UPop: Unified and Progressive Pruning for Compressing Vision-Language Transformers

Anonymous authors
Paper under double-blind review

Abstract

Data from the real world contains a vast amount of multimodal information, among which vision and language are the two most representative modalities. On the other hand, researchers have spent much effort on model compression to reduce the huge memory and computational consumption of increasingly large models. However, how to compress multimodal models, especially vision-language Transformers, is still under-explored. This paper proposes the Unified and Progressive Pruning (UPop) that compresses vision-language Transformers via pruning. UPop incorporates 1) unifiedly searching countless multimodal subnetworks in a continuous optimization space from the uncompressed model; 2) progressively and simultaneously retraining the subnetwork. The subnetworks are learned in multiple components, including the self-attention modules, MLPs in both vision and language branches, and cross-attention modules. To ease the progress of pruning, we design Unified Pruning to automatically assign the optimal pruning ratio to each compressable component, instead of manually assigning each component a pruning ratio. To explore the limitation of compression ratio, we propose Progressive Pruning to maintain convergence between search and retrain. In addition, UPop enables zero-cost subnetwork selection after searching countless multimodal subnetworks, and the searched subnetwork can be used without any retraining. Experiments on multiple discriminative and generative vision-language tasks demonstrate the versatility of the proposed UPop. For example, we achieve $2 \times$ compression and $1.66 \times$ FLOPs reduction on COCO dataset of Image Caption with $0.8$ SPICE drop, $4 \times$ compression and $2.96 \times$ FLOPs reduction with $2.1$ SPICE drop.

1 Introduction

The number of parameters and FLOPs of deep learning models [Devlin et al., 2018; Shoeybi et al., 2019; Brown et al., 2020; Shao et al., 2021; Smith et al., 2022] have proliferated in recent years, which makes model compression greatly critical for deploying the bigger and bigger models on edge devices. There are lots of approaches can be used to compress deep learning models, such as weight sharing (Lan et al., 2019), low-rank factorization (Yu et al., 2017), pruning (He et al., 2017), quantization (Wang et al., 2019), and knowledge distillation (Yang et al., 2022).

After the popularity of the Transformer (Vaswani et al., 2017), compression approaches dedicated to the Transformers have also attracted much attention. Depending on the compressed components, these approaches can be divided into two categories. The first category is token compression. By eliminating the number of input tokens, these approaches (Goyal et al., 2020; Rao et al., 2021) can significantly reduce the FLOPs of models. The second category is model compression. By reducing the embedding size, these approaches (Wang et al., 2020; 2021) can reduce both the parameters and FLOPs of models. This paper focuses on model compression. Therefore the parameters and FLOPs of models can be reduced simultaneously.

In real life, there are prevalent situations where humans need to receive and process information from multiple modalities, among which vision and language are the two most representative ones. There are lots of multimodal tasks that have been extensively studied, including but not limited to Image Caption (Lin et al., 2014) that requires generating a text description for a given image, Text-Image Retrieval (Jia et al., 2015) that requires selecting one image from the candidate list based
Under review as a conference paper at ICLR 2023

On a given text description, and NLVR\(^2\) (Suhr et al., 2018) that requires predicting whether a given sentence correctly describes a pair of given images.

To tackle these multimodal tasks, various multimodal models (Kiros et al., 2014; Karpathy et al., 2014; Antol et al., 2015; Vinyals et al., 2015; Yang et al., 2016; Huang et al., 2017) have been proposed accordingly. As Transformer (Vaswani et al., 2017) has been more and more popular among deep models, transformer-based models (Tan & Bansal, 2019; Lu et al., 2019; Zhou et al., 2020; Li et al., 2020; Kim et al., 2021; Jia et al., 2021; Yu et al., 2022; Wang et al., 2022a) have also dominated the recent studies of multimodal models. For example, CLIP (Radford et al., 2021) and BLIP (Li et al., 2022) are some of the most representative multimodal models among them. Benefiting from massive image-text pairs as pre-training datasets, they can learn joint representations of multiple modalities. They can be further used to fine-tune on kinds of multimodal tasks.

Although unimodal compression has been widely investigated, how to compress multimodal models, especially vision-language Transformers, is still under-explored. Only a few works (Jin et al., 2021; Fang et al., 2021; Wang et al., 2022b) have paid attention to this problem, and all of them have been trying to conduct compression from the perspective of knowledge distillation. This paper explores a new path that conducts compression for multimodal models. Inspired by the unimodal compression approach ViT-Slimming (Chavan et al., 2022), we propose a novel multimodal compression approach, Unified and Progressive Pruning (UPop).

A straightforward design of multimodal compression is to compress each modality separately via the unimodal compression approach. However, there exist two main challenges. One of the challenges is that we have to manually assign compression ratios for different components in both vision and language branches, which is inefficient and resource-consuming. To overcome this shortcoming, we propose to unify search on different modalities. Furthermore, we explore the feasible approach to unified search on heterogeneous structures. The unified search on different modalities and heterogeneous structures enables our approach to assign compression ratios among all compressible components adaptively. The second challenge is that the traditional compression diagram (i.e. retrain after obtaining pruned subnetworks) fails when it comes to large models or high compression ratios. The significant gap of parameter weights between the searched model and the model to be retraining severely degrades the model performance and even causes it hard to converge. Consequently, we propose an improved compression diagram that conducts search and retrain progressively and simultaneously, which can effectively eliminate the gap mentioned above.

In summary, our main contributions can be summarized as:

- For the first time, we propose a multimodal compression approach UPop for vision-language Transformers from the perspective of pruning. UPop searches multimodal models in continuous optimization space, and a round of search can yield countless multimodal subnetworks. (Section 3.1 & Section 4)
- We propose Unified Pruning, which unifiedly searches for all compressible components from heterogeneous structures of multimodal models, which enables adaptive compression ratio assignment among all compressible components. (Section 3.2 & Section 4.3)
- We propose Progressive Pruning, which conducts progressive and simultaneous searching and retraining as an improved compression diagram. Progressive Pruning effectively eliminates the gap in parameter weights between the searched model and the model to be retrained. (Section 3.3 & Section 4.4)

2 RELATED WORK

Multimodal Transformers We take multimodal models fine-tuned from the pretrained BLIP model (Li et al., 2022) as original models to be compressed. BLIP is one of the most representative SOTA multimodal models and has surpassed a series of derivative models (Lei et al., 2021; Xu et al., 2021) of CLIP (Radford et al., 2021) on various multimodal tasks. BLIP is a pure transformed-based multimodal model, which employs a Bert (Devlin et al., 2018) and a ViT (Dosovitskiy et al., 2020) as the text encoder and the image encoder, respectively. To allow interaction between vision and language modalities, BLIP injects vision information from the image encoder to the text encoder by inserting an additional cross-attention layer after the self-attention layer of each transformer block in the text encoder. BLIP is pre-trained via three objectives: Image-Text Contrastive loss, which
Model Compression via Pruning  ViT-Slimming (Chavan et al., 2022) is a pruning approach for unimodal Transformers compression. It can be used to compress Multi-Head Self-Attention (MHSA) and Multi-Layer Perceptron (MLP) in unimodal Transformers. ViT-Slimming consists of two phases, search and retraining. In the search phase, trainable masks are inserted together into the MHSA and MLP modules for training. After the search phase, the model to be retrained is selected from the searched model according to the magnitude of masks. In the retraining phase, the selected model will be retrained and finally output the compressed model.

3 METHODOLOGY

This section will illustrate how Unified and Progressive Pruning is built. Necessary notations and their corresponding descriptions are listed in Appendix A.

3.1 SIMPLE PRUNING

A straightforward design for multimodal compression is to compress each modality separately via the unimodal compression approach. We adapt the strategy that searches in continuous optimization space to the multimodal models. Consequently, a round of searches can yield countless multimodal subnetworks. After the search, we can conduct retraining on the searched model without extra search costs. Specifically, We can compress ViT used for vision modality of multimodal model BLIP by strategy above. Furthermore, Bert used for language modality and cross-attention of BLIP can also be compressed by simply adapting the compression approach of ViT to them.

The detailed implementation of this approach is outlined in the Algorithm 2 of Appendix B. One of the shortcomings of this approach is that each $\zeta$ is searched individually, which causes the compression ratio for different components to have to be manually assigned. Since different modalities and structures may have different importance, and the optimal assignment for different models may vary, the Algorithm 2 is inefficient and resource-consuming.

3.2 UNIFIED PRUNING

To overcome the shortcoming above, we propose an improved Unified Pruning approach as outlined in the Algorithm 3 of Appendix C. This improvement enables the Algorithm 3 to unifiedly search on different modalities and heterogeneous structures.

Unified Search on Different Modalities  Based on the Algorithm 2, we firstly group the mask $\zeta$ used for homogeneous structures together. For models fined-tuned from BLIP, there are two types of structures, MHSA and MLP. And the grouping results are

$$\zeta_a = \{\zeta_{v_{att}}, \zeta_{l_{att}}, \zeta_{c_{att}}\}, \quad \zeta_m = \{\zeta_{v_{mlp}}, \zeta_{l_{mlp}}\}$$ (1)

Instead of searching on each $\zeta_i \in \zeta$ individually as line 5 ~ 6 of the Algorithm 2 shows, search on different modalities of homogeneous structures would be a better approach:

$$\mathcal{M}_a \leftarrow \text{TopKMask}\{\{\zeta_{i(T_s)}|\zeta_i \in \zeta_a\}, p \cdot \text{Size}(\zeta_a)\}$$ (2)

$$\mathcal{M}_m \leftarrow \text{TopKMask}\{\{\zeta_{i(T_s)}|\zeta_i \in \zeta_m\}, p \cdot \text{Size}(\zeta_m)\}$$ (3)

This unified search on different modalities gives Algorithm 3 the ability to assign compression ratio of different modalities adaptively.
Unified Search on Heterogeneous Structures A unified search can also be conducted on heterogeneous structures. Simply uniting heterogeneous structures causes performance degradation, and the reason why simple union fails is that the magnitude distributions of masks used for heterogeneous structures vary greatly.

However, it is feasible to conduct unified searching after transforming the magnitudes of heterogeneous structures’ masks to the same distribution. For the simplicity of implementation, we individually transform the magnitudes to the standard normal distribution.

\[
\zeta_a(T_s) \leftarrow \frac{\zeta_a(T_s) - E[\zeta_a(T_s)]}{\sqrt{E[(\zeta_a(T_s) - E[\zeta_a(T_s)])^2]}} \quad \zeta_m(T_s) \leftarrow \frac{\zeta_m(T_s) - E[\zeta_m(T_s)]}{\sqrt{E[(\zeta_m(T_s) - E[\zeta_m(T_s)])^2]}}
\]

Then the Algorithm 3 can search for heterogeneous structures of different modalities:

\[
\mathcal{M} \leftarrow \text{TopKMask}(\zeta(T_s), \, p \cdot \text{Size}(\zeta))
\]

and thus can unified assign compression ratio among all compressible components.

3.3 Progressive Pruning

Retrain after searching is a traditional diagram for model compression. However, this diagram is unsuitable for compressing large models like multimodal Transformers, especially when the compression ratio is high. The reason is that there is a significant gap between the parameter weights of the searched model \(\Theta(T_s)\) and the model \(\hat{\Theta}\) to be retrained.

Take the MLP of ViT as an example. When the search phase is complete, the intermediate tensor of \(l^{th}\) MLP is

\[
f_1^e(m_i^T) \odot e^m_{mlp} \in \mathbb{R}^{N^\times M^e}
\]

Suppose its assigned compression ratio is \(p_l \leq 1\), then the intermediate tensor of model to be retrained has the shape \(\mathbb{R}^{N^\times (p_l \times M^e)}\). Notice that there is no guarantee that the magnitude of the mask corresponding to the eliminated positions will converge to 0, which means the operation that selects \(\hat{\Theta}\) from \(\Theta(T_s)\) is a mutation process. When the model has more parameters or given a higher compression ratio, this mutation is more drastic and it is more difficult for selected model \(\mathcal{F}(x|\hat{\Theta}, \zeta(T_s))\) to converge.

Paradigm of Progressive Pruning To address the above issue, we further propose the Algorithm 4 based on the Algorithm 3. Eliminating the abovementioned gap is equivalent to ensuring that each magnitude of the mask corresponding to the eliminated positions converges to 0. This can be achieved by separately updating \(\zeta\) with a function of the current iteration number \(t\), instead of updating \(\zeta\) with the same optimizer as \(\Theta\).

Specifically, we firstly collect gradients \(G^{(i)}\) of \(\zeta\) as shown in the Line 5 ~ 6 of the Algorithm 4. And generate compression mask \(\mathcal{M}^t\) according to the magnitude of the accumulated gradients \(\sum_{i=0}^t G^{(i)}\). The update strategy for \(\zeta\) can be written as

\[
\zeta_i^{(t+1)} \leftarrow \mathcal{M}_i^t + (1 - \mathcal{M}_i^t)(1 - \frac{p_t}{p}), \quad \zeta_i \in \zeta
\]

where \(p_t\) is the current compression ratio when the iteration number is \(t\). This strategy ensures that as \(p_t\) progressively increases to the final compression ratio \(p\), the masks’ magnitude of all positions to be eliminated will exactly converge to 0.

Besides, this approach achieves a new paradigm that conducts searching and retraining simultaneously. For example, when the iteration number is \(t + 1\) of the search phase, the model is compressed from \(\mathcal{F}_{p_t}\) to \(\mathcal{F}_{p_{t+1}}\). Then during the next iteration \(t + 2\) of the search phase, what the approach actually doing is retraining the \(p_{t+1}\%\)-compressed model \(\mathcal{F}_{p_{t+1}}\), meanwhile \(\mathcal{F}_{p_{t+1}}\) is searched and further be compressed to \(\mathcal{F}_{p_{t+2}}\).
Growth in Compression Ratio  The last remaining issue is how to properly update the current compression ratio $p_t$ as iteration number $t$ increases. On the one hand, the actual compression ratio should increase relatively slowly at the beginning of searching. Because when the iteration number $t$ is small, the cumulative gradients are relatively volatile, and the generated mask is relatively inaccurate. On the other hand, the actual compression ratio should also increase relatively slowly near the end. Because as the compression ratio gradually increases, the difficulty of compression also increases. We illustrate that the $p_t$ used in the Algorithm 1 satisfies the aforementioned requirements. The detailed derivation process is in Appendix D.

4 EXPERIMENTS

4.1 Multimodal Models and Tasks

We can get a diversified set of multimodal models by fine-tuning BLIP on various multimodal tasks. From them, we select three representative tasks, the NLVR², Image Caption, and Image-Text Retrieval, as well as their corresponding fine-tuned models, to conduct our experiments. In this section, the experiments conducted on the NLVR² will be firstly reported. Then based on the experimental results, a detailed analysis of the proposed approach will be provided. Finally, the extended experiments on the Image Caption and Image-Text Retrieval are reported to verify the versatility of the proposed approach.

4.2 Compression Results on the NLVR²

NLVR² is a binary classification task with two images and a text description as inputs. To quantitatively evaluate the proposed UPop, we compress the BLIP model fine-tuned on this task at a ratio of $2, 3, 4, 5$ and $10$ times, respectively. The model consists of two weight-shared ViT as image encoder and a Bert with two cross-attention as text encoder, therefore the mask $\zeta$ corresponding to the compressible components on this model is $\zeta = \{\zeta_{\text{att}}, \zeta_{\text{mlp}}, \zeta_{\text{att}}, \zeta_{\text{mlp}}, \zeta_{\text{att}}, \zeta_{\text{att}}\}$. We compress the original model with three aforementioned multimodal compression approaches, Simple Pruning (Algorithm 2), Unified Pruning (Algorithm 3), and UPop (Algorithm 1), respectively. Experimental results are shown in Table 1. It is worth noting that at a compression ratio of $N$ times, the total number of parameters of the compressed model will not be strictly equal to the $\frac{1}{N}$ of the original model. This is because some modules of the original model are not covered by the mask $\zeta$, such as

**Algorithm 1 UPop: Unified and Progressive Pruning**

| Input: $\zeta, \zeta_0, \zeta_m, \Theta, F, p, T_s, T_r, \alpha, \beta$ |
| for $t \leftarrow 0$ to $T_r - 1$ do |
| if $t < T_s$ then |
| $L \leftarrow L_0 + w_a \sum_{\zeta_i \in \zeta} \| \zeta_i \|_1 + w_m \sum_{\zeta_i \in \zeta_m} \| \zeta_i \|_1$ |
| $\Theta^{(t+1)} \leftarrow -\Theta^{(t)} - \alpha \frac{1}{n} \sum_{i=1}^{n} \nabla_{\Theta} L(\Theta^{(t)}, \zeta^{(t)})$ |
| $G^{(t)} \leftarrow \frac{1}{n} \sum_{i=1}^{n} \nabla_{\Theta} L(\Theta^{(t)}, \zeta^{(t)})$ |
| $G_\alpha^{(t)} \leftarrow \frac{\sqrt{E[G^{(t)} - E[G_\alpha^{(t)}]^2}}}{\sqrt{E[G^{(t)} - E[G_\alpha^{(t)}]^2]}}$, $G_{\theta,m}^{(t)} \leftarrow \frac{\sqrt{E[G^{(t)} - E[G_{\theta,m}^{(t)}]^2}}}{\sqrt{E[G^{(t)} - E[G_{\theta,m}^{(t)}]^2]}}$ |
| $p_t = p \sqrt{\left(1 - \cos\left(\frac{\pi t}{T_s - 1}\right)\right)^{\frac{1}{2}}}$ |
| $M_t \leftarrow \text{TopKMask}(\sum_{i=0}^{t} G^{(i)}, p_t \cdot \text{Size}(\zeta))$ |
| for $\zeta_i \in \zeta$ do |
| $\zeta_i^{(t+1)} \leftarrow M_t + (1 - M_t) (1 - \frac{p_t}{p})$ |
| $F_{p_{t+1}} \leftarrow F_p(x|\Theta^{(t+1)}, \zeta^{(t+1)})$ |
| else |
| $\Theta^{(t+1)} \leftarrow -\Theta^{(t)} - \beta \frac{1}{n} \sum_{i=1}^{n} \nabla_{\Theta} L(\Theta^{(t)})$ |
| $F^{\ast} \leftarrow F_p(x|\Theta^{(T_r)})$ |
| return $F^{\ast}$ |

Growth in Compression Ratio  The last remaining issue is how to properly update the current compression ratio $p_t$ as iteration number $t$ increases. On the one hand, the actual compression ratio should increase relatively slowly at the beginning of searching. Because when the iteration number $t$ is small, the cumulative gradients are relatively volatile, and the generated mask is relatively inaccurate. On the other hand, the actual compression ratio should also increase relatively slowly near the end. Because as the compression ratio gradually increases, the difficulty of compression also increases. We illustrate that the $p_t$ used in the Algorithm 1 satisfies the aforementioned requirements. The detailed derivation process is in Appendix D.

4 EXPERIMENTS

4.1 Multimodal Models and Tasks

We can get a diversified set of multimodal models by fine-tuning BLIP on various multimodal tasks. From them, we select three representative tasks, the NLVR², Image Caption, and Image-Text Retrieval, as well as their corresponding fine-tuned models, to conduct our experiments. In this section, the experiments conducted on the NLVR² will be firstly reported. Then based on the experimental results, a detailed analysis of the proposed approach will be provided. Finally, the extended experiments on the Image Caption and Image-Text Retrieval are reported to verify the versatility of the proposed approach.

4.2 Compression Results on the NLVR²

NLVR² is a binary classification task with two images and a text description as inputs. To quantitatively evaluate the proposed UPop, we compress the BLIP model fine-tuned on this task at a ratio of $2, 3, 4, 5$ and $10$ times, respectively. The model consists of two weight-shared ViT as image encoder and a Bert with two cross-attention as text encoder, therefore the mask $\zeta$ corresponding to the compressible components on this model is $\zeta = \{\zeta_{\text{att}}, \zeta_{\text{mlp}}, \zeta_{\text{att}}, \zeta_{\text{mlp}}, \zeta_{\text{att}}, \zeta_{\text{att}}\}$. We compress the original model with three aforementioned multimodal compression approaches, Simple Pruning (Algorithm 2), Unified Pruning (Algorithm 3), and UPop (Algorithm 1), respectively. Experimental results are shown in Table 1. It is worth noting that at a compression ratio of $N$ times, the total number of parameters of the compressed model will not be strictly equal to the $\frac{1}{N}$ of the original model. This is because some modules of the original model are not covered by the mask $\zeta$, such as
Table 1: Compression results on the NLVR$^2$. The column Reduce indicates the compression times of the model. The column Status indicates whether the model converges.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Reduce</th>
<th>Status</th>
<th>Dev Acc(↑)</th>
<th>Test-p Acc(↑)</th>
<th>Params(M)</th>
<th>FLOPs(G)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original Model</td>
<td>1×</td>
<td>✓</td>
<td>82.48</td>
<td>83.08</td>
<td>259.45</td>
<td>132.54</td>
</tr>
<tr>
<td>Simple Pruning (Algorithm 2)</td>
<td>2×</td>
<td>✓</td>
<td>75.74</td>
<td>76.44</td>
<td>146.18</td>
<td>66.88</td>
</tr>
<tr>
<td></td>
<td>3×</td>
<td>✗</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>4×</td>
<td>✗</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>5×</td>
<td>✗</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>10×</td>
<td>✗</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Unified Pruning (Algorithm 3)</td>
<td>2×</td>
<td>✓</td>
<td>79.50</td>
<td>80.32</td>
<td>149.90</td>
<td>95.01</td>
</tr>
<tr>
<td></td>
<td>3×</td>
<td>✓</td>
<td>71.25</td>
<td>71.66</td>
<td>106.33</td>
<td>68.19</td>
</tr>
<tr>
<td></td>
<td>4×</td>
<td>✗</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>5×</td>
<td>✗</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>10×</td>
<td>✗</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Unified and Progressive Pruning (Algorithm 1)</td>
<td>2×</td>
<td>✓</td>
<td>80.33</td>
<td>81.13</td>
<td>150.15</td>
<td>89.36</td>
</tr>
<tr>
<td></td>
<td>3×</td>
<td>✓</td>
<td>76.89</td>
<td>77.61</td>
<td>109.01</td>
<td>65.29</td>
</tr>
<tr>
<td></td>
<td>4×</td>
<td>✓</td>
<td>72.85</td>
<td>73.55</td>
<td>88.61</td>
<td>50.35</td>
</tr>
<tr>
<td></td>
<td>5×</td>
<td>✓</td>
<td>68.71</td>
<td>68.76</td>
<td>76.81</td>
<td>39.93</td>
</tr>
<tr>
<td></td>
<td>10×</td>
<td>✓</td>
<td>57.17</td>
<td>57.79</td>
<td>54.48</td>
<td>19.08</td>
</tr>
</tbody>
</table>

the patch embedding module of the image encoder, the word embedding module of the text encoder, and the classification head. In addition, at the same compression ratio, different search masks will also lead to different structures and FLOPs of the compressed model.

4.3 Effect of Unified Pruning

At the 2× compression ratio, the Table 1 shows that compared to the Simple Pruning (Algorithm 2), the Unified Pruning (Algorithm 3) gains 3.76% and 3.88% accuracy improvement on the dev set and test-p set, respectively. Furthermore, at the 3× compression ratio, the Unified Pruning converges successfully while the Simple Pruning does not.

As shown in Figure 1, Unified Pruning enables the model to adaptively assign a proper compression ratio among different components. The figure shows that each compressible component has significantly different compression trends and unbalanced compression assignments in different layers at different compression ratios.

For the convenience of analysis, Figure 2 of Appendix F demonstrates the variation of all components and layers as the total compression ratio increases. The left subfigure shows that the retained percentage of MHSA of ViT and MHSA of Bert among all compressible components significantly increases as the compression ratio increases. In contrast, the retained percentage of MLP of ViT and MLP of Bert decreases. This indicates that MHSA has higher importance than MLPs when the number of parameters is limited. It can also be observed that vision modality is more important than language modality in this task. The trend of the retained percentage of cross attention generally decreases and then increases. This phenomenon indicates that at low compression ratios, the parameters of the visual and language modalities are relatively adequate. Therefore cross attention is less important at this time. At high compression ratios, the vision and language modality lacks sufficient parameters, and cross attention becomes more critical.

Similarly, the right subfigure demonstrates the variation of all layers as the total compression ratio increases. It can be observed that the middle layers occupy an increasing proportion as the compression ratio increase, which indicates that the majority of modalities’ information is generated in the middle layers of the model. In the earlier layers, the information is not detailed enough. In contrast, in the last several layers, the refinement of the information becomes less critical when the number of parameters is limited.
The proportion of all compressible components retained in the compression model. This 6 subfigures represent the original model and the compressed model at the 2×, 3×, 4×, 5× and 10× compression ratio, respectively. In each subfigure, the horizontal axis represents the layer number, the vertical axis represents the compressible components corresponding to each ζ, and the number in cells represents the retained proportion of a certain component’s certain layer.

![Figure 1: The proportion of all compressible components retained in the compression model.](image)

<table>
<thead>
<tr>
<th>Approach</th>
<th>Reduce</th>
<th>Only Search</th>
<th>One Epoch Retrain</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Dev Acc(†)</td>
<td>Test-p Acc(†)</td>
<td>Dev Acc(†)</td>
</tr>
<tr>
<td>Simple Pruning</td>
<td>2×</td>
<td>×</td>
<td>62.82</td>
</tr>
<tr>
<td>Unified Pruning</td>
<td>2×</td>
<td>×</td>
<td>75.42</td>
</tr>
<tr>
<td>UPop</td>
<td>2×</td>
<td>76.89</td>
<td>77.84</td>
</tr>
</tbody>
</table>

### 4.4 Effect of Progressive Pruning

As shown in Table 1 at the 2× compression ratio, the Unified and Progressive Pruning (UPop) gains further 0.83% and 0.81% accuracy improvement on the dev set and test-p set compared to the Unified Pruning. Moreover, at the 3× compression, the improvements are extended to 5.64% and 5.95%, respectively. At the higher 4×, 5×, and 10× compression ratio, the Progressive Pruning can still enable the compressed model to converge successfully, while both Simple Pruning and Unified Pruning fail.

To further illustrate how Progressive Pruning strengthens the convergence capability of the compressed model, we compare the performance of extracted models in the situation of the search without retraining or search with only one epoch retrain. The Table 2 shows that the model compressed by UPop can converge without any retrain while the other two compression approaches fail. With only one epoch retrain, the model compressed by UPop converges at significantly superior performance to the other two approaches. The experiments in the Table 2 indicates that Progressive Pruning maintains the convergence capability of the compressed model by initializing the model to be retrained with better parameter weights.

### 4.5 Effect of Unified and Progressive Pruning

Unified Pruning and Progressive Pruning boost the performance of Simple Pruning in two aspects, respectively. At the same and relatively low compression ratio, the Unified Pruning gains significant performance improvements by adaptively assigning a proper compression ratio among all compress-
ible components. As the compression ratio rises, the gap in parameter weights between the search model and the model to be retrained becomes larger and larger. Therefore the compressed model will be increasingly difficult to converge. In such a situation, the Progressive Pruning plays the role of maintaining the convergence capability of the compressed model. Combined Progressive Pruning with Unified Pruning, the UPop gains the ability to achieve better performance at the same ratio and push the limitation of compression ratio to a greater extent.

4.6 Study on Growth Strategy of Compression Ratio

We also explore how the growth strategy of compression ratio affects the model performance. As explained in the section 3.3, the compression ratio $p_t$ is a monotonically increasing function of iteration number $t$. It is supposed to meet requirements that more slowly increase in early and late iterations than in middle iterations. Table 3 shows the performance compressed with different growth strategies. The first one $p_{T_s-1}$ is to increase $p_t$ uniformly as $t$ increases, while the last one $p\sqrt{(1 - \cos(\frac{\pi t}{T_s-1}))^\frac{1}{2}}$ is the strategy we actually adopted. There are obvious performance improvement when replace the $p_{T_s-1}$ with $p\sqrt{(1 - \cos(\frac{\pi t}{T_s-1}))^\frac{1}{2}}$. It is worth noting that $p\sqrt{(1 - \cos(\frac{\pi t}{T_s-1}))^\frac{1}{2}}$ is not the only feasible strategy. For example, the $p\frac{(2T_s-t+1)}{(T_s+1)t}$ is also a strategy that meet requirements mentioned above, which also achieve comparable performance to the $p\sqrt{(1 - \cos(\frac{\pi t}{T_s-1}))^\frac{1}{2}}$.

<table>
<thead>
<tr>
<th>$p_t$</th>
<th>Dev Acc</th>
<th>Test-p Acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_{T_s-1}$</td>
<td>79.94</td>
<td>80.84</td>
</tr>
<tr>
<td>$p\frac{(2T_s-t+1)}{(T_s+1)t}$</td>
<td>80.38</td>
<td>81.13</td>
</tr>
<tr>
<td>$p\sqrt{(1 - \cos(\frac{\pi t}{T_s-1}))^\frac{1}{2}}$</td>
<td>80.33</td>
<td>81.13</td>
</tr>
</tbody>
</table>

Table 3: How the growth strategy of compression ratio $p_t$ affects the model performance?

4.7 Study on the interval of Updating Compression Mask $\zeta$

We further explore whether the interval of updating mask $\zeta$ is necessary. There are at least two benefits that update compression mask $\zeta$ at intervals. The first obvious one is that it can reduce a small amount of computation during searching. Moreover, the other can be indicated from Table 4. The interval 1 means that updating $\zeta$ without interval, and it can be observed that updating the $\zeta$ too frequently causes the compressed model to tend to overfit on the validation set. Consequently, we adopt 50 as the interval of UPop to update $\zeta$, which mitigates the overfitting in the validation set and improves the performance on the test-p set.

<table>
<thead>
<tr>
<th>Interval</th>
<th>Dev Acc</th>
<th>Test-p Acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>80.97</td>
<td>80.14</td>
</tr>
<tr>
<td>10</td>
<td>80.48</td>
<td>80.86</td>
</tr>
<tr>
<td>50</td>
<td>80.33</td>
<td>81.13</td>
</tr>
</tbody>
</table>

Table 4: Research on the necessity of interval of updating compression mask $\zeta$

4.8 Compression Experiments on the Image Caption

To verify the versatility of the proposed UPop, we further conducted experiments on the Image Caption task. We compress the BLIP model fine-tuned on this task at a ratio of 2 and 4 times, respectively. The model consists of a ViT as image encoder and a Bert with cross-attention as text decoder, therefore the mask $\zeta$ corresponding to the compressible components on this model is $\zeta = \{\zeta_{att}^l, \zeta_{mlp}^l, \zeta_{att}^l, \zeta_{mlp}^l, \zeta_{att}^l\}$. Table 5 shows that UPop also achieves superior performance on this task.

4.9 Extended Experiments on the Image-Text Retrieval

Additionally, we also conducted experiments on Image-Text Retrieval task. We compress the BLIP model fine-tuned on this task at a ratio of 2 and 4 times, respectively. The model consists of a ViT as image encoder, a Bert with cross-attention as text encoder, an extra ViT as momentum image
Table 5: Compression results on the Image Caption task.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Reduce</th>
<th>CIDEr(↑)</th>
<th>SPICE(↑)</th>
<th>Params(M)</th>
<th>FLOPs(G)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original Model</td>
<td>1×</td>
<td>133.3</td>
<td>23.8</td>
<td>224.0</td>
<td>65.7</td>
</tr>
<tr>
<td>Simple Pruning (Algorithm 2)</td>
<td>2×</td>
<td>112.9</td>
<td>21.0</td>
<td>124.9</td>
<td>33.2</td>
</tr>
<tr>
<td>Unified Pruning (Algorithm 3)</td>
<td>4×</td>
<td>60.7</td>
<td>12.8</td>
<td>75.4</td>
<td>17.1</td>
</tr>
<tr>
<td>UPop (Algorithm 1)</td>
<td>2×</td>
<td>127.7</td>
<td>23.0</td>
<td>127.0</td>
<td>39.6</td>
</tr>
<tr>
<td></td>
<td>4×</td>
<td>100.3</td>
<td>19.1</td>
<td>77.5</td>
<td>25.6</td>
</tr>
</tbody>
</table>

encoder, and an extra Bert as momentum text encoder. Since the momentum models are updated by taking the moving average of normal models, we do not add the compression mask into the momentum models. Therefore, the mask \( \zeta \) corresponding to the compressible components on this model is \( \zeta = \{ \zeta_{v}^{att}, \zeta_{v}^{mlp}, \zeta_{l}^{att}, \zeta_{l}^{mlp}, \zeta_{c}^{att} \} \). Table 6 shows the improved performance of UPop on this task.

Table 6: Compression results on the Image-Text Retrieval task.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Reduce</th>
<th>Status</th>
<th>TR@1 (↑)</th>
<th>IR@1 (↑)</th>
<th>Params(M)</th>
<th>FLOPs(G)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original Model</td>
<td>1×</td>
<td>✓</td>
<td>81.90</td>
<td>64.31</td>
<td>447.64</td>
<td>153.24</td>
</tr>
<tr>
<td>Simple Pruning (Algorithm 2)</td>
<td>2×</td>
<td>✓</td>
<td>61.66</td>
<td>46.01</td>
<td>249.52</td>
<td>77.28</td>
</tr>
<tr>
<td></td>
<td>4×</td>
<td>✗</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Unified Pruning (Algorithm 3)</td>
<td>2×</td>
<td>✓</td>
<td>75.44</td>
<td>57.64</td>
<td>253.05</td>
<td>103.44</td>
</tr>
<tr>
<td></td>
<td>4×</td>
<td>✓</td>
<td>40.30</td>
<td>31.31</td>
<td>148.73</td>
<td>61.44</td>
</tr>
<tr>
<td>UPop (Algorithm 1)</td>
<td>2×</td>
<td>✓</td>
<td>77.36</td>
<td>59.77</td>
<td>248.90</td>
<td>88.28</td>
</tr>
<tr>
<td></td>
<td>4×</td>
<td>✓</td>
<td>62.88</td>
<td>47.41</td>
<td>147.92</td>
<td>50.20</td>
</tr>
</tbody>
</table>

5 CONCLUSION

This paper proposes a multimodal compression approach, Unified and Progressive Pruning (UPop), for vision-language Transformers. UPop unifiedly search on all compressible components, which consists of MHSA, MLP of all modalities, and cross-attentions, and therefore can adaptively assign proper compression ratio for all components. Furthermore, analysis of compression masks indicates that the importance of components for compression varies. Therefore, the proposed unified search is a better choice than manually assigning compression ratios among different components, which is inefficient and resource-consuming. Furthermore, UPop conducts search and retrain progressively, which effectively strengthens the convergence capability of the compressed model and pushes the limit of compression ratio to a greater extent. Finally, UPop is a practically deployable compression approach that physically extracts the compressed model from the original model.

LIMITATIONS

We only conduct experiments on the vision-language tasks and models. However, there are also multimodal tasks and models of other modalities, and experiments on them are necessary to verify the versatility of the proposed UPop further. Moreover, there is still room for compressing structures not included in the compressible components described in this paper.
REFERENCES


Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In International Conference on Machine Learning, pp. 4904–4916. PMLR, 2021.


Under review as a conference paper at ICLR 2023


A Notations and Descriptions

Table 7: Notations and their corresponding descriptions.

<table>
<thead>
<tr>
<th>NOTATION</th>
<th>DESCRIPTION</th>
<th>NOTATION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>( L^{(v,l)} )</td>
<td>Number of layers of {ViT, Bert}</td>
<td>( H^{(v,l)} )</td>
<td>Number of heads of {ViT, Bert}</td>
</tr>
<tr>
<td>( N^{(v,l)} )</td>
<td>Number of patches of {ViT, Bert}</td>
<td>( D^{(v,l)} )</td>
<td>Embedding size of {ViT, Bert}</td>
</tr>
<tr>
<td>( d^{(v,l)} )</td>
<td>Embedding size of each head</td>
<td>( M^{(v,l)} )</td>
<td>Size of intermediate tensor of MLP of {ViT, Bert}</td>
</tr>
<tr>
<td>( f_1^{(v,l)} )</td>
<td>The first fully connected layer</td>
<td>( f_2^{(v,l)} )</td>
<td>The second fully connected layer</td>
</tr>
<tr>
<td>( \zeta_{att} )</td>
<td>Mask for MHSA of ViT</td>
<td>( \zeta_{m,lp} )</td>
<td>Mask for MLP of ViT</td>
</tr>
<tr>
<td>( \zeta_{att} )</td>
<td>Mask for MHSA of Bert</td>
<td>( \zeta_{m,lp} )</td>
<td>Mask for MLP of Bert</td>
</tr>
<tr>
<td>( \zeta_{c} )</td>
<td>Mask for MHSA of cross-attention</td>
<td>( \zeta )</td>
<td>{( \zeta_{att}, \zeta_{m,lp}, \zeta_{att}, \zeta_{m,lp}, \zeta_{att} )}</td>
</tr>
<tr>
<td>( w_{(a,m)} )</td>
<td>Coefficient of ( \ell_1 )-norm about {( \zeta_{a}, \zeta_{m} )}</td>
<td>( p )</td>
<td>Total compression ratio</td>
</tr>
<tr>
<td>( \Theta )</td>
<td>Parameters of the original model</td>
<td>( \mathcal{L}_O )</td>
<td>Loss function of the original model</td>
</tr>
<tr>
<td>( \mathcal{F}_p )</td>
<td>( p % ) compressed model ( \mathcal{F}_p(x</td>
<td>\Theta, \zeta) )</td>
<td>( \mathcal{F}^* )</td>
</tr>
<tr>
<td>{( \alpha, \beta )}</td>
<td>Learning rate during {search, retrain}</td>
<td>( T_{(s,r)} )</td>
<td>Max iterations of {search, retrain}</td>
</tr>
</tbody>
</table>

B IMPLEMENTATION OF SIMPLE PRUNING

Algorithm 2 Simple Pruning for Compressing Vision-Language Transformers

Input: \( \zeta, \zeta_{a}, \zeta_{m}, \Theta, \mathcal{F}, p, T_s, T_r, \alpha, \beta \)

for \( t \leftarrow 0 \) to \( T_s - 1 \) do

\[ \mathcal{L} \leftarrow \mathcal{L}_O + w_a \sum_{i \in \zeta} \| \zeta_i \|_1 + w_m \sum_{i \in \zeta_{m}} \| \zeta_i \|_1 \]

\[ \Theta^{(t+1)} \leftarrow \Theta^{(t)} - \alpha \frac{1}{n} \sum_{i=1}^{n} \nabla_{\Theta} \mathcal{L}(\Theta^{(t)}, \zeta^{(t)}) \]

for \( \zeta_i \in \zeta \) do

\[ \mathcal{M}_i \leftarrow \text{TopKM} \text{Mask}(\zeta_i^{(T_r)}; p \cdot \text{size}(\zeta_i)) \]

\[ \hat{\Theta} \leftarrow \{\Theta_i^{(T_r)} | \mathcal{M}_i = 1\}, \quad \mathcal{F}_p \leftarrow \mathcal{F}(x|\hat{\Theta}, \zeta^{(T_r)}) \]

for \( t \leftarrow 0 \) to \( T_r - 1 \) do

\[ \hat{\Theta}^{(t+1)} \leftarrow \Theta^{(t)} - \beta \frac{1}{n} \sum_{i=1}^{n} \nabla_{\hat{\Theta}} \mathcal{L}_O(\hat{\Theta}^{(t)}) \]

return \( \mathcal{F}^* \leftarrow \mathcal{F}_p(x|\hat{\Theta}^{(T_r)}) \)

Search For the compression of ViT, additional trainable masks \( \zeta \) are initialized to 1 and inserted into MHSA and MLP layers for each transformer block of ViT. Denote the input of the \( l^{th} \) MHSA as

\[ a_l^{(v)} \in \mathbb{R}^{N^v \times D^v} \quad (9) \]

Every head \( h \) in the MHSA will then transform \( a_l^{(v)} \) into query, key, and value:

\[ q_{l,h}, k_{l,h}, v_{l,h} \in \mathbb{R}^{N^v \times d^v} \quad (10) \]

Then the attention map of each head can be derived from

\[ A_{l,h} = \text{Softmax}((q_{l,h} \odot \zeta_{att,l,h}^{(v)}) \times (k_{l,h} \odot \zeta_{att,l,h}^{(v)}))^{T} / \sqrt{d} \quad (11) \]

where \( \zeta_{att}^{(v)} \in \mathbb{R}^{L^v \times 1 \times d^v} \). And the corresponding output can be derived from

\[ O_{l,h} = A_{l,h} \times (v_{l,h} \odot \zeta_{att,l,h}^{(v)}) \in \mathbb{R}^{N \times d} \quad (12) \]

Denote the input of MLP as

\[ m_l^{(v)} \in \mathbb{R}^{N^v \times D^v} \quad (13) \]

Then the output of MLP can be derived from

\[ a_{l+1}^{(v)} = f_2^{(v)}(f_1^{(v)}(m_l^{(v)}) \odot \zeta_{mlp}^{(v)}) \in \mathbb{R}^{N^v \times D^v} \quad (14) \]
where $\zeta_{mlp}^\prime \in \mathbb{R}^{L \times M}$. Since the compressible components in Bert are also MHSA and MLP, and the compressible cross-attention is also a type of MHSA, the search on Bert and cross-attention can be derived similarly.

Besides, the $\ell_1$-norm of masks $\zeta$ are added as additional loss terms to drive the magnitude of masks smaller and smaller while searching:

$$
\mathcal{L} = \mathcal{L}_C + w_a \sum_{\zeta_i \in \zeta_a} \|\zeta_i\|_1 + w_m \sum_{\zeta_i \in \zeta_m} \|\zeta_i\|_1
$$

(15)

**Retrain** The extracted model is retrained during the retraining phase. For each mask $\zeta_i \in \zeta$, dimensions with the smallest magnitude of $p\%$ in the mask are eliminated from the original model. Then retrain the model after elimination, and the compressed model can be obtained finally.

### C IMPLEMENTATION OF UNIFIED PRUNING

**Algorithm 3 Unified Pruning for Compressing Vision-Language Transformers**

**Input:** $\zeta$, $\zeta_a$, $\zeta_m$, $\Theta$, $\mathcal{F}$, $p$, $T_s$, $T_r$, $\alpha$, $\beta$

1. for $t \leftarrow 0$ to $T_s - 1$
2. \hspace{1em} $\mathcal{L} \leftarrow \mathcal{L}_C + w_a \sum_{\zeta_i \in \zeta} \|\zeta_i\|_1 + w_m \sum_{\zeta_i \in \zeta_m} \|\zeta_i\|_1$
3. \hspace{1em} $\Theta^{(t+1)} \leftarrow \Theta^{(t)} - \alpha \frac{1}{n} \sum_i \nabla_{\Theta} \mathcal{L}(\Theta^{(t)},\zeta^{(t)})$
4. \hspace{1em} $\zeta^{(t+1)} \leftarrow \zeta^{(t)} - \alpha \frac{1}{n} \sum_i \nabla_{\zeta} \mathcal{L}(\Theta^{(t)},\zeta^{(t)})$
5. \hspace{1em} $\zeta_s \leftarrow \frac{\mathbb{E}[\zeta_s^{(T_s)} - E(\zeta_s^{(T_s)})]}{\sqrt{\mathbb{E}[\|\zeta_s^{(T_s)} - E(\zeta_s^{(T_s)})\|^2]}}$
6. $M \leftarrow \text{TopKMask}(\zeta^{(T_s)}, p \cdot \text{Size}(\zeta))$
7. $\hat{\Theta} \leftarrow \{ \Theta^{(T_s)} | M = 1 \}$, $\mathcal{F}_p \leftarrow \mathcal{F}(x|\hat{\Theta},\zeta^{(T_s)})$
8. for $t \leftarrow 0$ to $T_r - 1$
9. \hspace{1em} $\hat{\Theta}^{(t+1)} \leftarrow \hat{\Theta}^{(t)} - \beta \frac{1}{n} \sum_i \nabla_{\Theta} \mathcal{L}(\hat{\Theta}^{(t)})$
10. return $\mathcal{F}^* \leftarrow \mathcal{F}_p(x|\hat{\Theta}^{(T_r)})$

### D GROWTH IN COMPRESSION RATIO

According to the implementation of Algorithm 1, the current compression ratio $p_t$ of $t^{th}$ iteration means that $p_t\%$ of embeddings has been compressed by $\frac{p_t}{p}$. As a consequence, the actual compression ration $a_t$ should be the ratio of the compressed embedding size multiplied by the ratio of each embedding that is compressed:

$$
a_t = p_t \times \frac{p_t}{p} = \frac{p_t^2}{p}
$$

(16)

As described in Section 3.3, the function of $a^t$ concerning $t$ is expected to grow at a rate that first increases and then decreases. In other words, $a_t$ is expected to satisfy:

$$
\begin{align*}
a_t &= 0 \\
\frac{da_t}{dt} &\geq 0, \forall t \in [0, T_s - 1] \\
\exists t_0 \in (0, T_s) \text{ s.t. } \frac{d^2a_t}{dt^2} &> 0, \forall t \in (0, t_0), \text{ and } \frac{d^2a_t}{dt^2} &< 0, \forall t \in (t_0, T_s - 1)
\end{align*}
$$

(17)

For example, the integration of trigonometric function $\frac{\pi x}{T_s - 1}$ over $0 \leq t < T_s - 1$ obviously satisfy the latter two requirements of the Equation 17. To further satisfy the first two properties, we have:

$$
p_t = \frac{\int_0^{T_s - 1} \sin \frac{\pi x}{T_s - 1} dx}{\int_0^{T_s - 1} \sin \frac{\pi x}{T_s - 1} dx} = \frac{p}{2}(1 - \cos \frac{\pi t}{T_s - 1}) = a_t = \frac{p_t^2}{p}
$$

(18)

And thus

$$
p_t = p \sqrt{\left(1 - \cos \frac{\pi t}{T_s - 1}\right) \frac{1}{2}}
$$

(19)

is a function that satisfies all requirements.
E  EXPERIMENTAL DETAILS

E.1  COMPRESSIBLE COMPONENTS

Multi-Head Self-Attention (MHSA) and Multi-Layer Perceptron (MLP) are widely used structures in every transformer layer. Consequently, the compressible components for multimodal models in our experiments include MHSA of ViT, MLP of ViT, MHSA of Bert, MLP of Bert, and cross-attention. Note that cross-attention is also a type of MHSA.

E.2  PRACTICALITY

From a practical point of view, an essential difference between our approach and ViT-Slimming is that for embedding compression, our approach allows the compressed model to be physically extracted from the large model and actually accelerate the model, whiles ViT-Slimming cannot. ViT-Slimming compress heads of MHSA with unrestricted compression ratio, and thus the compressed model may have different embedding sizes of heads within a layer. However, the computation of the attention map requires each head of the query and key within a layer have the same embedding size. By restricting each head within the same layer to have the same compression ratio, UPop extracts all compressed models in the experiments and reports the model performance tested on them.

F  VARIATION OF COMPRESSIBLE COMPONENTS AND LAYERS AS COMPRESSION RATIO INCREASES.

Figure 2: The left subfigure: variation of compressible components as the compression ratio increases. The right subfigure: variation of layers as the compression ratio increases.