# Rethinking Pruning for Vision-Language Models: Strategies for Effective Sparsity and Performance Restoration

**Anonymous ACL submission**

## Abstract

Vision-Language Models (VLMs) integrate information from multiple modalities and have shown remarkable success across various tasks. However, deploying large-scale VLMs in resource-constrained scenarios is challenging. Pruning followed by finetuning offers a potential solution but remains underexplored for VLMs. This study addresses two key questions: how to distribute sparsity across different modality-specific models, and how to restore the performance of pruned sparse VLMs. Our preliminary studies identified two effective pruning settings: applying the same sparsity to both vision and language models, and pruning only the language models. While LoRA finetuning aims to restore sparse models, it faces challenges due to incompatibility with sparse models, disrupting the pruned sparsity. To overcome these issues, we propose SparseLoRA, which applies sparsity directly to LoRA weights. Our experimental results demonstrate significant improvements, including an 11.3% boost under 2:4 sparsity and a 47.6% enhancement under unstructured 70% sparsity. Code and scripts will be released upon acceptance.

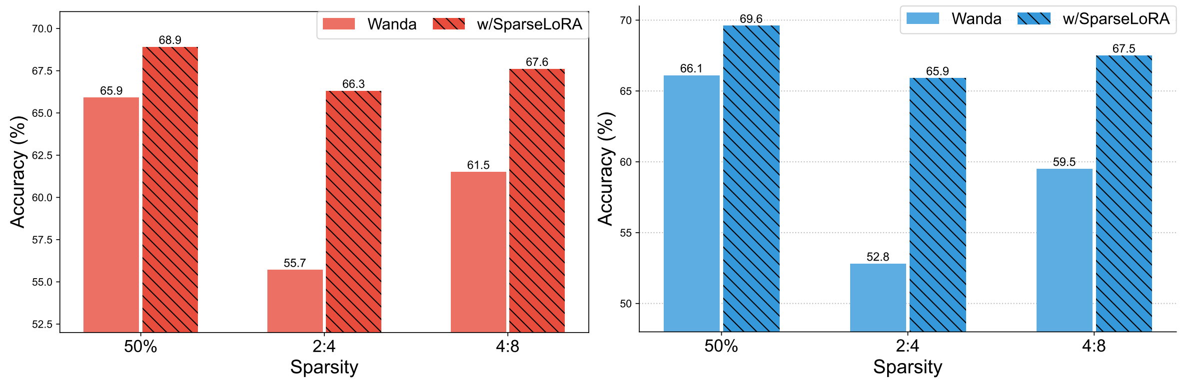
## 1 Introduction

Scaling deep learning models has demonstrated promising performance across various tasks in both vision and language domains (Brown et al., 2020; Jiang et al., 2024; Zhu et al., 2023). Vision-Language Models (VLMs) (Radford et al., 2021; Li et al., 2022; Liu et al., 2023), which leverage powerful vision and language models, have recently garnered significant attention in research (OpenAI et al., 2024; Liu et al., 2024), showcasing their cross-modality capabilities. However, the ever-increasing size of these models comes with substantial computational and memory costs, limiting their practical applicability in resource-constrained environments. Model pruning followed by finetuning (Dai et al., 2018; Fang et al., 2023; Tanaka et al., 2020), which reduces model size while preserving performance, holds promise for improving the real-world deployment of VLMs.

While pruning followed by finetuning has significantly improved the efficiency of vision models (Frankle and Carbin, 2019; Kusupati et al., 2020; Lee et al., 2019) and language models (Chen et al., 2020; Sun et al., 2023a; Frantar and Alistarh, 2023), the realm of Vision-Language Models (VLMs) remains relatively unexplored in terms of model pruning, prompting following questions: *how to distribute sparsity ratios between different modality-specific models* and *how to restore the performance of prune sparse VLMs*.

For the first question, we conducted empirical studies on pruning modality-specific models, experimenting with various combinations of sparsity ratios. Surprisingly, we found that applying the same sparsity ratios to both vision and language models yields nearly optimal performance. On the other hand, since language models are usually much larger than vision models, pruning only the language models offers a beneficial trade-off between performance and efficiency. However, as sparsity ratios increase, pruning significantly degrades performance, especially with structured sparsity patterns (e.g., N: M sparsity (Zhang et al., 2022; andYukun Ma et al., 2021)), underscoring the importance of post-pruning restoration.

While parameter-efficient LoRA finetuning has been proposed to repair the performance of sparse models, it faces a significant challenge due to the incompatibility of dense LoRA modules with sparse models. Merging LoRA modules with sparse models would destroy the sparse pattern, while maintaining LoRA modules would introduce extra latency and slow down the inference speed. To address the incompatibility issue of LoRA, we introduce SparseLoRA finetuning, which utilizes binary masks on LoRA weights, allowing seamless inte-




(a) InstructBLIP-FlanT5$_{XL}$ (b) InstructBLIP-Vicuna-7B

Figure 1: **The comparison of pruned VLMs ("Wanda") and restored VLMs ("w/SparseLoRA")** on multimodal tasks, taking InnstructBLIP (Dai et al., 2023) as the backbone.

gration with pruned weights.

Extensive experiments showcase the effectiveness of our proposed methods in repairing the performance of pruned sparse VLMs. For instance, as illustrated in Figure 1, SparseLoRA boosts the performance by 13.1% for InstructBLIP-Vicuana-7B with 2:4 sparsity. In summary, our contributions are threefold:

- We empirically study the modality-specific sparsity distributions and systematically demonstrate how sparsity affects the performance of VLMs.
- We propose a pipeline involving pruning and post-finetuning with SparseLoRA to restore pruned models.
- Extensive experiments validate the effectiveness and universality of SparseLoRA across various VLMs and tasks.

## 2 Related Work

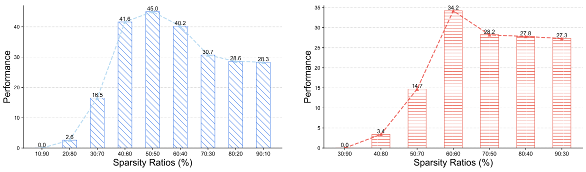
**Vision-Language Models.** Vision-language models, among the most sophisticated multi-modal architectures, have demonstrated outstanding performance across various cross-modality tasks, including image captions (Sharma et al., 2018), image retrieval (Plummer et al., 2015), visual QA (Kim et al., 2016), and image/video generation (Zhou et al., 2021; Singer et al., 2022). These models typically freeze the pretrained vision and language components, only fine-tuning a small, learnable interface (e.g., Qformer in BLIP-2 (Li et al., 2023)) to facilitate inter-modality interactions (Yin et al., 2023; Li et al., 2023), thus avoiding high training costs and potential catastrophic forgetting (Goodfellow et al., 2014).

**Model Pruning for Large Language Models.** While large vision and language models have shown promising advancements, their massive parameter sizes present challenges for practical deployment (Ma et al., 2023; Wang et al., 2020). To mitigate this, model pruning techniques have been introduced to remove redundant weights or structures (Han et al., 2016; Alvarez and Salzmann, 2016). The primary aim of model pruning is to minimize the disparity between models before and after pruning (Liu et al., 2021; He and Xiao, 2024; Frantar and Alistarh, 2023). Various metrics, such as magnitude, gradient (Yi-Lin Sung, 2024), and activation (Sun et al., 2023a), have been proposed to identify unimportant weights. However, pruning without finetuning often leads to a performance drop. (Zhang et al., 2024) utilize reconstruction errors-based metrics to update the weights. Other than the disparity between sparse models and dense models, our method also considers the task-specific objective of repairing sparse models and knowledge distillation from the original full models.

## 3 Preliminary Study

Vision-Language Models (VLMs) consist of modality-specific foundation models, namely visual and language models, as well as a cross-modality interface (e.g., QFormer (Li et al., 2023)) that aligns models from different modalities. Following (Yi-Lin Sung, 2024), we focus on pruning the vision and language models while keeping the Q-Former intact, as it is sufficiently lightweight. Parameters are not evenly distributed across the different modality-specific models; for instance, visual models are often considerably smaller than their corresponding language models (Li et al., 2023; Dai et al., 2023; Liu et al., 2023; Yang et al., 2022).

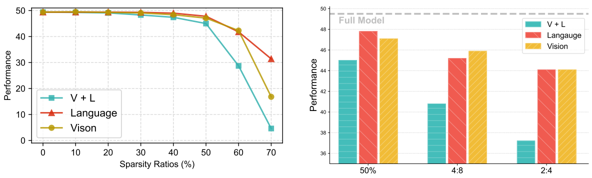
In this case, we pose two questions: (1) *how to distribute sparsity ratios between modality-specific models*, and (2) *how do different sparsity ratios affect the performance of VLMs?*




(a) $s_v + s_l = 100\%$ (b) $s_v + s_l = 120\%$

Figure 2: **Performance of BLIP-2 with different modality-specific sparsity distribution**. We denote the sparsity ratios for the vision and language modalities as "$s_v$:$s_l$". We adjust their distribution while constraining their summation "$s_v + s_l$" to be (a) 100% and (b) 120%.

For the first question, we first try various sparsity ratio combinations between visual models and language models. Specifically, we fix the summation of $s_v$ and $s_l$ and then adjust their distributions accordingly. We use Wanda (Sun et al., 2023a) as the default pruning method because it ensures relatively high performance and efficiency. Based on Figure 2, we found: (1) VLMs would collapse when the language models are under high sparsity ratios (i.e., $s_l > 70\%$), whereas sparsity imposed on visual models has a comparatively lower impact on performance; (2) When constrained by the summation of sparsity ($s_v + s_l$), pruning the modality-specific models with equal sparsity ratios leads to optimal performance.




(a) Unstructured Sparsity (b) Structured Sparsity

Figure 3: **Performance of BLIP-2 with different sparse ratios** (i.e., unstructured pruning and N:M pruning) for visual question-answering tasks.

For the second question, we initially prune VLMs with different unstructured sparsity ratios using the following strategies: pruning language models and visual models with the same sparsity ratios ("V + L"), pruning visual models only ("Vision"), and pruning language models only ("Language"). According to Figure 3a, when sparsity ratios exceed 50%, all settings experience a significant performance drop, although VLMs pruned by a single modality model maintain relatively high performance.

Similarly, in Figure 3b when employing structured N:M sparsity (andYukun Ma et al., 2021; Zhang et al., 2022) (i.e., in each contiguous block of $M$ values, $N$ values must be zero), all models encounter significant performance degradation and even collapse (2:4 for pruning both vision models and language models). This situation prompts us to reflect on *how to restore the pruning-caused performance degradation for VLMs.*

## 4 Methodology

In this section, we will develop a pipeline that involves pruning and post restoration, with the illustration in Figure 4.

### 4.1 Pruning with Few Samples

Model pruning identifies less important weights using predefined metrics (Han et al., 2016), typically measuring the reconstruction errors (Sun et al., 2023a; Frantar and Alistarh, 2023; Zhang et al., 2024) between models before and after pruning, such as magnitude, gradient, and activation. Calculating gradients or activation requires a small calibration dataset $\mathcal{D}_p$ with few samples. With predefined metric $\mathcal{S}$ and the calibration dataset, the weights of a model is scored as follows:

$$S \leftarrow \mathcal{S}(\boldsymbol{W}_0, \mathcal{D}_p), \tag{1}$$

where $\boldsymbol{W}_0$ denote the weights of the model while $S$ represent the importance scores for $\boldsymbol{W}_0$. Given the sparse ratios $s$, binary masks are utilized to locate the pruned weights and update the weights as follows:

$$\boldsymbol{M} \leftarrow (S > \tau), \quad \boldsymbol{W} \leftarrow \boldsymbol{W}_0 \odot \boldsymbol{M}, \tag{2}$$

where $\boldsymbol{W}$ denotes the pruned weights, while $\tau$ represents the threshold ($s$ percentile of $S$) and all weights with scores lower than $s$ will be removed. While the pruning metrics $\mathcal{S}$ aim to minimize reconstruction errors (Sun et al., 2023a; Zhang et al., 2024) or maintain performance (Yi-Lin Sung, 2024), model pruning often results in a significant performance drop and therefore needs to be recovered.

### 4.2 Sparse LoRA finetuning

VLMs, which incorporate both vision models and language models, are often too large to be fine-tuned through full-model fine-tuning techniques (Li

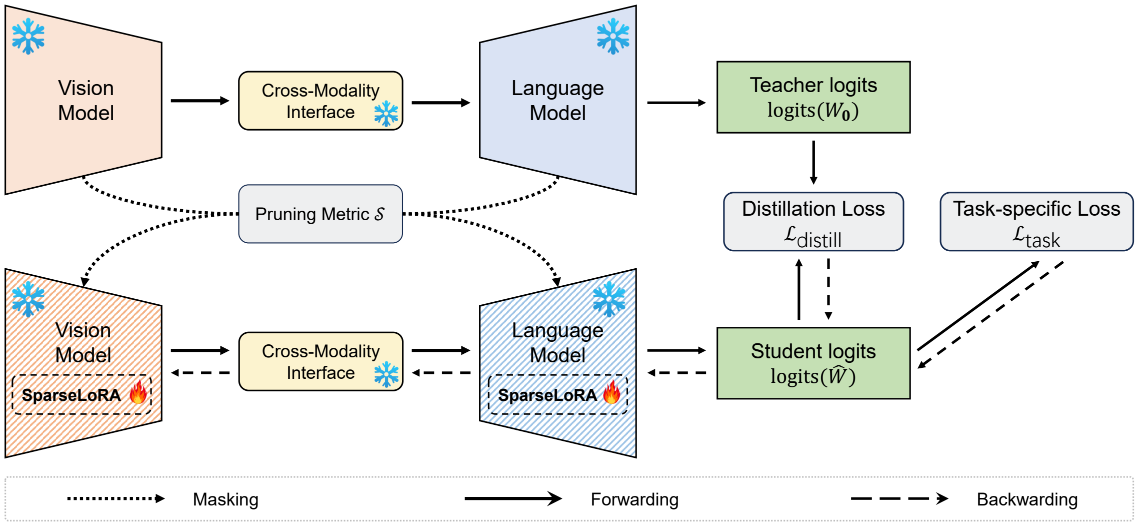



Figure 4: **Visualization of the pipeline of VLM Pruning and SparseLoRA finetuning,** which first prunes the vision model and language model based on a given pruning metric, then restores the pruned via SparseLoRA finetuning.

et al., 2023; Dai et al., 2023). Instead, parameter-efficient fine-tuning techniques (Houlsby et al., 2019; Mangrulkar et al., 2022; Hu et al., 2022) are employed to reduce the number of trainable parameters while maintaining comparable performance. Among these techniques, LoRA (Hu et al., 2022) stands out as one of the most widely used approaches. since it not only efficiently utilizes parameters but also allows for seamless integration with the original weights, thus avoiding potential latency during inference (Dery et al., 2024; Rücklé et al., 2021).

Traditional LoRA fine-tuning involves freezing the parameters of the pretrained model and injecting trainable rank decomposition matrices into each layer that requires fine-tuning. LoRA modules involves two small low-rank trainable weights $A$ and $B$, which can be merged with $\boldsymbol{W}$ after finetuning:

$$\boldsymbol{W} \leftarrow \boldsymbol{W} + \Delta\boldsymbol{W}, \quad \text{where } \Delta\boldsymbol{W} = BA. \tag{3}$$

However, as shown in Figure 5, the sparse pattern of pruned models would collapse after merging (Dery et al., 2024; He et al., 2023). Given that $\Delta\boldsymbol{W}$ are dense weights and $\boldsymbol{W}$ are sparse weights, the element-wise operation would destroy the sparse patterns. Additionally, without merging, the injected LoRA modules would increase latency and slow down inference speed (Mundra et al., 2023; Rücklé et al., 2021; Dery et al., 2024). Inspired by (He et al., 2022), we propose employing masks on $\boldsymbol{W}$ to preserve the sparse pattern:

$$\hat{\boldsymbol{W}} = \boldsymbol{W} + \Delta\boldsymbol{W} \odot \boldsymbol{M}. \tag{4}$$

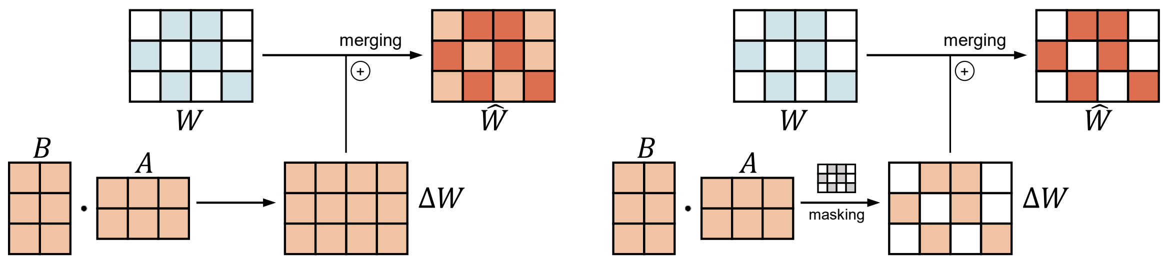
In such a case, $\Delta\boldsymbol{W}$ corresponding to pruned positions are masked and cannot be updated via gradient-backpropagation. Consequently, during backpropagation, during backpropagation, $A$ and $B$ can be optimized as follows:

$$\begin{aligned} B &\leftarrow B + \eta \cdot (\frac{\partial \mathcal{L}}{\partial \hat{\boldsymbol{W}}} \odot \boldsymbol{M}) A^T, \\ A &\leftarrow A + \eta \cdot B^T (\frac{\partial \mathcal{L}}{\partial \hat{\boldsymbol{W}}} \odot \boldsymbol{M}), \end{aligned} \tag{5}$$

where $\mathcal{L}$ denotes the loss and $\eta$ denotes the learning rate. After fine-tuning, SparseLoRA first prunes $\Delta\boldsymbol{W}$ with binary masks and then incorporate it with the pruned weights $\boldsymbol{W}$: $\boldsymbol{W} \leftarrow \hat{\boldsymbol{W}} = \boldsymbol{W} + BA \odot \boldsymbol{M}$. The adaptation of SparseLoRA fine-tuning ensures the sparsity of incremental weights, thus preserving the sparse pattern after merging. Other than the vision model and language model, VLMs also involve small learnable interfaces (e.g., QFormer (Li et al., 2023; Dai et al., 2023)) that align vision models and language models. Because of this, we also insert LoRA into the QFormer, which enhances cross-modality adaptation with minimal additional computational overhead.

### 4.3 Finetuning Objectives

To recover the performance of pruned VLMs, we introduce two finetuning objectives. Firstly, acknowledging the performance gap, we continue to finetune VLMs on the pretraining tasks by minimizing loss $\mathcal{L}_{\text{task}}$ to restore task-specific performance. On the other hand, we propose distilling knowledge (Hinton et al., 2015; Gou et al., 2021; Stanton




(a) LoRA (b) SparseLoRA

Figure 5: **Schematic comparison of (a) LoRA and (b) SparseLoRA.** With masking, SparseLoRA preserves the sparse patterns, while LoRA destroys them after merging.

et al., 2021) from the original models to the pruned models by constraining the KL divergence between their outputs. The distillation loss $\mathcal{L}_{\text{distill}}$ is formulated as follows:

$$\mathcal{L}_{\text{distill}} = D_{\text{KL}}\left(\text{logits}\left(\hat{\boldsymbol{W}}\right) \parallel \text{logits}\left(\boldsymbol{W}_0\right)\right), \tag{6}$$

where $D_{\text{KL}}$ represents the KL-divergence distance and logits$(\boldsymbol{W}_0)$ denotes the output logits of the model with weights $\boldsymbol{W}_0$. Based on the original model with weights $\boldsymbol{W}_0$, both logits $\left(\hat{\boldsymbol{W}}\right)$ and logits $(\boldsymbol{W}_0)$ can be obtained by forwarding with $\boldsymbol{W}_0$ and $(\boldsymbol{W}_0 + BA) \odot \boldsymbol{M}$ separately. This avoids hosting additional weights during training. The overall optimization objective of SparseLoRA is:

$$\mathcal{L} = \lambda \mathcal{L}_{\text{task}} + (1 - \lambda)\mathcal{L}_{\text{distill}}, \tag{7}$$

where $\lambda$ is a scalar weight. The procedure of VLM pruning and SparseLoRA is shown in Figure 4.

## 5 Experimental Setup

**Architectures.** We use multiple multi-modal architectures for experiments including BLIP-2 (Li et al., 2023) and InstructBLIP (Dai et al., 2023), which composes of pretrained EVA-ViT (ViT-g/14 from EVA-CLIP) (Sun et al., 2023b) and pretrained large language models (i.e., FlanT5 (Chung et al., 2022) and Vicuna (Chiang et al., 2023)).

**Evaluation Datasets and Metrics.** We evaluate the zero-shot ability of BLIP-2 and InstructBLIP on various datasets after pruning. We use VQAv2 (Goyal et al., 2016), OK-VQA (Marino et al., 2019), and GQA (Hudson and Manning, 2019) for visual question answering, NoCaps (Agrawal et al., 2019) for image captioning, and Flickr30k (Plummer et al., 2015) for image-text retrieval. We use CIDEr and SPICE to evaluate image captioning tasks and use TR@1 (top-1 text recall) and IR@1 (top-1 image recall) for image retrieval tasks.

**Calibration and Training Datasets.** Following (Yi-Lin Sung, 2024; Liu et al., 2023), our approach leverages a small subset of CC3M (Sharma et al., 2018) for calibration and training data. The number of training samples ranges from 1k to 10k, while the number of calibration samples is 128, which has been shown to be sufficient for pruning (Sun et al., 2023a; Frantar and Alistarh, 2023; Zhang et al., 2024; Yi-Lin Sung, 2024).

**Finetuning Details** We use Adam (Kingma and Ba, 2015) as the optimizer with $\beta_1$, $\beta_2$ = 0.9, 0.999. For regularization, we set the $\lambda$ as 0.1 and grid-search the learning rate from {1e-5, 2e-5, 5e-5, 1e-4, 2e-4}, where we warm up the learning rate in the first 10% steps (of the total training steps). For different model scales, we select a batch size from {16, 32, 64}, and finetune 1 epoch, which is enough for convergence. We perform a grid search for the rank of SparseLoRA, considering values from {4, 8, 16, 32}. By trial and error, we found that a rank of 4 suffices for the QFormer and the vision model, while a rank of 8 optimally suits the language model.

**Baselines.** We consider several pruning techniques, including Global Magnitude Pruning, Gradient-based Pruning, SparseGPT (Frantar and Alistarh, 2023), and Wanda (Sun et al., 2023a). Global Magnitude Pruning prunes are based on weight magnitude, while Gradient-based Pruning prunes use the product of first-order gradient and weight magnitude (Yi-Lin Sung, 2024). SparseGPT is a layer-wise Hessian-based method, and Wanda utilizes weight magnitude and input activation norm for layer-wise pruning. Additionally, we compare against ECoFLaP (Yi-Lin Sung, 2024), which adopts a zero-order gradient-based layer-wise sparsity for vision-language models. We also compare SparseLoRA against DS⊘T (Zhang et al., 2024) that updates the masks after pruning.

Table 1: **Comparison of Full Model, pruned models, and retrained pruned models on the zero-shot performance with BLIP-2 (Li et al., 2023) at 50% sparsity.** Metrics include accuracy for visual question answering, CIDEr and SPICE for image captioning, and TR@1 (text recall) and IR@1 (image recall) for image retrieval. Results are averaged over 5 runs, with the best-performing results marked in **bold** (Full Model not included).

| Method | Sparsity | Param. | Visual Question Answering | | | Image Captioning | | Image retrieval | | Macro Avg. |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | VQAv2 | OK-VQA | GQA | NoCaps | | Flickr30k | | |
| | | | Accuracy | | | CIDEr | SPICE | TR@1 | IR@1 | |
| Full Model | 0% | 3.9B | 63.1 | 41.1 | 44.1 | 105.4 | 13.8 | 96.1 | 87.5 | 64.4 |
| Magnitude | 50% | 2.1B | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.2 | 0.1 | 0.0 |
| Gradient | | | 55.1 | 35.7 | 39.8 | 92.3 | 11.6 | 91.4 | 81.6 | 58.2 |
| SparseGPT | | | 56.1 | 35.5 | 40.6 | 98.7 | 13.3 | 95.8 | 86.2 | 60.9 |
| Wanda | | | 57.7 | 35.4 | 41.9 | 100.1 | 13.4 | 95.2 | 84.5 | 61.2 |
| ECoFLaP | | | 57.5 | 36.2 | 42.1 | 99.0 | 12.5 | 95.7 | 85.8 | 61.3 |
| Wanda + DSØT | 50% | 2.1B | 57.3 | 35.5 | 42.5 | 100.9 | 13.3 | 95.3 | 85.4 | 61.5 |
| Wanda + SparseLoRA | | | **61.2** | **39.5** | **43.5** | **106.6** | **14.1** | **96.0** | **87.2** | **64.0** |

Table 2: **Performance comparison of pruning single modality** on InstructBLIP-Vicuna-7B.

| Method | Param. | VQA | | NoCaps | |
|---|---|---|---|---|---|
| | | VQAv2 | GQA | CIDEr | SPICE |
| Full Model | 7.9B | 76.7 | 49.1 | 123.9 | 15.9 |
| *2:4 Sparsity* | | | | | |
| Wanda | 4.7B | 60.5 | 41.2 | 110.2 | 15.4 |
| w/DSØT | | 64.9 | 43.5 | 107.2 | 14.8 |
| w/SparseLoRA | | **68.3** | **45.4** | **119.3** | **15.5** |
| *4:8 Sparsity* | | | | | |
| Wanda | 4.7B | 63.9 | 43.1 | 116.0 | 15.4 |
| w/DSØT | | 68.3 | 44.8 | 115.2 | 15.1 |
| w/SparseLoRA | | **71.4** | **46.5** | **121.6** | **15.6** |

# 6 Results

## 6.1 Main Experimental Results

**Unstructured Sparsity.** In Table 1, We compare the zero-shot performance on various datasets using BLIP-2 pruned by different pruning techniques at unstructured 50% sparsity ratios. Among all pruning methods, while Wanda and ECoFLaP achieve the best performance, Wanda does not require multiple forward passes and is much more time-efficient. On the other hand, considering, EcoFLaP does not apply for N:M sparsity, we use Wanda as the default pruning method.

Compared to DSØT that focuses on reconstruction errors, SparseLoRA also considers task-specific performance and knowledge distillation from original full models, consistently outperforming the baselines on all tasks. Notably, the average performance of SparseLoRA is comparable to that of the full model.

**N:M Sparsity.** In addition to unstructured sparsity, we also conduct experiments on N: M sparsity (andYukun Ma et al., 2021; Zhang et al., 2022), which can be applied to specific GPU cores and has more practical applications (Mishra et al., 2021). Compared to unstructured pruning, structured pruning causes a more significant performance drop and requires more extensive restores. Under more structured patterns, SparseLoRA recovers more performance, achieving a 10.5% improvement for 2:4 sparsity compared to 3.4% for unstructured sparsity. After restoring, all structured pruned models maintain over 90% of the performance of the original models, demonstrating the universality and effectiveness of SparseLoRA.

**Single Model Pruning.** Language models typically have much larger parameter sizes compared to the vision models in vision-language models, (Li et al., 2023; Dai et al., 2023) (e.g., 7B for Vicuna (Chiang et al., 2023) vs. 1.3B for EVA-ViT in parameters (Sun et al., 2023b)). As a result, the efficiency bottleneck primarily stems from the language model component. This prompted us to investigate the impact of solely pruning language models in VLMs, with experimental results presented in Table 2. With additional parameters in the vision model component, SparseLoRA restores InstructBLIP with significant improvement (e.g., from 69.0 to 71.6 on VQAv2), achieving performance comparable to the Full Model. Therefore, pruning language models only is an effective way to maintain performance and efficiency.

## 6.2 Detailed Analysis

To evaluate cross-modality adaptation, we integrate SparseLoRA into various models. Specifically, we denote the QFormer, vision model, and language model as "$Q$", "$V$", and "$L$" respectively. Different configurations are represented using combina-

Table 3: **Performance comparison at different sparse patterns** (i.e., unstructured 50%, 2:4 and 4:8). using InstructBLIP (Dai et al., 2023) as the backbone. The shown results are the averaged score for 5 runs and the absolute performance gain is denoted as $\Uparrow(\cdot)$.

| | Method | Sparsity | Visual Question Answering | | | Image Captioning | | Macro Avg. |
|---|---|---|---|---|---|---|---|---|
| | | | VQAv2 | OK-VQA | GQA | NoCaps | | |
| | | | | Accuracy | | CIDEr | SPICE | |
| InstructBLIP $\text{FlanT5}_{\text{XL}}$ | Full Model | 0% | 73.5 | 52.6 | 48.4 | 121.4 | 15.6 | 62.3 |
| | Wanda | 50% | 69.1 | 45.4 | 45.7 | 108.7 | 14.2 | 56.6 |
| | w/DSØT | | 68.6 | 45.5 | 45.6 | 107.0 | 14.2 | 56.2 |
| | w/SparseLoRA | | **71.0**$^{\Uparrow+1.9}$ | **48.4**$^{\Uparrow+3.0}$ | **46.7**$^{\Uparrow+1.0}$ | **118.4**$^{\Uparrow+9.7}$ | **15.4**$^{\Uparrow+1.2}$ | **60.0**$^{\Uparrow+3.4}$ |
| | Wanda | 2:4 | 61.2 | 33.9 | 42.1 | 82.5 | 11.9 | 46.3 |
| | w/DSØT | | 63.5 | 35.8 | 42.8 | 96.1 | 13.0 | 50.2 |
| | w/SparseLoRA | | **67.4**$^{\Uparrow+6.2}$ | **43.1**$^{\Uparrow+9.2}$ | **43.8**$^{\Uparrow+1.7}$ | **114.7**$^{\Uparrow+32.2}$ | **14.9**$^{\Uparrow+3.0}$ | **56.8**$^{\Uparrow+10.5}$ |
| | Wanda | 4:8 | 66.0 | 39.8 | 45.1 | 97.1 | 13.1 | 52.2 |
| | w/DSØT | | 67.3 | 41.4 | 46.3 | 105.7 | 13.9 | 54.9 |
| | w/SparseLoRA | | **69.4**$^{\Uparrow+3.4}$ | **45.0**$^{\Uparrow+5.2}$ | **46.9**$^{\Uparrow+1.8}$ | **116.1**$^{\Uparrow+19.0}$ | **15.1**$^{\Uparrow+2.0}$ | **58.5**$^{\Uparrow+6.2}$ |
| InstructBLIP Vicuna-7B | Full Model | 0% | 76.7 | 58.8 | 49.1 | 123.9 | 15.9 | 64.9 |
| | Wanda | 50% | 67.7 | 47.8 | 44.9 | 109.7 | 14.6 | 56.9 |
| | w/DSØT | | 67.5 | 47.6 | 44.8 | 109.3 | 14.6 | 56.8 |
| | w/SparseLoRA | | **72.2**$^{\Uparrow+4.5}$ | **52.0**$^{\Uparrow+4.2}$ | **48.3**$^{\Uparrow+3.4}$ | **118.2**$^{\Uparrow+8.5}$ | **15.1**$^{\Uparrow+0.5}$ | **61.2**$^{\Uparrow+4.3}$ |
| | Wanda | 2:4 | 58.7 | 32.1 | 39.0 | 68.8 | 12.9 | 42.3 |
| | w/DSØT | | 60.2 | 32.3 | 41.4 | 66.9 | 12.6 | 42.7 |
| | w/SparseLoRA | | **66.2**$^{\Uparrow+7.5}$ | **43.6**$^{\Uparrow+11.5}$ | **44.5**$^{\Uparrow+5.5}$ | **112.2**$^{\Uparrow+43.4}$ | **14.6**$^{\Uparrow+1.7}$ | **56.2**$^{\Uparrow+13.9}$ |
| | Wanda | 4:8 | 61.4 | 39.5 | 42.4 | 95.5 | 13.6 | 50.5 |
| | w/DSØT | | 63.3 | 39.6 | 44.6 | 101.1 | 13.9 | 52.5 |
| | w/SparseLoRA | | **69.5**$^{\Uparrow+8.1}$ | **47.4**$^{\Uparrow+7.9}$ | **45.8**$^{\Uparrow+3.4}$ | **115.1**$^{\Uparrow+19.6}$ | **14.9**$^{\Uparrow+1.3}$ | **58.5**$^{\Uparrow+8.0}$ |

Table 4: Comparison between LoRA and SparseLoRA.

| Method | Sparsity | VQAv2 | OK-VQA | GQA |
|---|---|---|---|---|
| Full Model | 0% | 76.7 | 58.8 | 49.1 |
| LoRA | 50% | **74.1** | 52.9 | 48.2 |
| SparseLoRA | | 74.0 | **53.3** | **48.6** |
| LoRA | 2:4 | 67.8 | 43.7 | 44.9 |
| SparseLoRA | | **68.3** | **44.6** | **45.4** |
| LoRA | 4:8 | 70.2 | 48.3 | 45.9 |
| SparseLoRA | | **71.4** | **49.2** | **46.5** |

Table 5: Ablation studies on different finetuning objectives.

| Method | Flickr30k | | NoCaps | |
|---|---|---|---|---|
| | TR@1 | IR@1 | CIDEr | SPICE |
| Full Model | 96.1 | 87.5 | 105.4 | 13.8 |
| $\mathcal{L}_{\text{task}}$ | 95.3 | 86.2 | 106.1 | 14.0 |
| $\mathcal{L}_{\text{distill}}$ | 95.4 | 86.6 | 102.2 | 13.4 |
| $\mathcal{L}_{\text{task}}\&\mathcal{L}_{\text{distill}}$ | **96.0** | **87.2** | **106.6** | **14.1** |

tions of these notations (e.g., "$QLV$" and "$LV$"). As shown in Table 6, for cross-modality pruning (i.e., Vision + Language), finetuning within a single model contributes to performance restoration, while finetuning models across two modalities further enhances performance. In cases of single modality pruning, finetuning the pruned model alone is sufficient for restoration. Notably, joint finetuning with the QFormer does not yield performance gains beyond finetuning the pruned models.

**SparseLoRA Finetuning Achieves Comparable Performance with LoRA.** LoRA weights cannot be merged with pruned weights, as this would disrupt the sparse pattern. Consequently, the presence of remaining LoRA modules leads to latency and slows down inference significantly (Dery et al., 2024; Rücklé et al., 2021). To address this issue, SparseLoRA aims to resolve the unmerged weights of LoRA and eliminate the latency caused by LoRA modules. Table 4 compares the performance of LoRA with SparseLoRA for VLMs with sparse language models. Remarkably, SparseLoRA finetuning achieves improved performance with fewer trainable parameters, consistent with findings from (He et al., 2022).

**The Effectiveness of Finetuning Objectives.** We further investigate the impact of the proposed finetuning objectives on BLIP-2-$\text{FlanT5}_{\text{XL}}$. In Table 5, we consider three finetuning objectives: $\mathcal{L}_{\text{task}}$, $\mathcal{L}_{\text{distill}}$, and $\mathcal{L}_{\text{task}}\&\mathcal{L}_{\text{distill}}$. $\mathcal{L}_{\text{task}}$ guides the task-specific performance while $\mathcal{L}_{\text{distill}}$ guides knowledge transferring from the original full model to the pruned dense model. either minimizing $\mathcal{L}_{\text{task}}$ or $\mathcal{L}_{\text{distill}}$ improves the performance. In addi-

Table 6: **Performance of SparseLoRA applied on pruning scenarios**, where “Vision + Language” denotes pruning both vision models and language models, and “Language” denotes pruning language models only. $V$, $L$, $Q$ represent the models for SparseLoRA.

| Method | Modality | Vision + Language | | | | Language | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | VQAv2 | OK-VQA | GQA | Avg. | VQAv2 | OK-VQA | GQA | Avg. |
| Wanda | – | 61.4 | 39.5 | 42.4 | 47.8 | 63.9 | 44.5 | 43.1 | 50.5 |
| w/SparseLoRA | $V$ | 64.3 | 42.6 | 45.8 | 51.0 | 66.0 | 44.7 | 43.3 | 51.3 |
| | $L$ | 66.4 | 46.7 | 44.3 | 52.5 | **70.8** | **46.5** | **49.5** | **55.6** |
| | $Q$ | 62.5 | 40.2 | 43.3 | 48.7 | 64.3 | 44.5 | 44.1 | 51.0 |
| | $V+L$ | **69.5** | **47.4** | **45.8** | **54.2** | 70.6 | 46.3 | 49.3 | 55.4 |
| | $V+L+Q$ | 69.0 | 46.9 | 45.4 | 53.8 | 70.2 | 45.9 | 48.1 | 54.7 |

Figure 6: Ablation study on sparsity ratios.

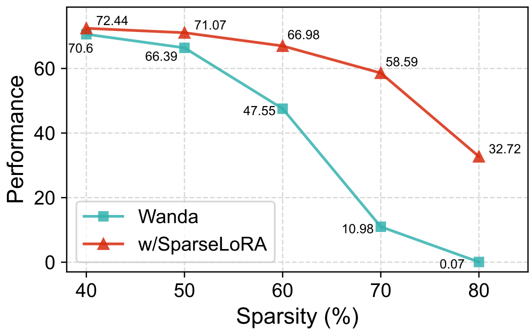
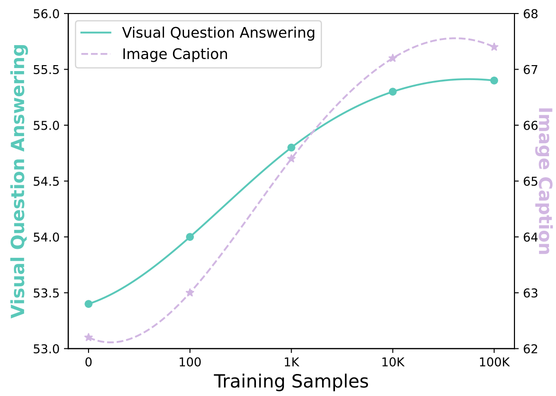



Figure 7: Impact of finetuning samples.




tion, jointly minimizing $\mathcal{L}_{\text{distill}}$ and $\mathcal{L}_{\text{task}}$ helps the pruned models further recover performance.

**Ablation Study on Sparsity.** To assess the effectiveness of SparseLoRA across a broader range of sparsity ratios, we experimented on InstructBLIP-Vicuna-7B with unstructured sparsity ratios ranging from 40% to 80%. As the sparse ratio $s$ exceeded 50%, the performance of pruned models began to deteriorate, eventually collapsing when $s \geq 70\%$, highlighting the necessity of restoring. In such scenarios, SparseLoRA significantly improved performance, particularly for higher sparsity ratios, achieving a recovery of 47.6% of scores at $s = 70\%$ and 32.7% at $s = 80\%$.

**Ablation Study on Calibration Datasets.** SparseLoRA utilizes calibration datasets for retraining. We conducted experiments to explore the impact of the number of training samples. Specifically, we randomly sampled $k$ ($k$ = 0, 100, 1k, 10k, 100k) training data points from CC3M (Sharma et al., 2018) to finetune InstructBLIP-FlanT5$_{\text{XL}}$ with 50% sparsity and report the average performance of visual question answering and image caption. As shown in Figure 7, we found that finetuning pruned VLMs with few-shot samples (i.e., 100) can improve performance by a substantial margin. Further finetuning with 10k training data points resulted in a significant boost in cross-modality ability. This suggests that a small amount of data is sufficient to restoration the pruned vision-language models, leveraging the knowledge and capabilities acquired during pretraining (Zhou et al., 2023). When $k \geq$ 10k, the model’s capability continues to improve with more training data and gradually becomes saturated.

## 7 Conclusion

In this paper, motivated by the challenges associated with deploying VLMs in real-world applications, we investigate the potential of pruning VLMs. Specifically, recognizing that VLMs encompass models from different modalities, we conduct empirical studies to explore the distribution of sparsity ratios across these models and how sparsity impacts performance, thereby highlighting the necessity of restoring pruned VLMs. Subsequently, we introduce MAF, which addresses this challenge by restoring pruned VLMs through cross-modality adaptation and SparseLoRA finetuning. Extensive experiments validate the effectiveness of MAF, providing valuable insights for future research on VLM sparsity.

## 8 Limitations

Despite our progress, limitations remain in our work. Although our proposed methods are universal for all VLM models, we have primarily focused on BLIP family models and selected tasks . We believe our methods can be easily extended to a broader range of models and tasks. On the other hand, given there may be potentially high-quality dataset for restoring pruned models, we believe the incorporation of such datasets would further promotes our proposed methods

## References


Harsh Agrawal, Karan Desai, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, Stefan Lee, and Peter Anderson. 2019. nocaps: novel object captioning at scale. *International Conference on Computer Vision*, pages 8947–8956.

Jose M Alvarez and Mathieu Salzmann. 2016. Learning the number of neurons in deep networks. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Aojun Zhou andYukun Ma, Junnan Zhu, Jianbo Liu, Zhijie Zhang, Kun Yuan, Wenxiu Sun, and Hongsheng Li. 2021. Learning n:m fine-grained structured sparse neural networks from scratch. In *International Conference on Learning Representations*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Tianlong Chen, Jonathan Frankle, Shiyu Chang, Sijia Liu, Yang Zhang, Zhangyang Wang, and Michael Carbin. 2020. The lottery ticket hypothesis for pretrained bert networks. *Preprint*, arXiv:2007.12223.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Hyung Won Chung, Le Hou, S. Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Wei Yu, Vincent Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed Huai hsin Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *ArXiv*, abs/2210.11416.

Bin Dai, Chen Zhu, Baining Guo, and David Wipf. 2018. Compressing neural networks using the variational information bottleneck. In *Proceedings of the 35th International Conference on Machine Learning (ICML 2018)*.

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

Lucio Dery, Steven Kolawole, Jean-François Kagy, Virginia Smith, Graham Neubig, and Ameet Talwalkar. 2024. Everybody prune now: Structured pruning of llms with only forward passes. *Preprint*, arXiv:2402.05406.

Gongfan Fang, Xinyin Ma, Mingli Song, Michael Bi Mi, and Xinchao Wang. 2023. Depgraph: Towards any structural pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16091–16101.

Jonathan Frankle and Michael Carbin. 2019. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *Preprint*, arXiv:1803.03635.

Elias Frantar and Dan Alistarh. 2023. SparseGPT: Massive language models can be accurately pruned in one-shot. *arXiv preprint arXiv:2301.00774*.

Ian J. Goodfellow, Mehdi Mirza, Xia Da, Aaron C. Courville, and Yoshua Bengio. 2014. An empirical investigation of catastrophic forgeting in gradient-based neural networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2016. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. *International Journal of Computer Vision*, 127:398 – 414.

Song Han, Huizi Mao, and William J. Dally. 2016. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *Preprint*, arXiv:1510.00149.

Shwai He, Liang Ding, Daize Dong, Jeremy Zhang, and Dacheng Tao. 2022. SparseAdapter: An easy approach for improving the parameter-efficiency of adapters. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2184–2190, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Shwai He, Chenbo Jiang, Daize Dong, and Liang Ding. 2023. Sd-conv: Towards the parameter-efficiency of dynamic convolution. *Preprint*, arXiv:2204.02227.

Yang He and Lingao Xiao. 2024. Structured pruning for deep convolutional neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 1–20.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *Preprint*, arXiv:1503.02531.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. *Preprint*, arXiv:1902.00751.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Drew A. Hudson and Christopher D. Manning. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6693–6702.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts. *Preprint*, arXiv:2401.04088.

Jin-Hwa Kim, Sang-Woo Lee, Donghyun Kwak, Min-Oh Heo, Jeonghee Kim, Jung-Woo Ha, and Byoung-Tak Zhang. 2016. Multimodal residual learning for visual qa. In *Neural Information Processing Systems*.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Aditya Kusupati, Vivek Ramanujan, Raghav Somani, Mitchell Wortsman, Prateek Jain, Sham Kakade, and Ali Farhadi. 2020. Soft threshold weight reparameterization for learnable sparsity. *Preprint*, arXiv:2002.03231.

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip Torr. 2019. SNIP: SINGLE-SHOT NETWORK PRUNING BASED ON CONNECTION SENSITIVITY. In *International Conference on Learning Representations*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Liyang Liu, Shilong Zhang, Zhanghui Kuang, Aojun Zhou, Jing-Hao Xue, Xinjiang Wang, Yimin Chen, Wenming Yang, Qingmin Liao, and Wayne Zhang. 2021. Group fisher pruning for practical network compression. *Preprint*, arXiv:2108.00708.

Yixin Liu, Kai Zhang, Yuan Li, Zhiling Yan, Chujie Gao, Ruoxi Chen, Zhengqing Yuan, Yue Huang, Hanchi Sun, Jianfeng Gao, Lifang He, and Lichao Sun. 2024. Sora: A review on background, technology, limitations, and opportunities of large vision models. *Preprint*, arXiv:2402.17177.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. Llm-pruner: On the structural pruning of large language models. In *Advances in Neural Information Processing Systems*.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`.

Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. 2019. Ok-vqa: A visual question answering benchmark requiring external knowledge. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3190–3199.

Asit Mishra, Jorge Albericio Latorre, Jeff Pool, Darko Stosic, Dusan Stosic, Ganesh Venkatesh, Chong Yu, and Paulius Micikevicius. 2021. Accelerating sparse deep neural networks. *arXiv preprint arXiv:2104.08378*.

Nandini Mundra, Sumanth Doddapaneni, Raj Dabre, Anoop Kunchukuttan, Ratish Puduppully, and Mitesh M. Khapra. 2023. A comprehensive analysis of adapter efficiency. *Preprint*, arXiv:2305.07491.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Bryan A. Plummer, Liwei Wang, Christopher M. Cervantes, Juan C. Caicedo, J. Hockenmaier, and Svetlana Lazebnik. 2015. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. *International Journal of Computer Vision*, 123:74 – 93.

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. 2021. Learning transferable visual models from natural language supervision. *Preprint*, arXiv:2103.00020.

Andreas Rücklé, Gregor Geigle, Max Glockner, Tilman Beck, Jonas Pfeiffer, Nils Reimers, and Iryna Gurevych. 2021. Adapterdrop: On the efficiency of adapters in transformers. *Preprint*, arXiv:2010.11918.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. 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, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 2556–2565. Association for Computational Linguistics.

Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. 2022. Make-a-video: Text-to-video generation without text-video data. *Preprint*, arXiv:2209.14792.

Samuel Stanton, Pavel Izmailov, Polina Kirichenko, Alexander A. Alemi, and Andrew Gordon Wilson. 2021. Does knowledge distillation really work? *Preprint*, arXiv:2106.05945.

Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. 2023a. A simple and effective pruning approach for large language models. *arXiv preprint arXiv:2306.11695*.

Quan Sun, Yuxin Fang, Ledell Yu Wu, Xinlong Wang, and Yue Cao. 2023b. Eva-clip: Improved training techniques for clip at scale. *ArXiv*, abs/2303.15389.

Hidenori Tanaka, Daniel Kunin, Daniel L. K. Yamins, and Surya Ganguli. 2020. Pruning neural networks without any data by iteratively conserving synaptic flow. *Preprint*, arXiv:2006.05467.

Ziheng Wang, Jeremy Wohlwend, and Tao Lei. 2020. Structured pruning of large language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6151–6162, Online. Association for Computational Linguistics.

Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Yumao Lu, Zicheng Liu, and Lijuan Wang. 2022. An empirical study of gpt-3 for few-shot knowledge-based vqa. In *AAAI*.

Mohit Bansal Yi-Lin Sung, Jaehong Yoon. 2024. Ecoflap: Efficient coarse-to-fine layer-wise pruning for vision-language models. In *International Conference on Learning Representations (ICLR)*.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. 2023. A survey on multimodal large language models. *Preprint*, arXiv:2306.13549.

Yuxin Zhang, Mingbao Lin, ZhiHang Lin, Yiting Luo, Ke Li, Fei Chao, YONGJIAN WU, and Rongrong Ji. 2022. Learning best combination for efficient n:m sparsity. In *Advances in Neural Information Processing Systems*.

Yuxin Zhang, Lirui Zhao, Mingbao Lin, Yunyun Sun, Yiwu Yao, Xingjia Han, Jared Tanner, Shiwei Liu, and Rongrong Ji. 2024. Dynamic sparse no training: Training-free fine-tuning for sparse llms. *Preprint*, arXiv:2310.08915.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. Lima: Less is more for alignment. *Preprint*, arXiv:2305.11206.

Yufan Zhou, Ruiyi Zhang, Changyou Chen, Chunyuan Li, Chris Tensmeyer, Tong Yu, Jiuxiang Gu, Jinhui Xu, and Tong Sun. 2021. Lafite: Towards language-free training for text-to-image generation. *arXiv preprint arXiv:2111.13792*.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *Preprint*, arXiv:2304.10592.