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

Haokun Lin[1,2,4], **Haoli Bai**[3], **Zhili Liu**[3], **Lu Hou**[3], **Muyi Sun**[2],
Linqi Song[4,5], **Ying Wei**[6†], and **Zhenan Sun**[2†]
[1] School of Artificial Intelligence, University of Chinese Academy of Sciences
[2] CRIPAC & MAIS, Institute of Automation, Chinese Academy of Sciences
[3]Huawei Noah's Ark Lab [4]City University of Hong Kong
[5] City University of Hong Kong Shenzhen Research Institute [6] Nanyang Technological University
haokun.lin@cripac.ia.ac.cn, ying.wei@ntu.edu.sg, znsun@nlpr.ia.ac.cn

Abstract

Vision-language pre-trained models have achieved impressive performance on various downstream tasks. However, their large model sizes hinder their utilization on platforms with limited computational resources. We find that directly using smaller pre-trained models and applying magnitude-based pruning on CLIP models leads to inflexibility and inferior performance. Recent efforts for VLP compression either adopt uni-modal compression metrics resulting in limited performance or involve costly mask-search processes with learnable masks. In this paper, we first propose the Module-wise Pruning Error (MoPE) metric, accurately assessing CLIP module importance by performance decline on cross-modal tasks. Using the MoPE metric, we introduce a unified pruning framework applicable to both pre-training and task-specific fine-tuning compression stages. For pre-training, MoPE-CLIP effectively leverages knowledge from the teacher model, significantly reducing pre-training costs while maintaining strong zero-shot capabilities. For fine-tuning, consecutive pruning from width to depth yields highly competitive task-specific models. Extensive experiments in two stages demonstrate the effectiveness of the MoPE metric, and MoPE-CLIP outperforms previous state-of-the-art VLP compression methods.

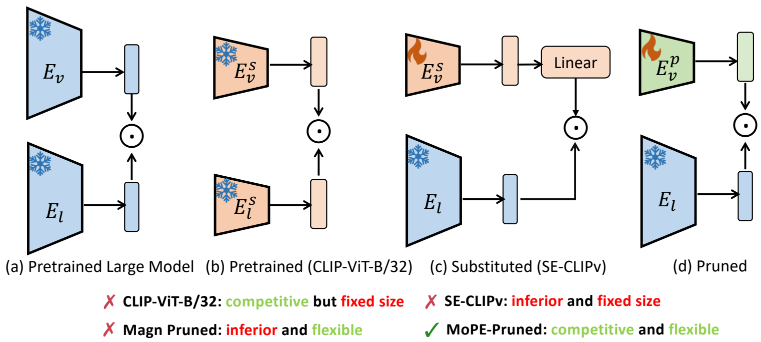
1. Introduction

Vision-Language Pre-training (VLP) models have demonstrated strong multi-modal representation learning abilities [19, 24, 30, 32]. However, their impressive performance comes at the cost of a large number of parameters, limiting their use on resource-constrained devices. Therefore, it is essential to explore compact VLP models for real-world applications [51, 57]. We identify two compression settings for different platforms. First, many edge servers lack the computational power to handle the entire pre-trained model. We define "**pre-training stage compression**" to address this, which involves compressing zero-shot VLP models and pre-training them on millions of image-text pairs to create compact, general-purpose models. Second, clients, like mobile phones, often require multiple task-specific models for various scenarios. To meet this demand, we introduce "**fine-tuning stage compression**". For example, CLIP [44] excels in cross-modal retrieval tasks, comprising image-to-text retrieval (TR) and text-to-image retrieval (IR). Given the pre-computable and offline storable nature of visual or textual representations [10, 63], our objective is to compress vision encoder for TR task and text encoder for IR task.

To reduce inference costs, smaller pre-trained models, like the various-sized ViT-based CLIP models in [44], are considered. However, individually pre-training each model is computationally expensive [8], and the limited architectural diversity may not meet various deployment needs. Consequently, we delve into more flexible solutions that leverage pruning techniques to compress VLP models. Nonetheless, the suboptimal performance of magnitude pruning on CLIP as shown in Figure 1 raises the challenge of identifying a more competitive pruning strategy.

Recent VLP pruning methods [51, 57, 59] can be broadly categorized into two categories. The simplest way involves applying uni-modal Transformer pruning methods. However, despite the effectiveness of metrics such as magnitude and loss-awareness on single-modality transformers [16, 17, 36, 39], our experiments have revealed unsatisfactory performance when directly applying them to the multimodal CLIP models. EfficientVLM [57] uses the "every other" pruning strategy during the pre-training stage, but

†Corresponding authors.



Model (Architecture)	Pruning Stage	Training Data	Params	TR@1
CLIP-ViT-L/14 (a)	N/A	MSCOCO	304M	76.2
CLIP-ViT-B/32 (b)	N/A	MSCOCO	88M	66.2
SE-CLIP$_V$ (c)	N/A	MSCOCO	88M	53.0
MagnCLIP$_V$ (d)	Fine-tuning	MSCOCO	153M	58.6
MoPE-CLIP$_V$ (d)	Fine-tuning	MSCOCO	86M	69.7
MoPE-CLIP$_{base}$ (d)	Pre-training	MSCOCO	86M	71.9

(1) Fine-tuned Evaluation On MSCOCO Dataset

Model (Architecture)	Pruning Stage	Training Data	Params	TR@1	IR@1
CLIP-ViT-L/14 (a)	N/A	WIT-400M	304M+85M	56.3	36.5
CLIP-ViT-B/32 (b)	N/A	WIT-400M	88M+38M	50.1	30.5
MagnCLIP (d)	Pre-training	CC3M	153M+42M	35.0	28.2
MoPE-CLIP$_{base}$ (d)	Pre-training	CC3M	86M+42M	52.8	37.3

(2) Zero-shot Evaluation on MSCOCO Dataset

Figure 1. Empirical comparison between (a) the original large CLIP model and three smaller models with compressed vision encoders, including (b) a **pre-trained** small CLIP Model; (c) a small model obtained by **substituting** the original vision encoder in (a) with the small vision encoder of (b); and (d) a small model with the vision encoder **pruned** from (a). We perform pruning during pre-training or fine-tuning, evaluated (1) after fine-tuning or (2) with zero shot. Note that we train the substituted encoder E_v^s in (c) and E_v^p in (d) with image-text contrastive loss $\mathcal{L}_{itc}$. TR and IR stand for image-to-text and text-to-image retrieval, respectively.

this approach, commonly used in BERT models, does not deliver optimal outcomes in our experiments. These findings underscore the inadequacy of existing metrics in assessing module impact on multi-modal tasks. On the other hand, mask-based pruning is employed to identify crucial modules. UPop [51] introduces a progressive searching process, which is unsuitable for the pre-training stage. TinyCLIP [59] suggests distillation with weight inheritance for small models, involving a time-consuming multi-stage distillation process. Additionally, TinyCLIP is pre-trained on the LAION400M dataset [47], leaving uncertainty about its effectiveness for fine-tuning stage compression with limited data. In summary, traditional pruning metrics need improvement for VLP pruning, and mask-based pruning is not efficient enough during pre-training. Thus, a unified solution for our identified two-stage compression is unexplored.

To tackle these challenges, we introduce **MoPE-CLIP**, an effective mask-free structured pruning solution for both pre-training and fine-tuning stage compression. We first propose the Module-wise Pruning Error (MoPE) metric, which quantifies a module's importance by measuring the performance drop in multi-modal tasks if that module is pruned. MoPE precisely evaluates the pruning sensitivity of heads, FFN neurons in the *width direction*, and Transformer layers in the *depth direction*. Based on the MoPE metric, we propose a unified mask-free pruning framework. In the pre-training stage, we calculate MoPE using zero-shot retrieval on the MSCOCO validation set and simultaneously prune both width and depth components. In the fine-tuning stage, MoPE is calculated by the performance decline on downstream tasks. To achieve higher pruning ratios, we prioritize pruning in the width direction before pruning in the depth direction. Moreover, we distill both *cross-modal* and *uni-modal* knowledge from the original model's aligned feature space and text/vision encoder to enhance the pruned model's capacity. Extensive evaluations demonstrate our MoPE-CLIP largely outperforms the same amount of parameters TinyCLIP [59] by 5.3% TR@1 and 4.0% IR@1 on MSCOCO retrieval tasks, while surpassing MCD [23] and ALIP [62] on 11 zero-shot classification tasks by 18.6% and 17.0%. The contributions of our work are:

- We introduce MoPE metric for precisely assessing the importance of CLIP modules in cross-modal tasks. Utilizing MoPE, we present a structured pruning framework combined with advanced distillation loss, offering a unified solution for pre-training and fine-tuning compression.
- MoPE-CLIP model exhibits SOTA performance in both training speed and accuracy across extensive experiments, surpassing existing benchmarks in various domains.

2. Preliminary Study of Downsizing CLIP

In pursuit of the objective to scale down a vision-language model such as CLIP, various alternatives come into consideration. Architecturally, one may opt to substitute an encoder with its counterpart in a smaller model, as exemplified by CLIP-ViT-B/32 in Figure 1(b), or alternatively, directly prune the encoder to any desired size. From a practical standpoint, downsizing can be executed either during pre-training prior to deployment for downstream tasks or during fine-tuning in clients. This section embarks on a preliminary examination of these alternatives, laying the groundwork for our proposed pruning strategy.

Substituting with smaller models proves unsatisfactory. We substitute the original vision encoder E_v of CLIP with a smaller one from CLIP-ViT-B/32, resulting in the downsized model SE-CLIP$_V$. We freeze the language encoder to facilitate applications like image-to-text retrieval (TR), where text features by the language encoder are oftentimes stored without modification. The modified vision encoder and the frozen language encoder are misaligned, necessitating further training. Concretely, we conduct fine-tuning of a linear layer and the vision encoder on the downstream

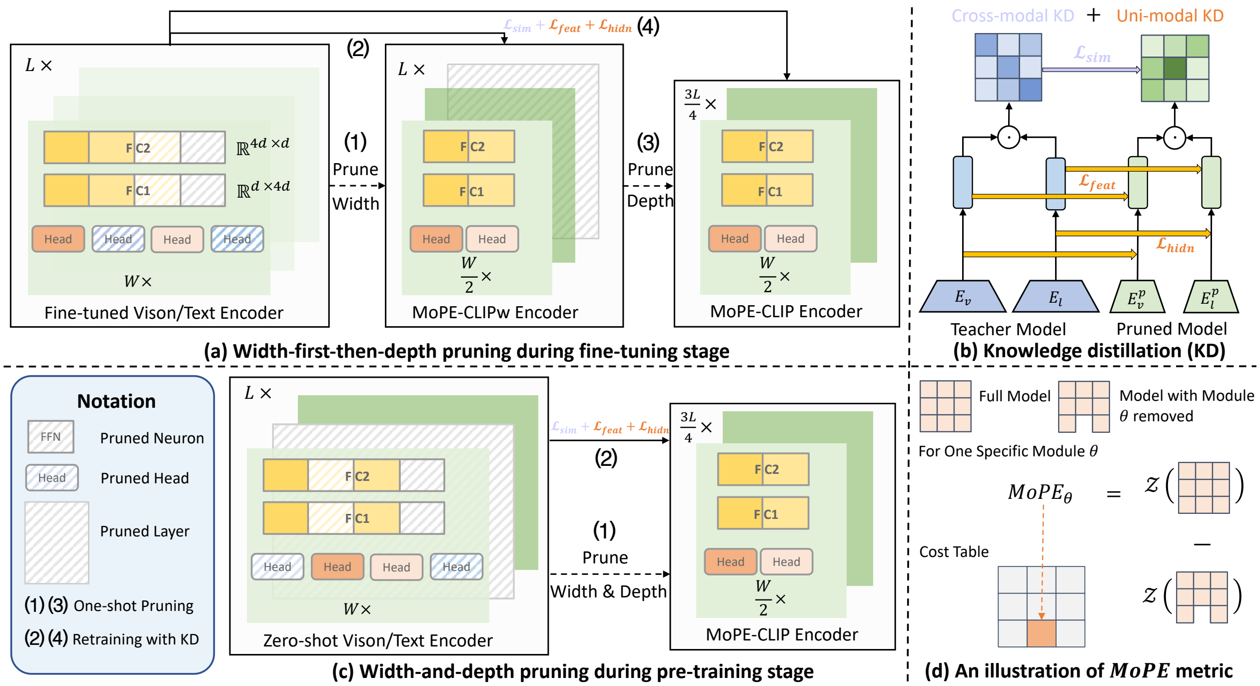


Figure 2. The overall workflow of training MoPE-CLIP. (a). During the fine-tuning stage, we apply width-first-then-depth pruning on fine-tuned CLIP vision or text encoder to obtain powerful task-specific models. (b). An illustration of our distillation process, transferring cross-modal and uni-modal knowledge. (c). During the pre-training stage, we apply consecutive pruning in the width and depth directions on zero-shot CLIP encoders. (d). An illustration of MoPE metric, measuring the performance drop of CLIP after removing the module θ.

dataset MSCOCO [33], and resort to the cross-modal contrastive loss function $\mathcal{L}_{itc}$, which is the InfoNCE loss computed between image and text features. Unfortunately, SE-CLIP$_V$ experiences a substantial performance decline compared with the original CLIP, as illustrated in Figure 1(1). This decline may be attributed to the formidable challenge of aligning two disassociated encoders originating from distinct vision-language models. This observation, coupled with the lack of flexibility in selecting a target size, dissuades us from further investigating this downsizing strategy during pre-training. Thus, we redirect our focus to the alternative choice of pruning.

Further investigation is required for successful pruning. Specifically, we implement MagnCLIP$_V$ [16], a widely adopted yet straightforward pruning strategy, which selectively prunes attention heads and FFN neurons below a specified magnitude threshold. Adjusting this threshold results in varying sizes of pruned models. We utilize the same objective function as $\mathcal{L}_{itc}$ in substitution to train the pruned model E_v^p. Despite the expected flexibility in target size, MagnCLIP$_V$ ensures only a relatively satisfactory performance, provided that at least 50% of parameters are retained, as depicted in Figure 1(1)(2). An imperative need exists for an effective pruning strategy that simultaneously meets flexibility and generalization capacity, thus forming the basis for our proposed approach.

Both pre-training and fine-tuning pruning merit consideration. An intriguing question to explore is whether a vision-language model, pre-trained before deployment, outperforms one pruned to the same size during fine-tuning. MoPE-CLIP$_{base}$ and MoPE-CLIP$_V$ represent the two versions using our proposed pruning framework which we will detail in the next. From Figure 1(1), we conclude that given the same target size, exploring both pre-training pruning and fine-tuning pruning is worthwhile. First, their application scenarios differ as discussed. Second, pruning during pre-training, when more parallel data is accessible for preserving cross-modal correlations, proves more effective, while pruning during fine-tuning which does not underperform significantly enjoys the advantage of high efficiency.

3. Method

We introduce the MoPE metric in Section 3.1 to measure module influence accurately in cross-modal tasks. In Sections 3.2 and 3.3, we present our pruning framework and knowledge distillation loss, jointly improving the two-stage compression performance.

3.1. Module-wise Pruning Error

We propose a new metric called *module-wise pruning error* (MoPE) to evaluate the importance of different modules in the CLIP model, such as Multi-Head Attention (MHA) heads, Feedforward (FFN) neurons, and entire Transformer layers. For heads and neurons, some commonly used metrics in width pruning, like magnitude [16], fail to accurately capture their impacts on multi-modal tasks, leading to suboptimal results. For Transformer layers, existing works [13, 45] mainly adopt every other strategy on depth pruning for BERT. Our experiments in Section 4.3 demonstrate that this simplistic strategy falls short when applied to CLIP models. We suppose that every other strategy cannot measure the pruning influence on another encoder and thus leads to inferior performance. These results present a new challenge in selecting a suitable metric to prune VLP models.

To overcome these issues, our proposed MoPE metric effectively assesses the module's importance with respect to multi-modal downstream tasks, offering a consistent and more accurate measure for both width and depth pruning. In particular, we regard different heads, groups of neurons, and layers as different modules. From Figure 2 (d), the importance of module θ is empirically measured by the performance decline between module θ removed CLIP model $f_{\varphi-\theta}$ and to the full CLIP f_φ counterpart as follows:

$$\mathrm{MoPE}_\theta = \mathcal{Z}\left[f_\varphi\right] - \mathcal{Z}\left[f_{\varphi-\theta}\right], \tag{1}$$

where $\mathcal{Z}$ is the zero-shot evaluation function, i.e., Recall Mean for retrieval tasks. One module θ with a higher MoPE_θ value indicates that this module is more sensitive to pruning and plays a more crucial role in cross-modal tasks. Thus, preserving such modules becomes a priority during pruning. By utilizing the MoPE_θ, we can easily create cost tables $\mathcal{C}_\theta = \sum_{i=1}^{n}\{\mathrm{MoPE}_{\theta_i}\}$. These cost tables are generated for different heads ($\mathcal{C}_{head}$), groups of neurons ($\mathcal{C}_{neuron}$), and layers ($\mathcal{C}_{layer}$). They serve as references for selecting optimal architectures, allowing us to retain critical modules while reducing the overall model size.

3.2. Unified Pruning Framework Based on MoPE

Recent VLM compression works focus either during the pre-training stage [59] or fine-tuning stage [51]. However, the comprehensive solution for these two stages is under-explored. Leveraging our MoPE metric, we introduce a unified pruning framework aimed at solving this challenge.

Fine-tuning Stage. The primary challenge lies in enhancing the performance of task-specific pruned models. To achieve high compression ratios, we explore three distinct pruning strategies in both width and depth directions. Empirical analysis reveals that the **width-first-then-depth** pruning paradigm is the most effective, as discussed in Section 4.3. Specifically, one encoder of CLIP has L layers, and each layer consists of a MHA block and a FFN block. First, we compress the fine-tuned CLIP model in the width direction, as shown in Figure 2 (a). For the MHA block with N_h heads, we independently prune $L \times N_h$ heads with their query, key, value, and output matrices. This process calculates the MoPE metric, subsequently establishing the cost table of heads $\mathcal{C}_{head}$. For the FFN block, which includes an up-projection $\mathbf{W}_1 \in \mathbb{R}^{d \times d_{ff}}$ and a down-projection layer $\mathbf{W}_2 \in \mathbb{R}^{d_{ff} \times d}$, where d is the hidden dimension and $d_{ff} = 4d$ is the number of intermediate neurons. Since it's time-cost to enumerate all d_{ff} neurons, we divide them into N groups and measure the MoPE of each group to obtain $\mathcal{C}_{neuron}$. Then the insignificant heads and groups of neurons would be pruned, and we use knowledge distillation to transfer knowledge from the fixed teacher model to the final MoPE-CLIPw. Second, we compress the MoPE-CLIPw in the depth direction. We compute the MoPE for L Transformer layers of MoPE-CLIPw and create the $\mathcal{C}_{layer}$. With the assistance of $\mathcal{C}_{layer}$, we evaluate the priority of layers precisely and prune less important ones. The final MoPE-CLIP is obtained by distilling from the fixed teacher model.

Pre-training Stage. We simultaneously compress the vision and text encoder of the large model to generate more general small models in the pre-training stage. In addition to model capacity, training cost is another crucial challenge. The width-first-then-depth strategy involves a two-stage retraining process, incurring high costs. Moreover, the knowledge acquired in each retraining process expands as more image-text pairs are introduced during pre-training. Therefore, we combine the **width-and-depth pruning** into a single stage, as depicted in Figure 2(c). In particular, we parallelly compute the MoPE metric for heads, groups of neurons, and layers of zero-shot CLIP's vision and text encoders. After creating the cost tables, the pruning process is completed directly in several seconds. Then we pretrain the pruned model on one small-scale image-text pretraining dataset (e.g., the CC3M dataset) and obtain the final MoPE-CLIP. Our experiments in Section 4.2 show that our MoPE-CLIP largely outperforms several efficient pretraining models [23, 28, 62], indicating that pruning of the large models provides a better initialization for pre-training.

Pruning Efficiency. Calculating the MoPE metric for each module takes a few seconds, and computations for all modules can be parallelized. Thus, the overall time of establishing cost tables is much less than a complete fine-tuning or pre-training process. Subsequently, we can directly prune at different ratios to obtain a series of compact models.

3.3. Distillation to MoPE-CLIP

In contrast to previous distillation methods applied to ViT or BERT [21, 64, 65, 70], we design an advanced distillation loss that effectively transfers both cross-modal and uni-modal knowledge from large CLIP (teacher model) to pruned MoPE-CLIP (student model) shown in Figure 2 (b).

Cross-modal Knowledge. The CLIP model computes the cross-modal similarity matrix for retrieval and classification

Approach	Vision Encoder			MSCOCO (5K test set)		
	Width	Depth	Parmas	TR@1	TR@5	TR@10
Teacher Model	1024	24	304M	76.2	92.9	96.4
CLIP-VIT-B/32	768	12	88M	67.5	88.0	93.4
SE-CLIP$_V$	768	12	88M	56.1	81.0	89.1
MoPE-CLIP$_V$	384	18	86M	**69.7**	**90.4**	**95.0**

Table 1. Image-to-text retrieval results of three small model architectures on MSCOCO dataset. All models are trained with distillation.

Approach	Text Encoder			MSCOCO (5K test set)		
	Width	Depth	Parmas	IR @1	IR @5	IR @10
Teacher Model	768	12	85M	58.8	82.8	89.5
CLIP-VIT-B/32	512	12	38M	49.4	75.8	84.7
SE-CLIP$_T$	512	12	38M	58.5	82.9	89.6
MoPE-CLIP$_T$	384	12	42M	**59.6**	**83.2**	**89.8**

Table 2. Text-to-image retrieval results of three small model architectures on MSCOCO dataset. All models are trained with distillation.

tasks. The teacher model exhibits a more closely aligned textual and visual embedding space, resulting in additional valuable knowledge within their similarity matrices. To enhance the cross-modal capabilities of pruned models, we minimize the soft cross-entropy loss (SCE) between student similarity matrix $\mathbf{S}$ and teacher similarity matrix $\hat{\mathbf{S}}$, i.e.,

$$\mathcal{L}_{sim} = \text{SCE}(\mathbf{S}, \hat{\mathbf{S}}). \tag{2}$$

Uni-modal Knowledge. The teacher model possesses more substantial and superior vision or text encoders. Hence, it becomes crucial to transfer the knowledge embedded within these larger encoders to the student models. Following [21], we utilize the mean squared error (MSE) loss to ensure that the student model's features $(\mathbf{F}_v, \mathbf{F}_l)$ are as similar as possible to those of the teacher model $(\hat{\mathbf{F}}_v, \hat{\mathbf{F}}_l)$:

$$\mathcal{L}_{feat} = \frac{1}{2}\text{MSE}(\mathbf{F}_v, \hat{\mathbf{F}}_v) + \frac{1}{2}\text{MSE}(\mathbf{F}_l, \hat{\mathbf{F}}_l). \tag{3}$$

Besides, we also perform intermediate-layer distillation to transfer the hidden states knowledge (i.e., the output of each Transformer layer) $\mathbf{H}_v^m (m = 1, 2, ..., M)$, $\mathbf{H}_l^k (k = 1, 2, ..., K)$ from the teacher model to the student model. The depth-pruned student would mimic the preserved intermediate layers in the teacher model. The hidden loss is

$$\mathcal{L}_{hidn}^v = \sum\nolimits_{m=1}^{M} \text{MSE}(\mathbf{H}_v^m, \hat{\mathbf{H}}_v^m), \tag{4}$$

$$\mathcal{L}_{hidn}^l = \sum\nolimits_{k=1}^{K} \text{MSE}(\mathbf{H}_l^k, \hat{\mathbf{H}}_l^k), \tag{5}$$

$$\mathcal{L}_{hidn} = \frac{1}{2}(\mathcal{L}_{hidn}^v + \mathcal{L}_{hidn}^l). \tag{6}$$

Learning Objective. Combining the cross-modal knowledge and uni-modal knowledge, we further incorporate the contrastive loss ($\mathcal{L}_{itc}$). Thus, the final training objective is

$$\mathcal{L} = \mathcal{L}_{itc} + \alpha\mathcal{L}_{sim} + \beta\mathcal{L}_{feat} + \gamma\mathcal{L}_{hidn} \tag{7}$$

By default, we do not tune and set $(\alpha, \beta, \gamma) = (1, 10^3, 1)$ to ensure a balanced magnitude of these losses.

4. Experiments

4.1. Fine-tuning Stage Compression

Experimental Settings. During the fine-tuning stage, we compress fine-tuned CLIP-ViT-L/14 (FT-L14) to create

Pruning	Vision Encoder			MSCOCO (5K test set)		
	Width	Depth	Parmas	TR @1	TR @5	TR @10
Teacher Model	1024	24	304M	76.2	92.9	96.4
MagnCLIP$_V$ [16]	512	24	153M	71.2	90.8	95.2
	384	24	115M	64.2	86.6	92.8
DynaCLIP$_V$ [18]	512	24	153M	73.9	92.0	96.0
	384	24	115M	70.3	90.0	94.9
	384	18	86M	67.6	88.7	94.1
UPop-CLIP [51]	N/A	N/A	474M‡	70.8	90.8	95.2
	N/A	N/A	280M‡	56.1	82.4	90.2
MoPE-CLIP$_V$	512	24	153M	**74.7**	**92.2**	**96.4**
	384	24	115M	**72.1**	**91.5**	**95.7**
	384	18	86M	**69.7**	**90.4**	**95.0**

Table 3. Image-to-text retrieval results of different pruning methods on the MSCOCO dataset with several pruning ratios. The Params labeled as ‡ denote the parameters of the entire model.

Pruning	Text Encoder			MSCOCO (5K test set)		
	Width	Depth	Params	IR @1	IR @5	IR @10
Teacher Model	768	12	85M	58.8	82.8	89.5
MagnCLIP$_T$ [16]	384	12	42M	59.2	82.9	89.1
	192	12	21M	56.6	81.9	89.2
DynaCLIP$_T$ [18]	384	12	42M	59.3	83.0	89.7
	192	12	21M	57.3	82.3	89.4
MoPE-CLIP$_T$	384	12	42M	**59.6**	**83.2**	**89.8**
	192	12	21M	**58.0**	**82.6**	**89.8**

Table 4. Text-to-image retrieval results of different pruning methods on the MSCOCO dataset with two pruning ratios.

task-specific models. We select cross-modal retrieval as our downstream tasks and evaluate the compressed model on the MSCOCO [33] and Flickr30K [43] datasets. Due to limited space, the results on Flickr30K are in Appendix C.2.

Implementation Details. We apply the width-first-then-depth pruning on the vision or text encoder of FT-L14 to obtain MoPE-CLIP$_V$ and MoPE-CLIP$_T$. The MoPE metric is computed by TR Mean and IR Mean, respectively. In particular, since individually processing all neurons is time-consuming, we first rewire all FFN neurons according to loss gradient like [18], then divide them into groups for acceleration. Knowledge distillation is added to enhance the performance of fine-tuned CLIP-ViT-B/32 and SE-CLIP.

Further evaluation of three small model architectures. Table 1 and Table 2 present the image-to-text retrieval and text-to-image retrieval performance of three architec-

Method	Vision Enocder		Text Encoder		Params(M)	MSCOCO (5K test set)						Flickr30K (1K test set)					
	Width	Depth	Width	Depth	Vision + Text	TR @1	TR @5	TR @10	IR @1	IR @5	IR @10	TR @1	TR @5	TR @10	IR @1	IR @5	IR @10
Pre-trained on WIT-400M																	
CLIP-ViT-L/14 [44]	1024	24	768	12	304 + 85	56.3	79.4	86.6	36.5	61.1	71.2	85.2	97.5	99.1	64.9	87.3	92.2
CLIP-ViT-B/32 [44]	768	12	512	12	88 + 38	50.1	75.0	83.5	30.5	56.0	66.9	78.8	94.9	98.2	58.8	93.6	90.2
Pre-trained on CC3M																	
EfficientVLM [57]	1024	12	768	6	152 + 42	46.6	71.7	81.3	35.9	61.6	71.8	78.8	94.9	98.2	58.8	93.6	90.2
TinyCLIP [59]	512	24	768	6	152 + 42	52.7	76.5	84.8	36.6	63.0	73.6	80.5	96.3	98.5	66.3	89.1	93.7
MoPE-CLIP$_{large}$	512	24	384	12	152 + 42	**58.0**	**81.6**	**88.5**	**40.6**	**66.0**	**75.5**	**86.5**	**97.7**	**99.0**	**69.8**	**90.6**	**95.3**
DynaCLIP$_{base}$ [18]	384	18	384	12	86 + 42	51.3	75.5	84.6	35.8	61.8	72.6	79.8	96.1	98.2	64.6	87.8	93.1
DynaCLIP$_{small}$ [18]	384	18	192	12	86 + 21	46.7	72.7	92.2	33.2	59.5	70.3	75.9	94.6	98.3	60.9	86.1	91.9
MoPE-CLIP$_{base}$	384	18	384	12	86 + 42	**52.8**	**78.1**	**86.0**	**37.3**	**63.5**	**73.6**	**82.8**	**97.1**	**98.8**	**66.7**	**88.7**	**94.1**
MoPE-CLIP$_{small}$	384	18	192	12	86 + 21	**50.3**	**75.9**	**84.8**	**35.6**	**61.7**	**72.2**	**80.2**	**95.6**	**98.5**	**64.7**	**87.8**	**93.0**
Pre-trained on YFCC15M																	
CLIP-ViT-B/32† [44]	768	12	512	12	88 + 38	20.8	43.9	55.7	13.0	31.7	42.7	34.9	63.9	75.9	23.4	47.2	58.9
SLIP-ViT-B/32† [40]	768	12	512	12	88 + 38	27.7	52.6	63.9	18.2	39.2	51.0	47.8	76.5	85.9	32.3	58.7	68.8
DeCLIP-ViT-B/32† [31]	768	12	512	12	88 + 38	28.3	53.2	64.5	18.4	39.6	51.4	51.4	80.2	88.9	34.3	60.3	70.7
UniCLIP-ViT-B/32† [28]	768	12	512	12	88 + 38	32.0	57.7	69.2	20.2	43.2	54.4	52.3	81.6	89.0	34.8	62.0	72.0
MCD-ViT-B/32† [23]	768	12	512	12	88 + 38	32.2	58.7	71.2	20.7	43.5	55.3	57.6	82.6	91.1	36.4	64.8	74.1
ALIP-ViT-B/32† [62]	768	12	512	12	88 + 38	46.8	72.4	81.8	29.3	54.4	65.4	70.5	91.9	95.7	48.9	75.1	82.9
MoPE-CLIP$_{base}$	384	18	384	12	86 + 42	**55.6**	**78.6**	**86.1**	**37.1**	**63.1**	**73.5**	**86.1**	**97.9**	**99.6**	**66.4**	**89.2**	**94.2**

Table 5. Zero-shot image-text retrieval results on MSCOCO and Flickr30K datasets. Our MoPE-CLIP$_{base}$ pre-trained on CC3M datasets outperforms the CLIP-ViT-B/32 pre-trained on WIT-400M on all the metrics. † denotes the results are reported from [23, 28, 62].

tures, respectively. With similar parameters, our MoPE-CLIP$_V$ performs best and surpasses CLIP-ViT-B/32 by 2.2% TR@1 and SE-CLIP$_V$ by 13.6% TR@1. MoPE-CLIP$_T$ at 2x compression ratio also outperforms SE-CLIP$_T$ and CLIP-ViT-B/32. These results indicate that compared to the **pretrained** small models and **substituted** encoder models, MoPE-CLIP$_V$ and MoPE-CLIP$_T$ provide better small CLIP models while maintaining flexibility. Additionally, we observe that the knowledge distillation process actually improves the TR@1 of CLIP-ViT-B/32 and SE-CLIP$_V$ in Figure 1. This demonstrates the effectiveness of the teacher's knowledge, but the architectural difference between ViT-L14 and ViT-B32 limits the final performance.

Comparison with Other Pruning Methods. We compare our models with the state-of-the-art VLP compression method UPop [51]. We also extend the uni-modal pruning methods on CLIP architecture, including the dynamic pruning method DynaBERT [18] and magnitude-based pruning [16]. Notably, distillation is applied to DynaCLIP and MagnCLIP, except for Upop whose result is from the original paper. As seen in Table 3, MoPE-CLIP$_V$ performs significantly better than other DynaCLIP$_V$ and MagnCLIP$_V$ at the same depth and width, especially the TR@1. Compared with UPop, our MoPE-CLIP$_V$ with 153M vision encoder termed an entire model of 234M parameters largely surpasses the UPop-CLIP with 474M parameters on all metrics. In addition, Table 4 shows that even at the 4x compression ratio, our MoPE-CLIP$_T$ still maintains high performance on the text-to-image retrieval task, with only a 0.8% drop in IR@1 compared to the teacher model. We report performance under more pruning ratios and compare Upop with KD in Appendix C.1. We also analyze the difference of preserved heads between MoPE-CLIP$_V$ and DynaCLIP$_V$ in Appendix C.3, which further demonstrates the accurate assessment of our MoPE metrics.

4.2. Pre-training Stage Compression

Experimental Setting During the pre-training stage, we compress the zero-shot CLIP-ViT-L/14 (ZS-14) model [44] to obtain compact general models. Subsequently, we pre-train our MoPE-CLIP and various baselines on a small-scale pre-training dataset, CC3M [50]. To further assess the capabilities of our MoPE-CLIP model, we scale up training using larger datasets, including CC12M [4] and YFCC15M [31]. In addition, we evaluate our pruning method on Open-CLIP ViT-B/16 and report results in Appendix C.5.

Implementation Details We simultaneously prune both vision and text encoders of ZS-L14. For the vision encoder, we adopt width-and-depth pruning and compress the encoder to 86M parameters, which is similar to CLIP-ViT-B/32. For the text encoder, we compress it in the width direction at two pruning ratios, resulting in MoPE-CLIP$_{base}$ and MoPE-CLIP$_{small}$. We also prune both the vision and text encoders to half-width, producing MoPE-CLIP$_{large}$. The module's importance is evaluated on the MSCOCO validation dataset and the Recall Mean serves as the MoPE metric. More details are left in Appendix B.

Zero-shot Image-text Retrieval. Table 5 shows the zero-shot retrieval results on MSCOCO and Flickr30K datasets. MoPE-CLIP$_{base}$ consistently surpasses the CLIP-ViT-B32 in all Recall metrics. MoPE-CLIP$_{small}$ maintains competitive results and outperforms the DynaCLIP$_{small}$ with a clear margin. In addition, when compared with previous efficient pre-training methods, MoPE-CLIP$_{base}$ pre-trained on CC3M achieves 52.8% TR@1 and 37.3% IR@1 on the MSCOCO dataset, which is 6.0% and 8.0% higher than ALIP [62] pre-trained on YFCC15M. The improvement is mainly attributed to the pruned large model providing a better initialization for pre-training vision-language models.

Method	Pre-training dataset	Training epochs	CIFAR10	CIFAR100	Caltech101	Flowers	Pets	DTD	Cars	Aircraft	SUN397	Food101	ImageNet	**Average**
CLIP-ViT-B/32† [44]	YFCC15M	50	62.3	33.6	55.4	6.3	19.4	16.9	2.1	1.4	40.2	33.7	31.3	27.5
SLIP-ViT-B/32† [40]	YFCC15M	50	72.2	45.3	65.9	6.8	28.3	21.8	2.9	1.9	45.1	44.7	38.3	33.9
DeCLIP-ViT-B/32† [31]	YFCC15M	50	72.1	39.7	70.1	7.1	30.2	24.2	3.9	2.5	41.6	46.9	39.2	34.3
UniCLIP-ViT-B/32† [28]	YFCC15M	50	78.6	47.2	73.0	8.1	32.5	23.3	3.4	2.8	50.4	48.7	41.2	37.2
MCD-ViT-B/32† [23]	YFCC15M	32	80.3	49.6	73.2	7.9	40.0	30.5	3.4	3.0	55.3	54.0	44.7	40.2
ALIP-ViT-B/32† [62]	YFCC15M	32	83.8	51.9	74.1	54.8	30.7	23.2	5.4	2.7	47.8	45.4	40.3	41.8
MoPE-CLIP$_{base}$	YFCC15M	20	**91.5**	**68.1**	**85.5**	**66.8**	69.3	**46.6**	16.6	6.0	61.2	**74.6**	**60.7**	58.8
MoPE-CLIP$_{base}$	CC3M	20	86.8	61.7	79.0	30.1	42.0	38.5	5.6	1.7	57.1	38.6	44.5	44.2
MoPE-CLIP$_{base}$	CC12M	20	91.2	67.3	85.0	45.0	**80.0**	41.1	**47.7**	**7.2**	**62.2**	70.6	**60.7**	**59.8**

Table 6. Top-1 accuracy(%) of zero-shot image classification on 11 downstream datasets. Our MoPE-CLIP$_{base}$ largely surpasses other state-of-the-art efficient pre-training methods using fewer training epochs. † denotes the results are reported from [23, 28, 62].

Zero-shot Classification. We adopt the Recall Mean on the MSCOCO validation dataset as the MoPE metric, which reflects the module influence of multi-modal tasks. To demonstrate the robustness of Recall Mean on uni-modal tasks, we further compare our MoPE-CLIP$_{base}$ with other efficient pre-training methods on zero-shot image classification tasks. SLIP [40], DeCLIP [31], and UniCLIP [28] incorporate fine-grained supervision to reduce the data requirement. ALIP [62] and MCD [23] propose new frameworks to reduce the noise and misalignments in image-text pairs. We utilize the same prompt templates following CLIP [16]. Table 6 presents the results on 11 widely used benchmarks. Our MoPE-CLIP$_{base}$ pre-trained on the YFCC15M dataset significantly surpasses previous methods and creates new state-of-the-art results, indicating the effectiveness of MoPE-CLIP towards classification tasks.

Comparison with VLP Compression Methods. We employ EfficientVLM [57] and TinyCLIP [59] to compress the zero-shot CLIP-ViT-L/14 model, then pre-train pruned models on the CC3M dataset using distillation loss in their papers. The zero-shot retrieval results, as presented in Table 5, unequivocally illustrate that our MoPE-CLIP$_{large}$ performs the best. Notably, even our MoPE-CLIP$_{base}$ with a reduction to 66M parameters still surpasses the TinyCLIP and EfficientVLM. We further compare the training process of these models on 8x Nvidia V100 GPUs in Figure 3. Our models achieve competitive results in less training time, highlighting the significant contribution of our MoPE metric in preserving crucial modules. In contrast to TinyCLIP, which focuses on cross-modal affinity, and EfficientVLM, which emphasizes uni-modal knowledge transfer, our approach combines cross- and uni-modal distillation, proving more effective in enhancing pruned model capacity.

Inference Speedup. We measure the latency using PyTorch inference mode with the batch size of 64 in Table 7, where MoPE-CLIP shows a significant speedup.

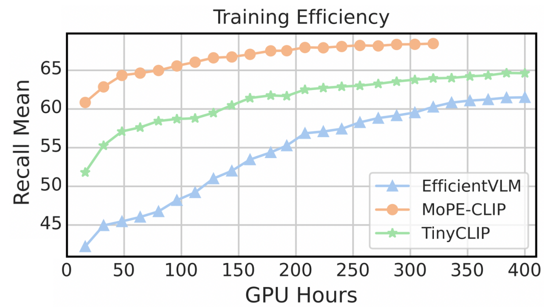


Figure 3. Comparsion of training efficiency. Models are pruned to half of the CLIP-ViT-L/14 and trained on the CC3M dataset. MoPE-CLIP performs the best with fewer training epochs.

Models	CLIP-ViT-L/14	MoPE$_{large}$	MoPE$_{base}$	MoPE$_{small}$
Params (M)	390	194 $\downarrow$ 50%	128 $\downarrow$ 67%	107 $\downarrow$ 73%
Latency (ms)	141.96	79.00 $\downarrow$ 44%	58.73 $\downarrow$ 59%	49.48 $\downarrow$ 65%

Table 7. Nvidia V100 GPU latency (ms) on MSCOCO test sets.

4.3. Ablation Study

Effects of MoPE Metric. To further demonstrate the effectiveness of our MoPE metric, we apply four commonly used layer pruning strategies in BERT [13, 45] to the 0.375-width MoPE-CLIP$_V$ models, including (i) removal of either the bottom or top layers, (ii) the "Every Other" strategy, and (iii) a Gradient-based approach that gauges layer importance by analyzing gradients for all heads and neurons within a layer. From Table 8, our MoPE metric outperforms other strategies with a clear margin. Notably, the Every Other strategy falls behind when pruning three layers. We assume that simply reducing every other layer in Transformer encoder may not influence the model capacity of uni-model tasks as proven in [13]. However, the unavailability of the other encoder results in a performance drop in cross-modal tasks. These findings indicate the importance of selecting an appropriate strategy for layer reduction in CLIP and we provide a straightforward yet valuable metric.

Setting	Method	COCO test set		
		TR@1	TR@5	TR@10
Prune 3 layers for 0.375width MWPE-CLIP$_V$	Top Layers	70.1	90.2	95.4
	Bottom Layers	70.8	90.4	95.1
	Every Other	69.2	90.0	94.9
	Loss Gradient	70.4	90.6	94.9
	MWPE metric	**72.2**	**91.2**	**95.5**
Prune 6 layers for 0.375width MWPE-CLIP$_V$	Top Layers	57.6	81.7	88.6
	Bottom Layers	63.9	88.0	93.5
	Every Other	66.6	88.9	93.6
	Loss Gradient	66.3	87.8	94.0
	MWPE metric	**69.7**	**90.4**	**95.0**

Table 8. Ablation study of Layer Selection strategies.

MoPE-CLIP$_V$	MSCOCO			
	TR@1	TR@5	TR@10	TRMean
Depth-first-then-width	64.0	87.0	92.4	81.1
Width-and-depth	61.4	84.8	90.9	79.0
Width-first-then-depth	**69.7**	**90.4**	**95.0**	**85.0**

Table 9. Ablation study in pruning 86M MoPE-CLIP$_V$.

Training Loss	MSCOCO			
	TR@1	TR@5	TR@10	TRMean
MoPE-CLIP$_V$	**69.7**	**90.4**	**95.0**	**85.0**
w/o $\mathcal{L}_{sim}$	68.8	89.8	94.7	84.4
w/o $\mathcal{L}_{feat}$	69.4	89.4	94.7	84.5
w/o $\mathcal{L}_{hidn}$	67.7	89.1	94.0	83.6
w/o *Distillation*	60.8	84.9	91.5	79.0

Table 10. Ablation study of knowledge distillation.

Effects of Pruning Framework. During the fine-tuning stage, we further explore the other two strategies, including pruning in a "depth-first" manner followed by "width pruning," as well as simultaneous "width-and-depth" pruning. From Table 9, the "width-first-then-depth" strategy yields the best performance. This may be attributed to the sequential computation of hidden states across different layers, making it challenging to accurately evaluate the importance of heads or neurons in layer-reduced models, as discussed in [18]. Furthermore, the fine-tuning dataset may not be sufficiently large to fully restore the model's capacity. Therefore, during the fine-tuning stage, the "width-first-then-depth" strategy stands out as the optimal choice for creating more competitive smaller models. In contrast, during the pre-training stage, adopting the "width-and-depth" pruning strategy is more convenient and efficient, with performance recovery facilitated by a much larger dataset.

Effects of Knowledge Distillation. We ablate our distillation objectives designed in Section 3.3, by investigating the learning process on MoPE-CLIP$_V$. From Table 10, we observe that all distilled models outperform models without distillation by a clear margin, demonstrating the importance of both cross-modal and uni-modal knowledge. Importantly, the TR@1 of the "w/o $\mathcal{L}_{hidn}$" model drops significantly from 69.7% to 67.7%, which indicates the intermediate layer knowledge in the teacher model is crucial for retraining the MoPE-CLIP. However, the discrepancy in patch numbers between the ViT-B/32 and ViT-L/14 leads to the failure of hidden distillation applied to CLIP-ViT-B32 and SE-CLIP$_V$. Consequently, the effectiveness of knowledge distillation is largely diminished for these architectures.

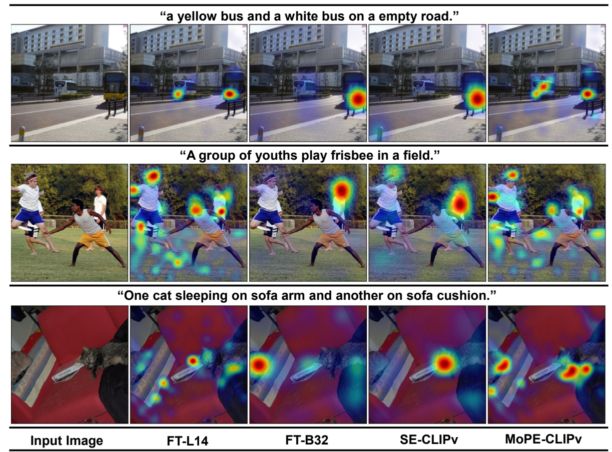


Figure 4. Grad-CAM visualization on the self-attention maps in the last layer of vision encoder for different models.

Visualization. To better understand the architecture influence on the retrieval task, we utilize Grad-CAM[49] to visualize the critical image regions corresponding to the caption. The Grad-CAM is computed on the average self-attention maps in the vision encoder's last layer, where gradients are acquired by the contrastive loss $\mathcal{L}_{itc}$. From Figure 4, the visualizations from CLIP-ViT-L/14 (FT-L14) are more precise than CLIP-ViT-B/32 (FT-B32). The FT-L14 model has a smaller patch size of 14 and thus locates more detailed regions, like the "frisbee" in the middle example. Both FT-B32 and SE-CLIP$_V$ miss "a white bus" in the top case while losing "one cat" in the bottom case. MoPE-CLIP$_V$ captures these important objects correctly. This indicates that MoPE-CLIP$_V$ provides fruitful information for the retrieval task.

5. Conclusion

This paper investigates diverse methods to downsize VLP models and focuses on exploring better pruning solutions. We propose the Module-wise pruning error (MoPE) metric, which accurately measures the CLIP module's importance. On top of the MoPE metric, we introduce a unified framework and an advanced distillation loss for structured pruning during the pre-training and fine-tuning stages. Extensive experiments demonstrated that our MoPE-CLIP achieves surprising success across various downstream tasks.

Acknowledgements: This work is supported in part by the National Natural Science Foundation of China under Grant 62371411 and 62306309, the Research Grants Council of the Hong Kong SAR under Grant GRF 11217823.

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] 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. 6

[5] 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

[6] 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

[7] 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

[8] 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

[9] 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

[10] 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

[11] 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

[12] 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

[13] Angela Fan, Edouard Grave, and Armand Joulin. Reducing transformer depth on demand with structured dropout. *arXiv preprint arXiv:1909.11556*, 2019. 4, 7, 1

[14] 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

[15] 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

[16] 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, 3, 4, 5, 6, 7

[17] 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

[18] 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. 5, 6, 8, 1, 3

[19] 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

[20] 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

[21] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351*, 2019. 4, 5

[22] 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

[23] 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. 2, 4, 6, 7, 1, 5

[24] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *International Conference on Machine Learning*, pages 5583–5594. PMLR, 2021. 1

[25] 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

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

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

[28] 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. 4, 6, 7, 1, 5

[29] 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

[30] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 1

[31] 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. 6, 7, 1, 5

[32] 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. 1

[33] 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. 3, 5, 1, 4

[34] 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

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

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

[37] 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

[38] 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

[39] 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

[40] 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. 6, 7, 1, 5

[41] 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

[42] 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

[43] 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. 5, 1, 3

[44] 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, 6, 7, 5

[45] 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. 4, 7, 1

[46] 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

[47] 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, 1

[48] 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

[49] 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. 8, 4

[50] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, 2018. 6

[51] 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, 2, 4, 5, 6, 3

[52] 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

[53] 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

[54] 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

[55] 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

[56] 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

[57] 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, 6, 7

[58] 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

[59] 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, 2, 4, 6, 7, 5

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

[61] 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

[62] 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, 4, 6, 7, 1, 5

[63] 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

[64] Hao Yu and Jianxin Wu. A unified pruning framework for vision transformers. *Science China Information Sciences*, 66 (7):1–2, 2023. 4

[65] Shixing Yu, Tianlong Chen, Jiayi Shen, Huan Yuan, Jianchao Tan, Sen Yang, Ji Liu, and Zhangyang Wang. Unified visual transformer compression. *arXiv preprint arXiv:2203.08243*, 2022. 4

[66] 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

[67] 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

[68] 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

[69] 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

[70] Wei Zhang, Lu Hou, Yichun Yin, Lifeng Shang, Xiao Chen, Xin Jiang, and Qun Liu. Ternarybert: Distillation-aware ultra-low bit bert. *arXiv preprint arXiv:2009.12812*, 2020. 4

[71] 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

[72] 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

[73] 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

[74] 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

[75] 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