

# EfficientVLM: Fast and Accurate Vision-Language Models via Knowledge Distillation and Modal-adaptive Pruning

Tiannan Wang\*

Beihang University  
tiannanwang@buaa.edu.cn

Yan Zeng

Bytedance AI Lab  
zengyan.yanne@bytedance.com

Wangchunshu Zhou\*†

ETH Zurich  
wangchunshu.zhou@inf.ethz.ch

Xinsong Zhang

Bytedance AI Lab  
zhangxinsong.0320@bytedance.com

## Abstract

Pre-trained vision-language models (VLMs) have achieved impressive results in a range of vision-language tasks. However, popular VLMs usually consist of hundreds of millions of parameters which brings challenges for fine-tuning and deployment in real-world applications due to space, memory, and latency constraints. In this work, we introduce a *distilling then pruning* framework to compress large vision-language models into smaller, faster, and more accurate ones. We first shrink the size of a pre-trained large VLM and apply knowledge distillation in the vision-language pre-training stage to obtain a task-agnostic compact VLM. Then we propose a modal-adaptive pruning algorithm to automatically infer the importance of vision and language modalities for different downstream tasks and adaptively remove redundant structures and neurons in different encoders with controllable target sparsity.

We apply our framework to train EfficientVLM, a fast and accurate vision-language model consisting of 6 vision layers, 3 text layers, and 3 cross-modal fusion layers, accounting for only 93 million parameters in total, which is 44.3% of the teacher model. EfficientVLM retains 98.4% performance of the teacher model and accelerates its inference speed by 2.2 $\times$ . EfficientVLM achieves a large absolute improvement over previous SoTA efficient VLMs of similar sizes by a large margin on various vision-language tasks, including VQAv2 (+4.9%), NLVR2 (+5.6%), ITR (R@1 on TR +17.2%, on IR + 15.6% ) and COCO caption generation (CIDEr +6.5), demonstrating a large potential on training lightweight VLMs.<sup>1</sup>

\*Equal contribution, work done during internship at Bytedance AI Lab

† Correspondence to: wangchunshu.zhou@inf.ethz.ch

<sup>1</sup>Our code and pretrained checkpoints are available at <https://github.com/swaggy-TN/EfficientVLM>.

## 1 Introduction

Inspired by the success of large pre-trained language models (Devlin et al., 2019; Radford et al., 2018) in the field of natural language processing (NLP), recent studies (Su et al., 2019; Li et al., 2020a; Radford et al., 2021a; Kim et al., 2021; Li et al., 2021b) in vision-language pretraining (VLP) have advanced the state-of-the-art on various vision-language tasks such as image captioning, visual question answering, and image-text retrieval.

However, in both NLP and vision-language domains, large Transformer-based pre-trained models often consist of hundreds of millions, if not billions, of parameters, bringing various practical challenges for deployment. As summarized in Schwartz et al. (2020a) and Xu et al. (2021d), large pre-trained models require large amounts of space (in terms of GPU memory and disk storage) and heavy computing for fine-tuning and inference, which is both costly and may lead to negative environmental impact. Furthermore, large models inevitably lead to low latency, which poses a challenge for the production environment.

Recent literature revealed that BERT (Devlin et al., 2019), a popular Transformer-based pre-trained language model, can be effectively compressed and accelerated via knowledge distillation (Sanh et al., 2019; Jiao et al., 2019; Xu et al., 2020; Wang et al., 2020b). However, only a few prior works investigated building efficient VLMs. For instance, Wang et al. (2020a) introduced MiniVLM which combines a lighter object detector with MiniLM (Wang et al., 2020b). Fang et al. (2021) further proposed DistilVLM, which uses knowledge distillation to pre-train a compact VLM with the guidance of a large pre-trained VLM. However, their approach is limited to object-feature-based VLMs. As such, the vision feature extractor cannot be distilled together with the Transformer model in an end-to-end manner,

which limits the potential of knowledge distillation. As a result, existing compact VLMs are generally falling short compared to regular-size VLMs.

In this work, we investigate strategies for VLM compression and introduce a *distilling then pruning* framework for compressing fully Transformer-based VLMs. Specifically, in the first stage, we use knowledge distillation for task-agnostic compression of a pre-trained VLM by aligning the logits, attention distribution, and hidden representations between the student model and the teacher model. This results in a **task-agnostic** compact VLM that achieves competitive results on many downstream vision-language tasks by simply fine-tuning. The general distillation stage reduces the size of all modules (i.e., vision encoder, text encoder, cross-modal encoder) equally so that the compressed model can be versatile to different downstream tasks. However, our preliminary study, which is described in detail in section 3.3, shows that not all modules are created equal in a VLM and their importance drastically varies on different downstream vision-language tasks requiring different levels of understanding on either vision and text modalities. This indicates that compressing a VLM requires modal- and **task-specific** designs. Therefore, in the second stage, we propose to prune the compact VLM when fine-tuning on different downstream tasks to flexibly adjust the model size/latency according to modal importance. Concretely, we propose a modal-adaptive pruning strategy that regularizes the model with a differentiable approximation to the  $L_0$ -norm regularization (Louizos et al., 2017) to automatically infer the importance of vision and language modalities with controllable target sparsity. In this way, our method can adaptively prune different modules in the VLM in the fine-tuning stage according to the relative importance of vision-language modalities on different downstream tasks.

We apply our framework to compress X-VLM (Zeng et al., 2021), a recent Transformer-based VLM and train EfficientVLM, a fast and accurate vision-language model. EfficientVLM consists of 6 vision layers, 3 text layers, and 3 cross-modal fusion layers, accounting for only 93 million parameters in total, which is 44.3% of the X-VLM model. EfficientVLM recovers 98.4% performance of X-VLM and accelerates its inference speed by  $2.2\times$ . Experimental results show that despite being trained with fewer image-text pairs, EfficientVLM achieves a large absolute improvement over Distil-

VLM, the previous best-performing efficient VLM with similar size and inference speed, on various vision-language tasks, including VQAv2 (Goyal et al., 2017) (+6.7%), NLVR2 (Suhr et al., 2018) (+7.8%), ITR-COCO (Lin et al., 2014) (R@1 on TR +19.9%, R@1 on IR + 15.6%) and COCO caption generation (Chen et al., 2015) (CIDEr +6.5), demonstrating a large potential on training lightweight VLMs.

To the best of our knowledge, our work is the first attempt to (1) compress a fully Transformer-based vision-language model, and (2) combine knowledge distillation with (modal-adaptive) pruning for vision-language model compression.

## 2 Related Work

**Vision-Language Pre-training** The existing work on vision language pre-training typically falls into two categories. Most methods rely on object detection (Tan and Bansal, 2019; Lu et al., 2019; Li et al., 2019; Su et al., 2019; Li et al., 2020a; Chen et al., 2020; Li et al., 2020b; Gan et al., 2020; Li et al., 2021b; Xu et al., 2021c; Liu et al., 2021; Li et al., 2022; Zhou et al., 2022b), where an image is represented by dozens of object-centric features. However, the object detection process requires high-resolution images as model input and is very time-consuming. Moreover, most works under this category utilize pre-trained object detectors (Ren et al., 2015; Anderson et al., 2018), and do not optimize the model in an end-to-end manner, yielding sub-optimal performance. Therefore, recent works turn to encoding images by convolutional network (Jiang et al., 2020; Huang et al., 2020, 2021; Wang et al., 2022) or vision transformer (Kim et al., 2021; Li et al., 2021a), largely improving the inference speed. Nevertheless, some recent work (Zhang et al., 2021; Zeng et al., 2021, 2022) shows that understanding fine-grained vision language alignments (e.g. object-level) is critical for some downstream tasks such as visual reasoning and visual grounding.

**Pre-trained Model Compression** Prior work has shown that BERT (Devlin et al., 2019), a popular encoder-only pre-trained Transformer (Vaswani et al., 2017), can be effectively compressed and accelerated. As summarized in Xu et al. (2021d) and Xu et al. (2021a), popular BERT compression techniques include knowledge distillation (Hinton et al., 2015; Sanh et al., 2019; Sun et al., 2019; Jiao et al., 2019; Wang et al., 2020b; Zhou et al., 2022a; Xu

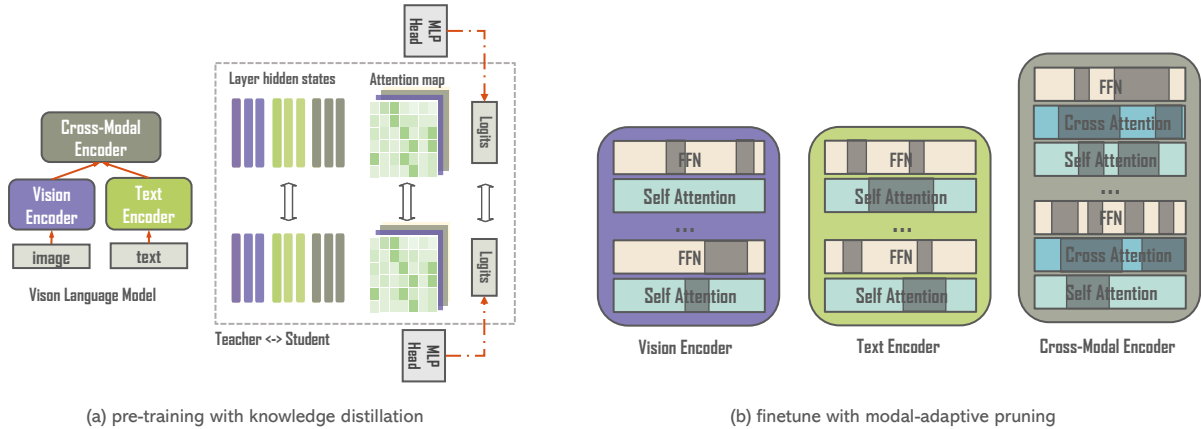


Figure 1: The *distilling then pruning* framework for training EfficientVLM. In the pre-training stage, we apply knowledge distillation with a pre-trained X-VLM model as the teacher. During fine-tuning, we use a modal-adaptive pruning method to adaptively prune encoders of different modalities.

et al., 2021b) which trains a compact student network to mimic the behavior of the original teacher model, pruning (LeCun et al., 1989; Michel et al., 2019; Gordon et al., 2020; Sanh et al., 2020; Lagunas et al., 2021; Wang et al., 2019; Xia et al., 2022) which prunes redundant neurons or structures in the original model, module replacing (Xu et al., 2020) which train compact successor sub-modules to replace that in the original model, and quantization (Shen et al., 2020; Zafrir et al., 2019) that compresses a neural network by reducing the number of bits used to represent its parameters. On the other hand, a number of work also investigated efficient inference with BERT-like models with early exit (Teerapittayanon et al., 2016; Xin et al., 2020; Liu et al., 2020; Schwartz et al., 2020b; Zhou et al., 2020) or adaptive computation time (Graves, 2016; Eyzaguirre et al., 2021; Zhou et al., 2023).

In contrast, only a few prior works investigated methods to compress a pre-trained vision-language model. Fang et al. (2021) explored distilling a pre-trained vision-language model into a more compact student model and proposed a teacher adaptation method that aligns object feature proposal. However, their approach is limited to the use of object detection based vision-language model, which makes end-to-end distillation infeasible and results in unsatisfactory performance compared to the recent state-of-the-art. Wang et al. (2021) explored distilling a vision-language model with a cross-modal fusion module to a dual-encoder model for efficient retrieval. Moreover, Gan et al. (2021) explored the lottery ticket hypothesis (Frankle and Carbin, 2018) in vision-language models and find that sparse winning tickets exist in pre-trained

VLMs. However, the process of finding and re-training winning tickets is less efficient compared to other compression methods.

### 3 EfficientVLM

In this section, we present EfficientVLM, a fast and accurate vision-language model trained with our *distilling then pruning* framework. We choose X-VLM (Zeng et al., 2021), one of the state-of-the-art vision-language models, as the teacher model.<sup>2</sup>

#### 3.1 Model Overview

EfficientVLM is a compressed version of X-VLM, a fully Transformer-based VLM. X-VLM has the same architecture as ALBEF (Li et al., 2021a), which consists of an image encoder, a text encoder, and a cross-modal encoder. The image encoder contains 12 transformer layers, while the text encoder and the cross-modal encoder each consist of 6 transformer layers. The cross-modal encoder fuses the vision features with the text features by cross-attention at each layer. EfficientVLM shrinks the size of X-VLM by half, thus consisting of 6 vision layers, 3 text layers, and 3 cross-modal layers, accounting for only 92 million parameters in total, which is 43.6% of the X-VLM model.

The teacher model is optimized by: 1) aligning the texts and visual concepts, where the alignments are in multi-granularity using a contrastive loss  $\mathcal{L}_{ITC}$ , a matching loss  $\mathcal{L}_{ITM}$ , and a masked language modeling loss  $\mathcal{L}_{MLM}$ ; 2) in the meantime

<sup>2</sup>In practice, our proposed method suits any VLMs that are equipped with modal-specific modules such as VLMo (Bao et al., 2022) or ALBEF (Li et al., 2021a).

locating visual concepts in the image given the corresponding texts by bounding box prediction loss  $\mathcal{L}_{\text{BBOX}}$ . Overall, the vision language pre-training loss is:

$$\mathcal{L}_{\text{VLP}} = \mathcal{L}_{\text{ITC}} + \mathcal{L}_{\text{ITM}} + \mathcal{L}_{\text{MLM}} + \mathcal{L}_{\text{BBOX}} \quad (1)$$

### 3.2 Pre-training with Knowledge Distillation

We initialize EfficientVLM with a pre-trained X-VLM and shrink its size by half by only retaining the even-numbered layers. Then we pre-train EfficientVLM on image-text pairs with both the original vision-language pre-training objectives of X-VLM and knowledge distillation objective with the pre-trained X-VLM as the teacher model. The knowledge distillation objective consists of attention distillation, hidden states distillation, and logits distillation.

**Attention Distillation** Prior work (Jiao et al., 2019) on BERT distillation has shown the effectiveness of transferring the latent knowledge in self-attention matrices:

$$\mathbf{A} = \text{softmax}(\mathbf{Q} \cdot \mathbf{K} / \sqrt{d_k}). \quad (2)$$

where  $\mathbf{Q}$  and  $\mathbf{K}$  denote the query and key matrix in the attention layer of a transformer block.  $d_k$  is the dimension of the key matrix as a scaling factor. We formulate attention distillation loss by minimizing the mean square error between the self-attention matrices of the teacher and the student:

$$\mathcal{L}_{\text{attn}} = \frac{1}{h} \sum_{j=1}^L \sum_{i=1}^h \text{MSE}(\mathbf{A}_{i,j}^S, \mathbf{A}_{i,2j}^T) \quad (3)$$

where  $L$  denotes the number of layers in each encoder of the student,  $h$  is the number of attention heads,  $\mathbf{A}_i$  refers to the normalized attention matrix corresponding to the  $i$ -th head in  $j$ -th layer of the student and in  $2j$ -th layer of the teacher. The attention matrix is in the shape of  $\mathbf{A} \in \mathbb{R}^{l \times p}$ .  $l$  and  $p$  are the length of query and key, respectively<sup>3</sup>.

**Hidden States Distillation** Following Transformer distillation in TinyBERT (Jiao et al., 2019), we also adopt the hidden states distillation to better utilize the information from the teacher model. The loss function is defined as follows:

$$\mathcal{L}_{\text{hid}} = \sum_{i=1}^L \text{MSE}(\mathbf{H}_i^S, \mathbf{H}_{2i}^T), \quad (4)$$

$\mathbf{H}^S \in \mathbb{R}^{l \times d'}$  and  $\mathbf{H}^T \in \mathbb{R}^{l \times d}$  refer to the hidden states of student and teacher networks in the corresponding layer.

<sup>3</sup>In the cross-attention module of cross-modal encoder,  $p$  represents the length of patch sequence of vision encoder otherwise  $l$  and  $p$  are equal

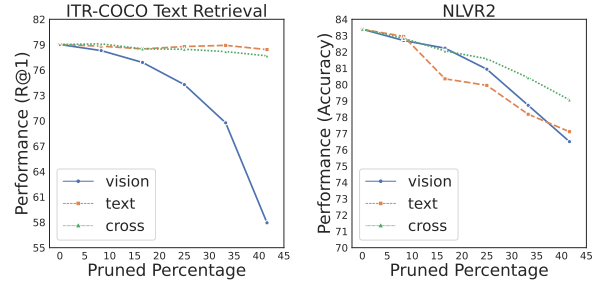


Figure 2: Empirical study of modal-encoders importance on NLVR2 and ITR-COCO tasks.

**Logits Distillation** In addition to imitating the behaviors of intermediate layers, we also use knowledge distillation to fit the predictions of teacher model as in (Hinton et al., 2015). We adopt KL divergence as the optimization objective:

**Pre-training** We formulate the final loss by combing the original vision-language pre-training loss with general distillation loss.

$$\begin{aligned} \mathcal{L}_{\text{KD}} &= \alpha \mathcal{L}_{\text{attn}} + \beta \mathcal{L}_{\text{hid}} + \gamma \mathcal{L}_{\text{logits}} \\ \mathcal{L}_{\text{pretrain}} &= \lambda \mathcal{L}_{\text{VLP}} + (1 - \lambda) \mathcal{L}_{\text{KD}} \end{aligned}$$

where  $\alpha$ ,  $\beta$ ,  $\gamma$  and  $\lambda$  are the weights of the loss terms. We only adjust the weights to scale the losses to similar values so that the optimization process can perform more robustly.

### 3.3 Fine-tuning with Pruning

To flexibly adjust the efficiency-performance trade-off of EfficientVLM on different downstream tasks according to varying resource constraints, we propose a modal-adaptive pruning method to further compress EfficientVLM to a desired size in the fine-tuning stage.

**Are All Modalities Created Equal in VLMs?** Unlike prior work (Lagunas et al., 2021) on BERT pruning where there is only one Transformer encoder, pruning VLMs are more challenging because the importance of vision and language clues may not be equally important (Cao et al., 2020). This is also verified by our preliminary experiments where we prune 40% attention heads in each encoder and find that the performance drops drastically, which is contrary to prior findings on pruning BERT (Michel et al., 2019).

To this end, we conduct an empirical study to investigate whether encoders for vision/language modalities have similar importance across different vision-language tasks. We prune each encoder

in a fine-tuned teacher model at one time while leaving other encoders untouched. From Figure 2, we observe that: (1) the encoders of different modalities have different sensitivity with respect to head pruning, and (2) the difference in sensitivity varies on different downstream tasks. Specifically, on the ITR-COCO task, pruning 40% heads in the text encoder and the cross-modal encoder does not significantly impact performance while pruning the vision encoder causes a large performance drop. However, the results on NLVR2 show that the text encoder is as important as the image encoder in this task while cross-modal encoders are not very sensitive to head pruning. These results suggest that encoders of different modalities are not created equal in a vision-language model, motivating us to explore modal-specific pruning methods for VLMs.

**Modal-adaptive pruning** A naive way to achieve modal-specific pruning is to manually adjust the pruning percentage of different encoders based on the prior observation. Specifically, we consider a baseline that prunes 30% parameters out of each encoder as the baseline. Then for ITR-COCO, we prune 10% parameters in the vision encoder while pruning 40% parameters in the text and the cross-modal encoder. For NLVR2, we set this percentage to 10%, 10%, and 60% for image, text, and cross-modal encoders, respectively. These percentages are heuristically adjusted according to the previous findings and the empirical performance. Moreover, the relative sparsity is set to ensure the overall sparsity of the model is similar.

sparsity	Text Retrieval			Image Retrieval			NLVR2	
	R@1	R@5	R@10	R@1	R@5	R@10	val	test
.3/.3/.3	76.4	93.4	96.8	58.6	83.2	90.0	78.9	77.9
.1/.4/.4	78.1	94.2	97.1	60.2	84.1	90.5	-	-
.1/.1/.6	-	-	-	-	-	-	80.9	80.9

Table 1: Modal-specific pruning results on NLVR2 and ITR-COCO. All models are trained with pruning and knowledge distillation.

The results are shown in Table 4. We find that manually specifying sparsity levels for different encoders according to their "importance" leads to substantial improvements, demonstrating the effectiveness of modal-specific pruning. However, manually determining the sparsity for different encoders could be laborious and sub-optimal. Therefore, we propose **modal-adaptive pruning**, an end-to-end pruning algorithm using a differentiable approximation of  $L_0$  regularization (Louizos et al., 2017)

to automatically infer the importance of vision and language modalities and adaptively remove redundant structures and neurons in different encoders with controllable target sparsity.

Consider a given neural network model  $f(\cdot; \theta)$  parameterized by  $\theta = \{\theta_j\}_{j=1}^n$ , where each  $\theta_j$  represents an individual parameter weight or a block of weights (e.g. a column of a weight matrix) and  $n$  denotes the number of blocks. By introducing additional binary variables  $\mathbf{z} = \{z_j\}_{j=1}^n$  such that  $z_j \in \{0, 1\}$ , we can formulate the optimization objective as below

$$\min \mathbb{E}_{\mathbf{z}} \left[ \frac{1}{D} \sum_{i=1}^D \mathcal{L}(\mathbf{x}_i, \mathbf{y}_i; \tilde{\theta}) + \lambda \|\tilde{\theta}\|_0 \right] \quad (5)$$

where  $\tilde{\theta} = \{\tilde{\theta}_j\}$  denotes the set of model parameters after pruning and its  $L_0$  norm,  $\|\tilde{\theta}\|_0 = \sum_{j=1}^n z_j$ , measures the effective size of the pruned model.  $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^D$  are training examples,  $\mathcal{L}$  is the training loss function and  $\lambda > 0$  is a constant hyperparameter. The masking variables  $\mathbf{z}$  are learned during training as real numbers in the range  $[0, 1]$ . In contrast, during inference, all the variables that are below a threshold are set to 0 so that our pruned model can achieve the expected sparsity. See Appendix A for more details.

We also adopt knowledge distillation at fine-tuning with pruning stage to help the student model better preserve capacity on downstream tasks. The final training objective is as follows:

$$\mathcal{L}_{\text{ft}} = \lambda \mathcal{L}_{\text{VL}} + (1 - \lambda) \mathcal{L}_{\text{KD}} + \mathcal{L}_{\text{Lgr}} \quad (6)$$

where  $\mathcal{L}_{\text{VL}}$  represents the task-specific fine-tuning loss brought by the re-parameterized student model, the  $\mathcal{L}_{\text{KD}}$  is the task-specific knowledge distillation loss and  $\mathcal{L}_{\text{Lgr}}$  infers to the lagrangian loss.

## 4 Experiments

### 4.1 Baselines

We mainly compare EfficientVLM with two baselines: MiniVLM (Wang et al., 2020a), a compact VLM consists of a lightweight object detection model and a compact Transformers-based vision-language encoder, which is initialized by MiniLM (Wang et al., 2020b), a compressed pre-trained language model; and DistillVLM (Fang et al., 2021), which adopts the same model architecture with MiniVLM and apply knowledge distillation for further boosting model’s performance.

Method	Input Length image/text	End-to-End Time(ms)	Vision Module			Text(and)Fusion Module		
			Para(M)	Time(ms)	FLOPs(B)	Para(M)	Time(ms)	FLOPs(B)
X-VLM <sub>clip</sub>	196/35	17.8	86.1	9.0	18.9	123	8.8(14.2)	4.2
- CPU	-	395.5	-	355.1	-	-	40.4(56.8)	-
OSCAR <sub>B</sub>	50/35	135.2	63.8	121.9	767.0	109	13.3	8.2
- CPU	-	12347.1	-	12300	-	-	47.1	-
MiniVLM	50/35	23.6	7.5	12.2	4.4	34.5	11.4	2.3
- CPU	-	418.2	-	393.9	-	-	24.3	-
ViLT	200/40	21	2.4	0.7	0.6	109	20.3	22.8
- CPU	-	69.1	-	0.8	-	-	68.3	-
EfficientVLM	196/35	9.7	42.0	5.0	8.3	50.3	8.5	1.3
- CPU	-	180.8	-	171.5	-	-	17.1	-

Table 2: Model size and actual inference time for visual feature extractor and vision-language fusion model of compared models. DistilVLM is of the same size and speed as MiniVLM. Actual Inference time is reported on both GPU and CPU.

For reference, we also include the performance of DistilDualEnc (Wang et al., 2021), ViLT (Kim et al., 2021) and X-VLM<sub>small</sub> in our comparison. DistilDualEnc is a dual-encoder VLM distilled from a fusion-based VLM. ViLT is a single-stream VLM that feeds vision features without using region features nor deep convolutional visual embedders and X-VLM<sub>small</sub> use the same initialization as EfficientVLM but trained without knowledge distillation or pruning.

To better illustrate our comparison, Table 2 shows the model size and inference speed of the models compared. We test model inference time<sup>4</sup> on both GPU and CPU devices which are Nvidia Tesla V100 GPU and Intel(R) Xeon(R) Platinum 8260 CPU @2.40GHz, respectively. Since the number of FLOPs is affected by the input sequence length, we show the input image token length and average text length of each model in their settings in the table. We can see that despite the fully Transformer-based visual feature extractor being heavier on model size, it consumes much less time during inference than MiniVLM. As for the Transformer-based text/fusion module, EfficientVLM is slightly larger than MiniVLM and DistilVLM while much faster thanks to the parallel nature of image and text encoders in its architecture. Despite the extremely efficient vision module of ViLT, it consumes more time because of its heavy text and fusion encoder. Specifically, when comparing with their corresponding teacher model, DistilVLM only reduces the inference time of the Transformer encoder by around 15% on GPU, while EfficientVLM achieves a speed-up ratio of  $1.9\times$  on

<sup>4</sup>In practice, the text encoder can be run in parallel with the image encoder while being much faster. Therefore, the inference time of text encoders does not actually contribute to the overall actual inference time of the model.

GPU and  $2.2\times$  on CPU.

## 4.2 Datasets and Tasks

**Pre-training datasets** We construct our pre-training dataset following (Zeng et al., 2021) 4M-setting using two in-domain datasets, COCO (Lin et al., 2014) and Visual Genome (VG) (Krishna et al., 2017), and two out-of-domain datasets, SBU Captions (Ordonez et al., 2011) and Conceptual Captions (CC) (Sharma et al., 2018). Note that we have cleaned the pre-training datasets to avoid data leaks since downstream V+L tasks have overlaps in images with COCO and Visual Genome. The statistics of our pre-training dataset are presented in Appendix B.

**Image-Text Retrieval** There are two subtasks: text retrieval (TR) and image retrieval (IR). We evaluate X-VLM on MSCOCO datasets. We adopt the widely used Karpathy split (Karpathy and Li, 2015) datasets. Following ALBEF and X-VLM, we optimize  $\mathcal{L}_{ITC}$  and  $\mathcal{L}_{ITM}$  and fine-tune the model for 10 epochs. During inference, we first compute  $s(I, T)$  for all images and texts, and then take the top- $k$  candidates and calculate  $p^{\text{match}}(I, T)$  for ranking.  $k$  is set to 256 for MSCOCO following Zeng et al. (2021).

**Visual Question Answering (VQA 2.0)** (Goyal et al., 2017) It requires the model to predict an answer given an image and a question. Following ALBEF and X-VLM, we use a three-layer Transformer decoder initialized by the cross-modal encoder of EfficientVLM to generate answers based on the outputs of the cross-modal encoder. We fine-tune the model for 10 epochs. During inference, we constrain the decoder to only generate from the 3,129 candidate answers following Zeng et al. (2021); Li et al. (2021a).

**Natural Language for Visual Reasoning**

Method	ITR-TR			ITR-IR			NLVR2		VQA 2.0		COCO-Caption			
	R@1	R@5	R@10	R@1	R@5	R@10	val	test	test-dev	test-std	B@4	M	C	S
X-VLM <sub>clip</sub>	79.0	94.5	97.9	61.5	84.6	90.8	83.15	83.48	76.92	77.02	39.4	30.5	131.0	23.6
-98%	77.4	92.6	95.9	60.3	82.9	89.0	81.49	81.81	75.38	75.48	38.6	29.9	128.4	23.1
OSCAR <sub>B</sub>	70.0	91.1	95.5	54.0	80.8	88.5	78.07	78.36	73.4	73.2	36.5	30.3	123.7	23.1
-98%	68.6	89.3	93.6	52.9	79.2	86.7	76.51	76.79	71.93	71.74	35.8	29.7	121.2	22.6
DistilDualEnc	-	-	-	-	-	-	74.16	74.30	68.05	-	-	-	-	-
ViLT	61.5	86.3	92.7	42.7	72.9	83.1	75.7	76.1	71.3	-	-	-	-	-
MiniVLM	58.8	85.1	91.7	45.0	74.1	84.0	73.71	73.93	69.1	69.4	35.6	28.6	119.8	21.6
DistillVLM	58.3	84.1	91.3	43.9	73.7	83.3	-	-	69.8	69.6	35.6	28.7	120.8	22.1
X-VLM <sub>small</sub>	74.5	92.3	96.0	56.1	81.6	88.7	79.34	79.26	73.7	73.93	37.2	29.4	123.4	22.4
EfficientVLM	<b>78.7</b>	<b>94.5</b>	<b>97.5</b>	<b>60.6</b>	<b>84.4</b>	<b>90.5</b>	<b>81.83</b>	<b>81.72</b>	<b>76.2</b>	<b>76.28</b>	<b>38.1</b>	<b>30.1</b>	<b>127.3</b>	<b>23.1</b>

Table 3: Main results on various downstream vision-language tasks. The top groups are teacher models and their 98% performance, which is used for reference. The bottom group contains previous efficient VLMs and the X-VLM<sub>small</sub> baseline.

(NLVR2 (Suhr et al., 2018)) The task prescribes the model to predict whether a text describes the relations between two images. Following ALBEF and X-VLM, we extend the cross-modal encoder to enable reasoning over two images and perform a domain pre-training step for two epochs. We then fine-tune the model for 10 epochs.

**Image Captioning** The task requires a model to generate textual descriptions of input images. We evaluate X-VLM on the COCO Captioning dataset (Chen et al., 2015). We report BLEU-4 (Papineni et al., 2002), METEOR (Denkowski and Lavie, 2014), SPICE (Anderson et al., 2016) and CIDEr (Vedantam et al., 2015) scores on the Karparthy test split. Following Zeng et al. (2021), we simply adapt EfficientVLM to a multi-modal decoder for caption generation. We train EfficientVLM with language modeling loss for two epochs on 4M data. Then, we fine-tune it on the COCO Captioning dataset for 10 epochs.

### 4.3 Experiment Setup

**Teacher Models** We initialized the teacher X-VLM model with a pre-trained CLIP ViT (Radford et al., 2021b) and a pre-trained BERT. We pre-train the X-VLM on 4 million image-text pairs for 200k steps. Then we fine-tune the teacher model on downstream tasks following Zeng et al. (2021).

**Pre-training** We pre-train EfficientVLM on the aforementioned 4 million image-text pairs for 400k steps with  $16 \times$  V100 32G GPU. We adopt AdamW (Loshchilov and Hutter, 2019) optimizer and set the learning rate and weight decay as  $1e-4$  and 0.01 respectively. The batch size is set to 1024.

**Fine-tuning** We combine the modal-adaptive pruning algorithm with knowledge distillation from the fine-tuned teacher models. We set pruning sparsity

at 25%. Other fine-tuning hyper-parameters are presented in the Appendix C.

## 4.4 Experimental Results

### 4.4.1 Main Results

We present the main results in Table 3. The top group of models denotes the base-size VLMs used as the teacher model for different compact VLMs. We also list the 98% performance of these models for better comparison. Specifically, X-VLM<sub>clip</sub><sup>5</sup> is the teacher of EfficientVLM while OSCAR<sub>B</sub> is the teacher of DistillVLM. In the bottom group, we compare EfficientVLM with other efficient vision-language models as well as the X-VLM<sub>small</sub> baseline. We can see that EfficientVLM substantially outperforms all compared models by a large margin despite DistillVLM and MiniVLM being trained with 7 million image-text pairs while EfficientVLM is only trained with 4 million image-text pairs. Specifically, EfficientVLM achieves a R@1 of 78.7% and 60.6% on Image Retrieval and Text Retrieval respectively, accounting for a large absolute improvement of 17.2% and 15.6% compared to the previous compact SoTA VLMs. We also achieve 81.83% and 81.72% accuracy on the validation set and test-P set of NLVR2, respectively, surpassing prior efficient VLMs by a large margin. Similar observation can also be found on VQA 2.0 and COCO Captioning, where EfficientVLM achieves 76.2% accuracy and 76.28 on the test-dev set and test-std set, and 127.3 CIDEr score, respectively. EfficientVLM also consistently outperforms X-VLM<sub>small</sub> by a large margin on all datasets de-

<sup>5</sup>We adopted the first version of X-VLM model as teacher instead of the latest one that uses Swin-Transformer as its vision encoder because the model architecture of Swin-Transformer makes the general distillation more difficult.

Method	ITR-TR			ITR-IR			NLVR2		VQA	COCO-Caption			
	R@1	R@5	R@10	R@1	R@5	R@10	val	test	test-dev	B@4	M	C	S
<i>Ablation Study Results on Pre-train Distillation Objectives</i>													
X-VLM <sub>small</sub>	73.0	91.8	96.0	55.3	81.1	88.6	78.68	78.39	73.39	35.7	29.0	117.9	21.8
+ Logits	76.6	93.4	96.8	58.7	82.9	89.4	81.16	80.97	74.91	36.4	29.5	121.5	22.2
+ Hidden	76.7	93.6	96.8	59.1	83.0	89.7	80.74	81.13	75.12	36.9	29.8	126.2	22.9
+ Attn	76.5	94.1	97.0	59.0	83.0	89.6	81.06	81.01	75.22	37.9	29.8	126.2	22.9
<i>Ablation Study Results on Fine-tuning Objectives</i>													
EfficientVLM	78.7	94.5	97.5	60.6	84.4	90.5	81.83	81.72	76.2	38.1	30.1	127.3	23.1
- KD only	78.2	94.4	97.2	60.4	84.2	90.5	82.73	81.92	76.48	38.2	30.1	127.7	23.1
- Pruning only	77.9	94.3	97.3	59.7	83.8	90.1	80.71	80.47	74.87	6.9	10.9	8.2	3.5
- Fine-tune only	77.5	94.2	97.4	59.2	83.5	89.9	81.56	81.47	75.65	37.7	29.9	126.8	22.9

Table 4: Ablation study results. The top group shows the effects of gradually adding different distilled knowledge at pre-training stage. We take checkpoints at 10w training steps for evaluation. The bottom group presents ablation experiments of pruning and knowledge distillation at fine-tuning stage.

spite being more compact and efficient, demonstrating the effectiveness of the proposed *distilling then pruning framework*. Moreover, we find that EfficientVLM surpasses 98% performance of the teacher model on most datasets. In contrast, DistilVLM underperforms the 98% OSCAR<sub>B</sub> baseline by a large margin. Actually, EfficientVLM recovers 98.4% performance of X-VLM<sub>clip</sub> on average, while DistilVLM only retains 89.3% performance of OSCAR<sub>B</sub> on average. This further confirms the effectiveness of our method.

#### 4.4.2 Ablation Study

We also conduct a series of ablation studies to better understand the effectiveness of EfficientVLM.

**Impact of Knowledge Distillation** We first investigate the impact of different distillation objectives by gradually adding logits distillation, hidden states distillation, and attention distillation starting with X-VLM<sub>small</sub>. The results are shown in the top group of Table 4. We find that adding each component improves the overall performance, demonstrating the effectiveness of combining these components for pre-train distillation.

**Impact of Fine-tuning Objectives** We then study the effect of modal-adaptive pruning and knowledge distillation in the fine-tuning stage. The results are shown in Table 4. First, by comparing the results of EfficientVLM and that in Table 1, we can see that modal-adaptive pruning with learned sparsity for encoders of each modality substantially outperforms manually tuned sparsity. We also find that EfficientVLM performs similarly to the KD-only variant. These results confirm the effectiveness of modal-adaptive pruning. We also find that pruning without distillation results in worse results, demonstrating the necessity of knowledge distillation dur-

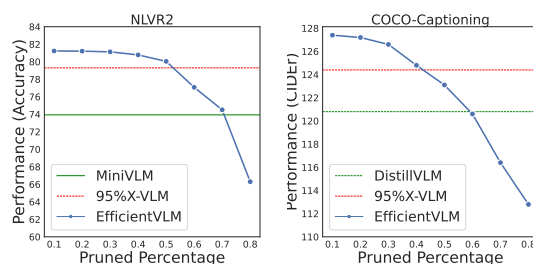


Figure 3: Results on NLVR2 and COCO Captioning tasks with different sparsity ranging from 10% to 80% .

ing fine-tuning. Finally, we can see that simply fine-tuning the compact task-agnostic pre-trained EfficientVLM performs not as well. However, it still outperforms existing baselines by a huge margin. This shows that EfficientVLM can also be used as a good compact task-agnostic VLM.

**Impact of Pruning Sparsity** We also investigate the performance of our modal-adaptive pruning methods with different target sparsity ranging from 10% to 80%. The results are shown in Figure 3. We can see that EfficientVLM retains over 95% performance of the teacher model with a sparsity of 50% and 40% on NLVR2 and COCO Captioning, respectively. EfficientVLM also outperforms the previous best results of compact VLMs with a sparsity up to 70% and 60% on these tasks. This shows EfficientVLM also performs well with larger sparsity.

## 5 Conclusion

We introduce EfficientVLM, a fast and accurate vision-language model trained with a distilling then pruning framework. Empirical results show that EfficientVLM retains 98.4% performance of the base-size teacher model while only preserving



44.3% parameters and achieving a speed-up ratio of  $2.2\times$ . EfficientVLM also achieves a large absolute improvement over previous efficient VLMs, demonstrating a large potential towards lightweight VLMs.

## Limitations

EfficientVLM is applied on X-VLM. However, there are also many recent fully Transformer VLMs achieving comparable or better performance. Therefore, applying our *distilling then pruning* framework on other state-of-the-art VLMs can be interesting. Also, we do not apply quantization or matrix decomposition, which are prevalent model compression techniques.

## Ethics Statement

Our method is used to compress VLMs. Therefore, ethical considerations of VLMs generally apply to our method. We encourage users to assess potential biases before deploying EfficientVLM.

## References

- Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. Spice: Semantic propositional image caption evaluation. In *European conference on computer vision*, pages 382–398. Springer.
- Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. [Bottom-up and top-down attention for image captioning and visual question answering](#). In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 6077–6086. IEEE Computer Society.
- Hangbo Bao, Wenhui Wang, Li Dong, Qiang Liu, Owais Khan Mohammed, Kriti Aggarwal, Subhojit Som, and Furu Wei. 2022. [Vlmo: Unified vision-language pre-training with mixture-of-modality-experts](#).
- Jize Cao, Zhe Gan, Yu Cheng, Licheng Yu, Yen-Chun Chen, and Jingjing Liu. 2020. [Behind the scene: Revealing the secrets of pre-trained vision-and-language models](#). *CoRR*, abs/2005.07310.
- Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. 2015. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*.
- Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Uniter: Universal image-text representation learning. In *European conference on computer vision*, pages 104–120. Springer.
- Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the ninth workshop on statistical machine translation*, pages 376–380.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *North American Chapter of the Association for Computational Linguistics*.
- Cristóbal Eyzaguirre, Felipe del Río, Vladimir Araujo, and Álvaro Soto. 2021. Dact-bert: Differentiable adaptive computation time for an efficient bert inference. *arXiv preprint arXiv:2109.11745*.
- Zhiyuan Fang, Jianfeng Wang, Xiaowei Hu, Lijuan Wang, Yezhou Yang, and Zicheng Liu. 2021. Compressing visual-linguistic model via knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1428–1438.
- Jonathan Frankle and Michael Carbin. 2018. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*.
- Zhe Gan, Yen-Chun Chen, Linjie Li, Tianlong Chen, Yu Cheng, Shuohang Wang, Jingjing Liu, Lijuan Wang, and Zicheng Liu. 2021. [Playing lottery tickets with vision and language](#). *arXiv preprint arXiv:2104.11832*, 2.
- Zhe Gan, Yen-Chun Chen, Linjie Li, Chen Zhu, Yu Cheng, and Jingjing Liu. 2020. Large-scale adversarial training for vision-and-language representation learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.
- Mitchell A Gordon, Kevin Duh, and Nicholas Andrews. 2020. [Compressing bert: Studying the effects of weight pruning on transfer learning](#). *arXiv preprint arXiv:2002.08307*.
- Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913.
- Alex Graves. 2016. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*.
- Demi Guo, Alexander M Rush, and Yoon Kim. 2020. [Parameter-efficient transfer learning with diff pruning](#). *arXiv preprint arXiv:2012.07463*.

- Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7).
- Zhicheng Huang, Zhaoyang Zeng, Yupan Huang, Bei Liu, Dongmei Fu, and Jianlong Fu. 2021. Seeing out of the box: End-to-end pre-training for vision-language representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12976–12985.
- Zhicheng Huang, Zhaoyang Zeng, Bei Liu, Dongmei Fu, and Jianlong Fu. 2020. Pixel-bert: Aligning image pixels with text by deep multi-modal transformers. *arXiv preprint arXiv:2004.00849*.
- Huaizu Jiang, Ishan Misra, Marcus Rohrbach, Erik Learned-Miller, and Xinlei Chen. 2020. In defense of grid features for visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10267–10276.
- Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2019. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351*.
- Andrej Karpathy and Fei-Fei Li. 2015. Deep visual-semantic alignments for generating image descriptions. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 3128–3137. IEEE Computer Society.
- Wonjae Kim, Bokyung Son, and Ildoo Kim. 2021. Vilt: Vision-and-language transformer without convolution or region supervision. In *International Conference on Machine Learning*, pages 5583–5594. PMLR.
- Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73.
- François Lagunas, Ella Charlaix, Victor Sanh, and Alexander M Rush. 2021. Block pruning for faster transformers. *arXiv preprint arXiv:2109.04838*.
- Yann LeCun, John Denker, and Sara Solla. 1989. Optimal brain damage. *Advances in neural information processing systems*, 2.
- Gen Li, Nan Duan, Yuejian Fang, Ming Gong, and Daxin Jiang. 2020a. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11336–11344.
- Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. 2021a. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34:9694–9705.
- Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*.
- Wei Li, Can Gao, Guocheng Niu, Xinyan Xiao, Hao Liu, Jiachen Liu, Hua Wu, and Haifeng Wang. 2021b. UNIMO: Towards unified-modal understanding and generation via cross-modal contrastive learning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2592–2607, Online. Association for Computational Linguistics.
- Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. 2020b. Oscar: Object-semantic aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*, pages 121–137. Springer.
- Yehao Li, Jiahao Fan, Yingwei Pan, Ting Yao, Weiyao Lin, and Tao Mei. 2022. Uni-eden: Universal encoder-decoder network by multi-granular vision-language pre-training. *ACM Trans. Multim. Comput. Commun. Appl.*, 18(2):48:1–48:16.
- Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.
- Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Haotang Deng, and Qi Ju. 2020. Fastbert: a self-distilling bert with adaptive inference time. *arXiv preprint arXiv:2004.02178*.
- Yongfei Liu, Chenfei Wu, Shao-yen Tseng, Vasudev Lal, Xuming He, and Nan Duan. 2021. Kd-vlp: Improving end-to-end vision-and-language pretraining with object knowledge distillation. *arXiv preprint arXiv:2109.10504*.
- Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.
- Christos Louizos, Max Welling, and Diederik P Kingma. 2017. Learning sparse neural networks through  $l_0$  regularization. *arXiv preprint arXiv:1712.01312*.
- Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vlbart: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13–23.

- Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. 2017. The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. In *Proceedings of ICLR*.
- Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? *Advances in neural information processing systems*, 32.
- Vicente Ordonez, Girish Kulkarni, and Tamara L. Berg. 2011. Im2text: Describing images using 1 million captioned photographs. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain*, pages 1143–1151.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021a. [Learning transferable visual models from natural language supervision](#). *CoRR*, abs/2103.00020.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021b. [Learning transferable visual models from natural language supervision](#). In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.
- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.
- Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. 2015. Faster R-CNN: towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 91–99.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.
- Victor Sanh, Thomas Wolf, and Alexander Rush. 2020. Movement pruning: Adaptive sparsity by fine-tuning. *Advances in Neural Information Processing Systems*, 33:20378–20389.
- Roy Schwartz, Jesse Dodge, Noah A Smith, and Oren Etzioni. 2020a. Green ai. *Communications of the ACM*, 63(12):54–63.
- Roy Schwartz, Gabriel Stanovsky, Swabha Swayamdipta, Jesse Dodge, and Noah A Smith. 2020b. The right tool for the job: Matching model and instance complexities. *arXiv preprint arXiv:2004.07453*.
- Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. [Conceptual captions: A cleaned, hypernamed, 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, Melbourne, Australia. Association for Computational Linguistics.
- Sheng Shen, Zhen Dong, Jiayu Ye, Linjian Ma, Zhewei Yao, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. 2020. Q-bert: Hessian based ultra low precision quantization of bert. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8815–8821.
- Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. 2019. Vi-bert: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*.
- Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. 2018. A corpus for reasoning about natural language grounded in photographs. *arXiv preprint arXiv:1811.00491*.
- Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. Patient knowledge distillation for bert model compression. *arXiv preprint arXiv:1908.09355*.
- Hao Tan and Mohit Bansal. 2019. [LXMERT: Learning cross-modality encoder representations from transformers](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5100–5111, Hong Kong, China. Association for Computational Linguistics.
- Surat Teerapittayanon, Bradley McDanel, and Hsiang-Tsung Kung. 2016. Branchynet: Fast inference via early exiting from deep neural networks. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 2464–2469. IEEE.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575.
- Jianfeng Wang, Xiaowei Hu, Pengchuan Zhang, Xijun Li, Lijuan Wang, Lei Zhang, Jianfeng Gao, and Zicheng Liu. 2020a. Minivlm: A smaller

- and faster vision-language model. *arXiv preprint arXiv:2012.06946*.
- Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. 2022. Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. *arXiv preprint arXiv:2202.03052*.
- Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020b. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788.
- Zekun Wang, Wenhui Wang, Haichao Zhu, Ming Liu, Bing Qin, and Furu Wei. 2021. Distilled dual-encoder model for vision-language understanding. *arXiv preprint arXiv:2112.08723*.
- Ziheng Wang, Jeremy Wohlwend, and Tao Lei. 2019. Structured pruning of large language models. *arXiv preprint arXiv:1910.04732*.
- Mengzhou Xia, Zexuan Zhong, and Danqi Chen. 2022. Structured pruning learns compact and accurate models. *arXiv preprint arXiv:2204.00408*.
- Ji Xin, Raphael Tang, Jaejun Lee, Yaoliang Yu, and Jimmy Lin. 2020. Deebert: Dynamic early exiting for accelerating bert inference. *arXiv preprint arXiv:2004.12993*.
- Canwen Xu, Wangchunshu Zhou, Tao Ge, Furu Wei, and Ming Zhou. 2020. Bert-of-theseus: Compressing bert by progressive module replacing. *arXiv preprint arXiv:2002.02925*.
- Canwen Xu, Wangchunshu Zhou, Tao Ge, Ke Xu, Julian McAuley, and Furu Wei. 2021a. [Beyond preserved accuracy: Evaluating loyalty and robustness of BERT compression](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10653–10659, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Canwen Xu, Wangchunshu Zhou, Tao Ge, Ke Xu, Julian McAuley, and Furu Wei. 2021b. Beyond preserved accuracy: Evaluating loyalty and robustness of bert compression. *arXiv preprint arXiv:2109.03228*.
- Haiyang Xu, Ming Yan, Chenliang Li, Bin Bi, Songfang Huang, Wenming Xiao, and Fei Huang. 2021c. [E2E-VLP: End-to-end vision-language pre-training enhanced by visual learning](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 503–513, Online. Association for Computational Linguistics.
- Jingjing Xu, Wangchunshu Zhou, Zhiyi Fu, Hao Zhou, and Lei Li. 2021d. A survey on green deep learning. *arXiv preprint arXiv:2111.05193*.
- Ofir Zafrir, Guy Boudoukh, Peter Izsak, and Moshe Wasserblat. 2019. Q8bert: Quantized 8bit bert. In *2019 Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing-NeurIPS Edition (EMC2-NIPS)*, pages 36–39. IEEE.
- Yan Zeng, Xinsong Zhang, and Hang Li. 2021. Multi-grained vision language pre-training: Aligning texts with visual concepts. *arXiv preprint arXiv:2111.08276*.
- Yan Zeng, Xinsong Zhang, Hang Li, Jiawei Wang, Jipeng Zhang, and Wangchunshu Zhou. 2022. [X<sup>2</sup>-vlm: All-in-one pre-trained model for vision-language tasks](#).
- Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. 2021. Vinvl: Revisiting visual representations in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5579–5588.
- Wangchunshu Zhou, Yuchen Eleanor Jiang, Ryan Cotterell, and Mrinmaya Sachan. 2023. [Efficient prompting via dynamic in-context learning](#).
- Wangchunshu Zhou, Canwen Xu, Tao Ge, Julian McAuley, Ke Xu, and Furu Wei. 2020. Bert loses patience: Fast and robust inference with early exit. *Advances in Neural Information Processing Systems*, 33:18330–18341.
- Wangchunshu Zhou, Canwen Xu, and Julian McAuley. 2022a. Bert learns to teach: Knowledge distillation with meta learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7037–7049.
- Wangchunshu Zhou, Yan Zeng, Shizhe Diao, and Xinsong Zhang. 2022b. [VLUE: A multi-task multi-dimension benchmark for evaluating vision-language pre-training](#). In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 27395–27411. PMLR.

## A Differentiable $L_0$ -Norm Regularization

The formulation of Equation 5 is still hard for gradient-based optimization by the discrete nature of masks, but the expectation provides some guidance for empirically effective relaxations. Following prior work (Louizos et al., 2017; Wang et al., 2019; Guo et al., 2020), we apply Hard-Concrete distribution (Maddison et al., 2017) to relax  $\mathbf{z}$  into continuous space  $[0, 1]^d$ . Specifically,  $\mathbf{z}$  is now defined to be a deterministic and (sub)differentiable function of a sample  $\mathbf{u}$  from a uniform distribution,

$$\begin{aligned}\mathbf{u} &\sim U(0, 1) \\ \mathbf{s} &= \text{sigmoid}(\log \mathbf{u} - \log(1 - \mathbf{u}) + \boldsymbol{\alpha}) \\ \bar{\mathbf{s}} &= \mathbf{s} \times (r - l) + l \\ \mathbf{z} &= \min(1, \max(0, \bar{\mathbf{s}}))\end{aligned}$$

Here  $l < 0$  and  $r > 1$  are two constants used to stretch  $\mathbf{s}$  into the interval  $(l, r)^d$  before it is clamped to  $[0, 1]^d$  with the  $\min(\mathbf{1}, \max(\mathbf{0}, \cdot))$  operation. In this case we have a differentiable closed-form expression for the expected  $L_0$ -norm,

$$\begin{aligned}\mathbb{E} \left[ \|\tilde{\boldsymbol{\theta}}\|_0 \right] &= \sum_{j=1}^n \mathbb{E} [z_j > 0] \\ &= \sum_{j=1}^n \text{sigmoid} \left( \alpha_j - \log \frac{-l}{r} \right) \quad (7)\end{aligned}$$

To better control the expected sparsity of the student model, we follow Wang et al. (2019) to replace the vanilla  $l_0$  objective with a Lagrangian multiplier. Let  $t$  be the target model size and  $s(\boldsymbol{\alpha})$  be the constrained model size determined by the Hard Concrete parameter  $\boldsymbol{\alpha}$ .

The Lagrangian method imposes an equality constraint  $s(\boldsymbol{\alpha}) = t$  by introducing a violation penalty,

$$\mathcal{L}_{\text{Lgr}} = \lambda_1 \cdot (s(\boldsymbol{\alpha}) - t) + \lambda_2 \cdot (s(\boldsymbol{\alpha}) - t)^2$$

where  $\lambda_1, \lambda_2 \in \mathbb{R}$  are two Lagrangian multipliers that will be jointly updated during training.

## B Pre-train Datasets

Dataset	# Images	# Captions	# Ann
COCO	0.11M	0.55M	0.45M
VG	0.10M	-	5.7M
SBU	0.86M	0.86M	-
CC-3M	2.9M	2.9M	-

Table 5: Statistics of the pre-training datasets.

## C Hyperparameters

The hyperparameters to reproduce fine-tuning results are in Table 6. Tasks with \* need two-stage fine-tuning.

Tasks	Learning Rate	Batch Size	Epoch
ITR-COCO	3e-5	384	10
NLVR*	3e-5	80	10
Captioning*	1e-5	256	5
VQA	5e-5	192	10

Table 6: Hyper-parameters for fine-tuning on downstream tasks.

## ACL 2023 Responsible NLP Checklist

---

### A For every submission:

- A1. Did you describe the limitations of your work?  
6
- A2. Did you discuss any potential risks of your work?  
*As far as we know, our proposed method doesn't have any risks.*
- A3. Do the abstract and introduction summarize the paper's main claims?  
0,1
- A4. Have you used AI writing assistants when working on this paper?  
*Left blank.*

### B Did you use or create scientific artifacts?

2, 3, 4

- B1. Did you cite the creators of artifacts you used?  
2,3,4
- B2. Did you discuss the license or terms for use and / or distribution of any artifacts?  
*The artifacts we used are publicly available. Researchers and engineers commonly use them.*
- B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?  
3
- B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?  
*We used the data following previous works.*
- B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?  
*The data we used are publicly available. We used the data following previous works.*
- B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.  
*Appendix*

### C Did you run computational experiments?

4

- C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?

4

*The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.*

- C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?

4

- C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?

4

- C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?

4

**D  Did you use human annotators (e.g., crowdworkers) or research with human participants?**

*Left blank.*

- D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?

*No response.*

- D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?

*No response.*

- D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?

*No response.*

- D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?

*No response.*

- D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?

*No response.*