViT-L/14 [39]	1024	24	768	12	304 + 85	76.2	92.9	96.4	58.8	82.8	89.5	96.3	99.8	100.0	84.7	97.4	99.0
CLIP-ViT-B/32 [39]	768	12	512	12	88 + 38	66.2	87.7	92.8	49.4	75.8	84.7	87.7	97.7	99.3	74.7	93.4	96.9
Pre-trained on CC3M																	
DynaCLIP$_{base}$ [17]	384	18	384	12	86 + 42	70.7	90.0	94.6	53.8	80.5	87.9	90.0	98.8	99.7	79.0	95.5	97.9
DynaCLIP$_{small}$ [17]	384	18	192	12	86 + 21	69.3	89.5	94.5	52.3	79.1	87.1	89.4	98.1	99.7	77.3	95.0	97.4
MoPE-CLIP$_{base}$	384	18	384	12	86 + 42	**71.9**	**91.4**	**95.7**	**54.9**	**81.1**	**88.6**	**92.1**	**98.8**	**99.0**	**80.6**	**95.6**	**98.1**
MoPE-CLIP$_{small}$	384	18	192	12	86 + 21	**71.2**	**90.9**	**95.0**	**53.7**	**80.5**	**87.9**	**90.8**	**98.6**	**99.6**	**79.3**	**95.5**	**97.9**
Pre-trained on YFCC15M																	
CLIP-ViT-B/32† [39]	768	12	512	12	88 + 38	34.5	63.5	75.2	24.0	50.8	63.5	57.4	84.7	90.2	40.4	69.5	79.6
SLIP-ViT-B/32† [35]	768	12	512	12	88 + 38	43.7	71.8	82.4	31.0	58.8	70.3	68.9	91.9	95.1	51.0	79.5	86.8
DeCLIP-ViT-B/32† [27]	768	12	512	12	88 + 38	47.9	75.5	84.6	33.8	62.7	71.4	73.6	93.9	97.2	55.9	83.4	90.2
UniCLIP-ViT-B/32† [25]	768	12	512	12	88 + 38	52.7	78.6	87.4	37.6	66.3	77.0	77.9	95.1	98.0	61.0	85.9	92.2
MCD-ViT-B/32† [21]	768	12	512	12	88 + 38	55.6	81.2	89.5	38.2	67.4	78.5	79.3	95.2	98.0	63.1	87.2	92.3
MoPE-CLIP$_{base}$	384	18	384	12	86 + 42	**74.3**	**92.3**	**95.9**	**56.7**	**82.0**	**89.4**	**93.3**	**99.4**	**99.9**	**82.0**	**96.4**	**98.7**

Table C7. Fine-tuned image-text retrieval results on MSCOCO and Flickr30K datasets. DynaCLIP and MoPE-CLIP are pruned during the pre-training stage and further fine-tuned on downstream datasets. † denotes the results are reported from [25, 56].

Method	Vision Enocder Width	Vision Enocder Depth	Text Encoder Width	Text Encoder Depth	Params (M) Vision + Text	Training Details Dataset	GPU	Batch size	MSCOCO TR @1	MSCOCO IR @1	Flickr30K TR @1	Flickr30K IR @1
OpenCLIP [7]	12	12	8	12	88 + 39	LAION-2B	176x A100	33792	59.4	42.4	86.2	69.8
TinyCLIP [53]	N/A	N/A	8	6	39 + 19	YFCC15M	32x A100	4096	54.9	38.9	84.4	66.7
MoPE-CLIP	6	12	4	12	43 + 19	YFCC15M	8x V100	1024	**56.2**	**39.4**	**84.5**	**67.4**

Table C8. Zero-shot image-text retrieval results of TinyCLIP and MoPE-CLIP. The original model is OpenCLIP-ViT-B/16 pre-trained on the LAION-2B dataset.

Pruning Strategy	MSCOCO TR @1	MSCOCO IR @1	Flickr30K TR @1	Flickr30K IR @1	Training cost Epochs	Training cost GPU Hours
Width-and-depth	52.8	37.3	82.8	66.7	20	320
Width-first-then-depth	54.3	38.1	84.1	67.9	40	640

Table C9. Comprasion of retrieval performance and training cost in pruning 86M+42M MoPE-CLIP$_{base}$.

Width-and-depth pruning is preferred for pre-training compression. Following Section 4.3, we extend our investigation to include both "width-and-depth pruning" and "width-first-then-depth pruning" strategies during the pre-training stage compression. We exclude the "depth-first-then-width" strategy since it falls behind the "width-first-then-depth pruning" during the fine-tuning stage. As indicated in Table C9, "width-first-then-depth pruning" shows superior performance. However, the performance gap with "width-and-depth pruning" narrows significantly compared to the fine-tuning stage. Notably, "width-first-then-depth pruning" requires an additional 20 epochs in pre-training, which can be resource-intensive for many researchers. On the other hand, "width-and-depth pruning" offers the dual benefits of one-stage pruning for faster training and the utilization of a larger set of image-text pairs, thereby yielding competitive performance. Consequently, we advocate for "width-and-depth pruning" during the pre-training stage compression, as it strikes an optimal balance between training efficiency and model capability.

C.4. Fine-tuned Evaluation for Pre-training Stage

As we discussed in Section 2, whether pruning during the pre-training stage and then fine-tuning outperforms pruning during the fine-tuning stage is an interesting question. Therefore, we further fine-tune the DynaCLIP and MoPE-CLIP on downstream datasets and compare them with other baselines. From Table C7, we observe that the finetuned MoPE-CLIP and DynaCLIP exhibit significant performance on two datasets and enlarge the gap compared to fine-tuned CLIP-ViT-B/32. This indicates that pruned models continually inherit the knowledge from the fine-tuned CLIP-ViT-L/14 during full fine-tuning. Consequently, we compare the fine-tuned MoPE-CLIP$_{base}$ with MoPE-CLIP$_{V}$ in Table 1 and find that the former showcases better TR@1. This indicates that pruning during the pre-training stage is more effective because more image-text pairs are included for learning, while the pruning during fine-tuning stage exhibits competitive results with much less training time. In addition, if we enlarge the pre-training dataset to YFCC15M, fine-tuned UniCLIP [25] and MCD [21] still fall short in comparison to MoPE-CLIP$_{base}$. This aligns with the conclusion in Section 4.2 that pruning offers a superior solution for obtaining compact VLP models.

C.5. MoPE on OpenCLIP

To assess our MoPE metric across various vision-language models, we adopted the setting used in TinyCLIP [53] and further compressed the OpenCLIP-ViT-B/16 [7], which is pre-trained on the LAION-2B dataset [43]. Specifically, we prune both the vision and language encoders to half their original widths. The MoPE metric is computed by Recall Mean on the MSCOCO validation dataset, following Section 4.2. We then pre-train the reduced model on the

YFCC15M dataset for 25 epochs, employing 16x NVIDIA V100 GPUs, and the results are presented in Table C8. We observe that our MoPE-CLIP, utilizing significantly fewer GPU resources, surpasses TinyCLIP in retrieval tasks on both MSCOCO and Flickr30K benchmarks, and narrows the performance gap with OpenCLIP. However, due to limited computational resources, we were unable to increase the batch size to 4096 as done in TinyCLIP. Therefore, we anticipate further enhancements with the availability of more GPUs. These experiments validate the effectiveness of the MoPE metric across different VLP models and also demonstrate that our MoPE-CLIP offers a straightforward yet efficient approach for pre-training stage compression.

References

[1] Reza Abbasi-Asl and Bin Yu. Structural compression of convolutional neural networks. *arXiv preprint arXiv:1705.07356*, 2017. 1

[2] Alberto Baldrati, Marco Bertini, Tiberio Uricchio, and Alberto Del Bimbo. Conditioned and composed image retrieval combining and partially fine-tuning clip-based features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4959–4968, 2022. 4

[3] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13*, pages 446–461. Springer, 2014. 1

[4] Arnav Chavan, Zhiqiang Shen, Zhuang Liu, Zechun Liu, Kwang-Ting Cheng, and Eric P Xing. Vision transformer slimming: Multi-dimension searching in continuous optimization space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4931–4941, 2022. 1

[5] Fei-Long Chen, Du-Zhen Zhang, Ming-Lun Han, Xiu-Yi Chen, Jing Shi, Shuang Xu, and Bo Xu. Vlp: A survey on vision-language pre-training. *Machine Intelligence Research*, 20(1):38–56, 2023. 1

[6] Tianlong Chen, Yu Cheng, Zhe Gan, Lu Yuan, Lei Zhang, and Zhangyang Wang. Chasing sparsity in vision transformers: An end-to-end exploration. *Advances in Neural Information Processing Systems*, 34:19974–19988, 2021. 1

[7] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2818–2829, 2023. 1, 5

[8] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613, 2014. 1

[9] Wenliang Dai, Lu Hou, Lifeng Shang, Xin Jiang, Qun Liu, and Pascale Fung. Enabling multimodal generation on clip via vision-language knowledge distillation. *arXiv preprint arXiv:2203.06386*, 2022. 1

[10] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *arXiv preprint arXiv:2305.06500*, 2023. 1

[11] 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. 1

[12] Angela Fan, Edouard Grave, and Armand Joulin. Reducing transformer depth on demand with structured dropout. *arXiv preprint arXiv:1909.11556*, 2019. 1

[13] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, pages 178–178. IEEE, 2004. 1

[14] Yunhao Gou, Zhili Liu, Kai Chen, Lanqing Hong, Hang Xu, Aoxue Li, Dit-Yan Yeung, James T Kwok, and Yu Zhang. Mixture of cluster-conditional lora experts for vision-language instruction tuning. *arXiv preprint arXiv:2312.12379*, 2023. 1

[15] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28, 2015. 1

[16] Yang He, Guoliang Kang, Xuanyi Dong, Yanwei Fu, and Yi Yang. Soft filter pruning for accelerating deep convolutional neural networks. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 2234–2240, 2018. 1

[17] Lu Hou, Zhiqi Huang, Lifeng Shang, Xin Jiang, Xiao Chen, and Qun Liu. Dynabert: Dynamic bert with adaptive width and depth. *Advances in Neural Information Processing Systems*, 33:9782–9793, 2020. 1, 3, 5

[18] 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

[19] Nan-Fei Jiang, Xu Zhao, Chao-Yang Zhao, Yong-Qi An, Ming Tang, and Jin-Qiao Wang. Pruning-aware sparse regularization for network pruning. *Machine Intelligence Research*, 20(1):109–120, 2023. 1

[20] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015. 1

[21] Bumsoo Kim, Yeonsik Jo, Jinhyung Kim, and Seunghwan Kim. Misalign, contrast then distill: Rethinking misalignments in language-image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2563–2572, 2023. 1, 5

[22] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013. 1

[23] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 1

[24] François Lagunas, Ella Charlaix, Victor Sanh, and Alexander M Rush. Block pruning for faster transformers. *arXiv preprint arXiv:2109.04838*, 2021. 1

[25] Janghyeon Lee, Jongsuk Kim, Hyounguk Shon, Bumsoo Kim, Seung Hwan Kim, Honglak Lee, and Junmo Kim. Uniclip: Unified framework for contrastive language-image pre-training. *arXiv preprint arXiv:2209.13430*, 2022. 1, 5

[26] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34:9694–9705, 2021. 1

[27] 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. 1, 5

[28] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 1, 4

[29] Zhili Liu, Kai Chen, Jianhua Han, Lanqing Hong, Hang Xu, Zhenguo Li, and James T Kwok. Task-customized masked autoencoder via mixture of cluster-conditional experts. *International Conference on Learning Representations*, 2024. 1

[30] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 1

[31] Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? *Advances in neural information processing systems*, 32, 2019. 1

[32] Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient inference. *arXiv preprint arXiv:1611.06440*, 2016. 1

[33] Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient inference. *arXiv preprint arXiv:1611.06440*, 2016. 1

[34] Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz. Importance estimation for neural network pruning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11264–11272, 2019. 1

[35] Norman Mu, Alexander Kirillov, David Wagner, and Saining Xie. Slip: Self-supervision meets language-image pre-training. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVI*, pages 529–544. Springer, 2022. 1, 5

[36] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pages 722–729. IEEE, 2008. 1

[37] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012. 1

[38] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649, 2015. 1, 3

[39] 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, 5

[40] Hassan Sajjad, Fahim Dalvi, Nadir Durrani, and Preslav Nakov. Poor man's bert: Smaller and faster transformer models. *arXiv preprint arXiv:2004.03844*, 2(2), 2020. 1

[41] Victor Sanh, Thomas Wolf, and Alexander Rush. Movement pruning: Adaptive sparsity by fine-tuning. *Advances in Neural Information Processing Systems*, 33:20378–20389, 2020. 1

[42] 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. 1

[43] 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. 5

[44] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 4

[45] Dachuan Shi, Chaofan Tao, Ying Jin, Zhendong Yang, Chun Yuan, and Jiaqi Wang. Upop: Unified and progressive pruning for compressing vision-language transformers. *arXiv preprint arXiv:2301.13741*, 2023. 1, 3

[46] Dachuan Shi, Chaofan Tao, Anyi Rao, Zhendong Yang, Chun Yuan, and Jiaqi Wang. Crossget: Cross-guided ensemble of tokens for accelerating vision-language transformers. *arXiv preprint arXiv:2305.17455*, 2023. 1

[47] Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. Mobilebert: a compact task-agnostic bert for resource-limited devices. *arXiv preprint arXiv:2004.02984*, 2020. 1

[48] Chaofan Tao, Lu Hou, Haoli Bai, Jiansheng Wei, Xin Jiang, Qun Liu, Ping Luo, and Ngai Wong. Structured pruning for efficient generative pre-trained language models. In *Findings*

of the Association for Computational Linguistics: ACL 2023, pages 10880–10895, 2023. 1

[49] Vishaal Udandarao, Ankush Gupta, and Samuel Albanie. Sus-x: Training-free name-only transfer of vision-language models. *arXiv preprint arXiv:2211.16198*, 2022. 4

[50] Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. *arXiv preprint arXiv:1905.09418*, 2019. 1

[51] Tiannan Wang, Wangchunshu Zhou, Yan Zeng, and Xinsong Zhang. Efficientvlm: Fast and accurate vision-language models via knowledge distillation and modal-adaptive pruning. Preprint arXiv:2210.07795, 2022. 1

[52] Wenxiao Wang, Shuai Zhao, Minghao Chen, Jinming Hu, Deng Cai, and Haifeng Liu. Dbp: Discrimination based block-level pruning for deep model acceleration. *arXiv preprint arXiv:1912.10178*, 2019. 1

[53] 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. 1, 5

[54] Mengzhou Xia, Zexuan Zhong, and Danqi Chen. Structured pruning learns compact and accurate models. *arXiv preprint arXiv:2204.00408*, 2022. 1

[55] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3485–3492. IEEE, 2010. 1

[56] 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. 1, 5

[57] 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. 1

[58] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021. 1

[59] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18123–18133, 2022. 1

[60] Renrui Zhang, Rongyao Fang, Wei Zhang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free clip-adapter for better vision-language modeling. *arXiv preprint arXiv:2111.03930*, 2021. 1

[61] Renrui Zhang, Xiangfei Hu, Bohao Li, Siyuan Huang, Hanqiu Deng, Yu Qiao, Peng Gao, and Hongsheng Li. Prompt, generate, then cache: Cascade of foundation models makes strong few-shot learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15211–15222, 2023. 1

[62] Yingtao Zhang, Haoli Bai, Haokun Lin, Jialin Zhao, Lu Hou, and Carlo Vittorio Cannistraci. An efficient plug-and-play post-training pruning strategy in large language models. 2023. 1

[63] Yingtao Zhang, Jialin Zhao, Wenjing Wu, Alessandro Muscoloni, and Carlo Vittorio Cannistraci. Epitopological sparse ultra-deep learning: A brain-network topological theory carves communities in sparse and percolated hyperbolic anns. 2023. 1

[64] Liu Zhili, Jianhua Han, Lanqing Hong, Hang Xu, Kai Chen, Chunjing Xu, and Zhenguo Li. Task-customized self-supervised pre-training with scalable dynamic routing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1854–1862, 2022. 1

[65] Michael Zhu and Suyog Gupta. To prune, or not to prune: exploring the efficacy of pruning for model compression. *arXiv preprint arXiv:1710.01878*, 2017. 1

[66] Xiangyang Zhu, Renrui Zhang, Bowei He, Aojun Zhou, Dong Wang, Bin Zhao, and Peng Gao. Not all features matter: Enhancing few-shot clip with adaptive prior refinement. *arXiv preprint arXiv:2304.01195*, 2023. 4