SmartTrim: Adaptive Tokens and Attention Pruning for Efficient Vision-Language Models

Zekun Wang[1*], **Jingchang Chen**[1*], **Wangchunshu Zhou**[2], **Haichao Zhu**
Jiafeng Liang[1], **Liping Shan**[3], **Ming Liu**[1,4], **Dongliang Xu**[3], **Qing Yang**[3],
Bing Qin[1,4]
[1]Harbin Institute of Technology [2] ETH Zurich
[3]Du Xiaoman (Beijing) Science Technology Co., Ltd [4] Peng Cheng Laboratory
{zkwang, jcchen, mliu, qinb}@ir.hit.edu.cn

Abstract

Despite achieving remarkable performance on various vision-language tasks, Transformer-based Vision-Language Models (VLMs) suffer from redundancy in inputs and parameters, significantly hampering their efficiency in real-world applications. Moreover, the degree of redundancy in token representations and model parameters, such as attention heads, varies significantly for different inputs. In light of the challenges, we propose SmartTrim, an adaptive acceleration framework for VLMs, which adjusts the computational overhead per instance. Specifically, we integrate lightweight modules into the original backbone to identify and prune redundant token representations and attention heads within each layer. Furthermore, we devise a self-distillation strategy to enhance the consistency between the predictions of the pruned model and its fully-capacity counterpart. Experimental results across various vision-language tasks consistently demonstrate that SmartTrim accelerates the original model by 2-3 times with minimal performance degradation, highlighting the effectiveness and efficiency compared to previous approaches. Code will be available at `https://github.com/kugwzk/SmartTrim`.

Keywords: Vision-Language Model, Adaptive Inference, Pruning, Dynamic Network

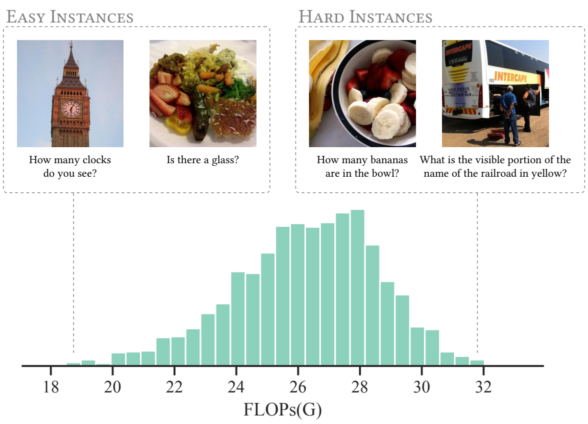

Figure 1: FLOPs histogram of SmartTrim on VQA. SmartTrim allocates diverse computational overhead based on cross-modal complexity, assigning fewer computations to **easy** instances (left) and more to **hard** ones (right).

1. Introduction

Transformer-based (Vaswani et al., 2017) Vision-Language Models (VLMs) have shown great success on various vision-language tasks with their delicate model structures (Radford et al., 2021; Wang et al., 2023b; Chen et al., 2023). Despite achieving superior performance, these models are computationally expensive due to the long input sequences and large number of parameters, hindering their deployment in the production environment.

In pursuit of efficient VLMs, a few acceleration approaches have been proposed, including knowledge distillation (Fang et al., 2021; Wang et al., 2023a), parameter pruning (Gan et al., 2022; Shi et al., 2023), and token pruning (Jiang et al., 2022; Cao et al., 2023). These methods reduce inference overhead, implying that a large proportion of parameters and token representations are redundant. However, they adhere to a static computational architecture for all instances, overlooking the variation of complexities among different instances, leading to severe performance degradation at higher acceleration ratios (Kaya et al., 2019; Liu et al., 2020). As demonstrated in Figure 1, the instances involving complex cross-modal interactions naturally require more computations to fully comprehend the intricate details of images and associated questions. Conversely, easy instances can be solved with less overhead. Consequently, enormous original VLMs may overthink simple instances, leading to wasted computation, while static accelerated models struggle with complex ones, incurring extensive performance degradation.

To this end, we focus on adaptive acceleration on a per-input basis, which is orthogonal to static approaches and more flexible to meet different constraints. In this work, we propose SmartTrim, an adaptive pruning framework for VLM (shown in Figure 2), which streamlines the model from two aspects with significant redundancy: token represen-

LREC-COLING 2024, pages 14937–14953
20-25 May, 2024.

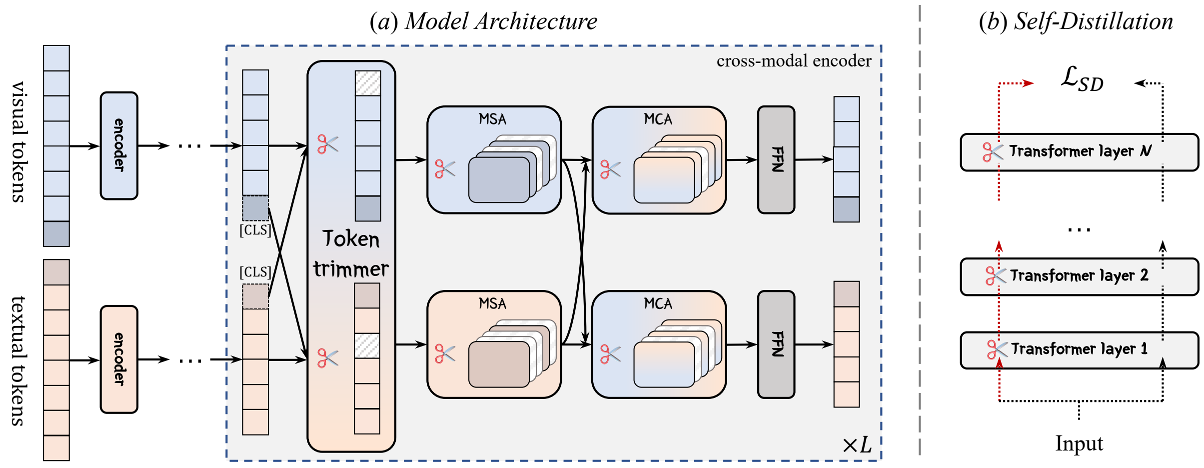

Figure 2: Overview of our SmartTrim framework, best viewed in color. (a) **Model Architecture** of SmartTrim. We incorporate the trimmers into layers of the uni-modal encoders and the cross-modal encoder to prune redundant tokens and heads. Given a set of image-text pairs, SmartTrim adjusts the computations for each instance based on the trimmer outputs. (b) **Self-Distillation** strategy. At each training step, the predictions of the pruned model are aligned to its fully-capacity counterpart.

tation and attention heads. SmartTrim integrates the lightweight modules (called trimmers) into layers of the original backbone to identify redundant tokens and heads guided by cross-modal information. Specifically, the *XModal-aware token trimmers* are introduced to determine which tokens to retain considering not only their representations but also their importance in cross-modal interactions. For head pruning, we introduce *Modal-adaptive head trimmers* in different attention modules to adaptively select which heads to activate. During training, we propose a self-distillation strategy, which encourages the predictions of the pruned model to align with its fully-capacity counterpart at the same step. The self-distillation scheme alleviates the need for a separately fine-tuned teacher model in conventional knowledge distillation. Furthermore, with a curriculum training scheduler, SmartTrim has a smoother and more stable optimization process. Compared to previous methods, our approach not only avoids additional expensive pre-training, but also provides more fine-grained control to better explore efficiency-performance trade-offs.

We evaluate the proposed SmartTrim on two representative VLMs with different architectures: METER (Dou et al., 2022), an encoder-based model; and BLIP (Li et al., 2022), an encoder-decoder-based model. Experimental results reveal that SmartTrim consistently outperforms previous methods on various datasets. Notably, SmartTrim achieves an impressive speed-up from $1.5\times$ to $4\times$ on the original model while incurring only a marginal performance drop (1%~3%). Further analysis indicates that SmartTrim effectively learns to adaptively allocate computational budgets based on the complexity of cross-modal interactions.

2. Preliminary

2.1. Transformer-based VLM

Uni-Modal Encoders The input image and text are tokenized into visual and textual tokens, respectively. The two sequences are fed into visual and textual encoders to extract the respective features, where each layer consists of a multi-head self-attention module (MSA) and a feed-forward network module (FFN).

Cross-Modal Encoder To capture cross-modal interactions, the co-attention mechanism (Lu et al., 2019) is employed in each layer of cross-modal encoder. Specifically, in addition to MSA and FFN, a multi-head cross-attention module (MCA) is introduced, where query features are projected from one modality (*e.g.*, vision), while key and value features are obtained from another modality (*e.g.*, language).

2.2. Empirical Analyses

The long sequence in VLMs incurs substantial computational overhead as the complexity of attention modules scales quadratically with length. In addition, hundreds of millions of parameters further burden the situation. Previous studies of uni-modal Transformers reveal that redundancy is present in token representations or attention heads (Michel et al., 2019; Goyal et al., 2020; Wang et al., 2022a). To investigate whether redundancy also exists in

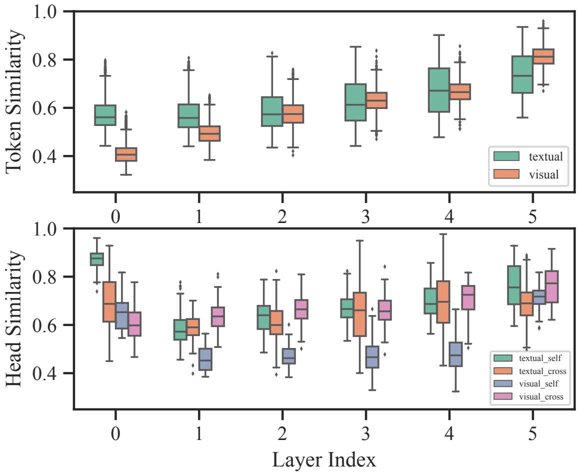

Figure 3: The similarities in representations of tokens (top) and heads (bottom) in cross-modal encoder of METER fine-tuned on VQA.

VLMs, we measure cosine similarities between different token representations and heads at each layer of a fine-tuned METER. As shown in Figure 3, our empirical findings are as follows: ❶ Similarities between the representations of tokens and heads are consistently high across all layers, implying significant redundancy within the model. ❷ The similarity of token representations increases progressively with depth, indicating a growing redundancy in deeper layers. ❸ Similarities vary greatly between instances, prompting the need to investigate input-dependent adaptive pruning.

3. Methodology

In this section, we introduce the proposed adaptive pruning method for VLMs named SmartTrim, as shown in Figure 2. We first describe the details of adaptive trimmers and then introduce the end-to-end training recipe for SmartTrim.

3.1. Adaptive Trimmers

XModal-Aware Token Trimmer As shown in Figure 2 (a), SmartTrim progressively prunes token representations in blocks, delivering more important tokens to subsequent blocks, and eliminating the rest[1]. To estimate the importance of token representations, we insert a lightweight MLP-based module (named *XModal-aware trimmer*) before each block of uni-modal and cross-modal encoders. Taking the cross-modal encoder block, for example, the N_t token representations $\boldsymbol{X} \in \mathbb{R}^{N_t \times D}$ are first fed into the *local* policy network:

$$\boldsymbol{\pi_t^l} = \mathrm{MLP}_t(\boldsymbol{X}') = \mathrm{MLP}_t(\mathrm{Linear}(\boldsymbol{X}))$$

[1]We retain `[CLS]` tokens in each block of model.

where $\boldsymbol{\pi_t^l} \in \mathbb{R}^{N_t}$ is the local importance score of tokens, $\boldsymbol{X}' \in \mathbb{R}^{N_t \times D'}$ is obtained by the dimension reduction of $\boldsymbol{X}$. The $\boldsymbol{\pi_t^l}$ is only computed based on the independent representations of tokens, without considering their contribution in cross-modal interactions. To estimate the importance of cross-modal interactions without imposing excessive additional computation, we fuse global representations [2] of visual and textual modality and then project to obtain the cross-modal global representation $\boldsymbol{g}$, which contains global information of both modalities. Then, we feed $\boldsymbol{g}$ and $\boldsymbol{X}'$ to the *global* policy network to calculate the XModal-global importance score $\boldsymbol{\pi_t^g}$:

$$\boldsymbol{\pi_t^g} = \mathrm{norm}(\boldsymbol{g}\boldsymbol{W}_g\boldsymbol{X}'^{\intercal})$$

where $\boldsymbol{W}_g$ is the projection layer. The final token importance score $\boldsymbol{\pi_t}$ sums $\boldsymbol{\pi_t^l}$ and $\boldsymbol{\pi_t^g}$: $\pi_t = \pi_t^l + \pi_t^g$. During inference, the pruning mask $\boldsymbol{M}_t \in \{0,1\}^{N_t}$ is sampled directly from $\mathrm{sigmoid}(\boldsymbol{\pi_t})$: 1 indicates that the token is retained; otherwise, the token is removed. By this pruning, our token trimmers reduce the amount of computation in both the attention and FFN modules for subsequent blocks.

Modal-adaptive Head Trimmer The VLMs capture intra-modal and inter-modal interactions via MSA and MCA, respectively. However, the computational overhead required for modeling varies depending on the input complexity of attention, leading to redundancy in attention modules, as shown in Section 2.2. To this end, we integrate the *modal-adaptive* head trimmer into the attention modules. Specifically, we take the global representations of input sequences to feed into head trimmers:

$$\boldsymbol{\pi_h} = \begin{cases} \mathrm{MLP}_h^{self}(\boldsymbol{x}_{\text{cls}}) & \text{(MSA)} \\ \mathrm{MLP}_h^{cross}([\boldsymbol{x}_{\text{cls}}, \boldsymbol{y}_{\text{cls}}]) & \text{(MCA)} \end{cases}$$

where $\boldsymbol{x}_{\text{cls}}, \boldsymbol{y}_{\text{cls}}$ are the `[CLS]` representations of the self-modality and another modality, respectively. Like the token trimmer, the head trimmer samples $\boldsymbol{M}_h$ from $\mathrm{sigmoid}(\boldsymbol{\pi_h})$ to determine which heads to keep or remove.

Note that our trimmers introduce only a minor number of parameters (3%) that yield a negligible computational overhead on FLOPs (1%) compared to the original backbone. In addition, adaptive trimmers are more hardware-friendly by avoiding the use of costly operations like top-*k* in other methods (Wang et al., 2021).

3.2. Training Recipe

The adaptive trimmers are seamlessly integrated into the backbone network fine-tuned with the task-

[2]We choose the representations of `[CLS]` tokens as global representations of each modality, which is better than other strategies in preliminary experiments, such as average or attentive pooling.

specific objective $\mathcal{L}_{Task}$. To achieve end-to-end optimization, we adopt the reparameterization technique (Jang et al., 2017) to sample discrete masks $\boldsymbol{M}$ from the output distributions of trimmers:

$$\boldsymbol{M} = \frac{\exp((\boldsymbol{\pi} + \boldsymbol{G}')/\tau)}{\exp((\boldsymbol{\pi} + \boldsymbol{G}')/\tau) + \exp(\boldsymbol{G}''/\tau)} \tag{1}$$

where $\boldsymbol{G}'$ and $\boldsymbol{G}''$ are two independent Gumbel noises, and τ is a temperature factor. To better control the overall computations of the model, we introduce a cost loss $\mathcal{L}_{Cost}$:

$$\mathcal{L}_{Cost} = (\beta_{\mathcal{T}} - \gamma_{\mathcal{T}})^2 + (\beta_{\mathcal{H}} - \gamma_{\mathcal{H}})^2 \tag{2}$$

$$\beta_{\mathcal{T}} = \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}}\frac{m_t}{N_t}, \beta_{\mathcal{H}} = \frac{1}{|\mathcal{H}|}\sum_{h\in\mathcal{H}}\frac{m_h}{N_h} \tag{3}$$

where $\beta_{\mathcal{T}}$ and $\beta_{\mathcal{H}}$ represent the retention ratios of tokens and attention heads for each example in the batch. $\mathcal{T}$ and $\mathcal{H}$ are the sets of modules with token and head trimmers, respectively. γ is the overall target budget for token and head trimmers set in advance. $m = \|\boldsymbol{M}\|_0$ and N represent the retained and total number of tokens or heads in the module.

Self-Distillation During training, we propose a self-distillation objective to encourage the predictions of the pruned model θ_s, to align with its fully-capacity counterpart θ_t, as shown in Figure 2 (b). Note that θ_s and θ_t are **share** parameters, the only difference is that the trimmers are activated in the forward of θ_s while disabled in θ_t. At each training step, both the sparse and full models are optimized simultaneously. The self-distillation objective $\mathcal{L}_{SD}$ is calculated as:

$$\mathcal{L}_{SD} = \mathcal{L}_{Task}(\theta_t, y) + D_{KL}(p(\theta_s, x) \parallel p(\theta_t, x))$$

where x is the input and p are output logits. This scheme alleviates the need for additional fine-tuned teacher models in traditional knowledge distillation. The overall training objective of SmartTrim is as follows:

$$\mathcal{L} = \mathcal{L}_{Task} + \lambda_{SD}\mathcal{L}_{SD} + \lambda_{Cost}\mathcal{L}_{Cost} \tag{4}$$

where $\lambda_{SD}, \lambda_{Cost}$ are hyperparameters.

Curriculum Training Integrating trimmers into the pretrained backbone introduces drastic adaptation to the original parameters, which potentially causes vulnerable and unstable training. To enhance the stability of optimization, we propose a training scheduler driven by curriculum learning (Bengio et al., 2009). Specifically, at the beginning of training, we initialize trimmers to ensure the retention of all tokens and heads. Subsequently, we linearly decrease the ratio γ from 1.0 to the target ratio over a specified percentage of steps. In this way, we encourage the training to focus on downstream tasks initially and then gradually learn adaptive pruning.

4. Experiments

4.1. Setup

Evaluation Datasets and Metrics We consider a diverse set of visual-language downstream tasks for evaluation: NLVR2 (Suhr et al., 2019), VQA (Goyal et al., 2017) and SNLI-VE (Xie et al., 2019) for vision-language understanding, Flickr30K (Plummer et al., 2015) for image-text retrieval, COCO (Lin et al., 2014) and No-Caps (Agrawal et al., 2019) for image captioning. We report the accuracy for vision-language understanding tasks, and mean recall metrics for image retrieval (IR) and text retrieval (TR). BLEU-4, CIDEr and SPICE are used to evaluate image captioning.

Implementation Details We adopt the pretrained METER and BLIP as backbones to initialize SmartTrim. The adaptive trimmers consist of two linear layers with GeLU activation (Hendrycks and Gimpel, 2016), we set $D' = D/12$. Fine-tuning hyperparameters mainly follow the defaults in Dou et al. (2022) and Li et al. (2022). We set λ_{Cost} to 20.0 and λ_{SD} to 1.0. Curriculum training is performed within the 60% training step. We employ FLOPs as the efficiency measurement of the models, which is hardware-independent[3].

Baselines We compare SmartTrim with the following VLM acceleration methods in the task-specific fine-tuning setting. On the METER backbone: **Fine-tuning Knowledge Distillation (FTKD)**, which initializes the student model by truncating the pretrained backbone following Sun et al. (2019) and then fine-tunes the model with logits/hidden representation/attention distillation objectives the same as Jiao et al. (2020). **TRIPS** (Jiang et al., 2022), which performs static token pruning based on attention scores to reduce the number of tokens in the visual encoder. Note that we reimplement the method directly in the fine-tuning stage without additional pre-training for a fair comparison. **PuMer** (Cao et al., 2023), which is another static acceleration method that utilizes token pruning and merging. Note that PuMer only prunes tokens in the cross-modal encoder. **MuE** (Tang et al., 2023), the only previous adaptive acceleration approach for VLM, which performs early exiting in terms of the similarities of layer-wise features. We exhaustively search for the optimal settings and hyperparameters for the reimplemented baselines. On the BLIP backbone, we mainly compare with the previous state-of-the-art method **UPop** (Shi et al., 2023),

[3] To prevent pseudo-improvement caused by pruning padding tokens, we evaluate without padding (single instance usage), similar to previous work (Ye et al., 2021; Modarressi et al., 2022).

Methods	NLVR2		VQA	SNLI-VE		ITR		FLOPs(G)
	dev	test-P	test-dev	val	test	IR	TR	
METER (backbone) (Dou et al., 2022)	82.05	82.32	77.43	81.24	80.91	92.5	98.1	88.5
MiniVLM (Wang et al., 2020a)	73.71	73.93	69.10	-	-	-	-	-
DistillVLM (Fang et al., 2021)	-	-	69.80	-	-	-	-	-
EfficientVLM (Wang et al., 2023a)	81.83	81.72	76.20	-	-	-	-	-
$1.5\times$ acceleration ratio								
MuE† (Tang et al., 2023)	66.26	66.34	72.44	75.73	75.88	65.7	86.8	66.4
TRIPS† (Jiang et al., 2022)	81.34	82.01	76.50	80.55	80.57	91.8	97.5	59.0
PuMer (Cao et al., 2023)	-	82.20	76.80	-	80.30	91.7	97.6	64.7
SmartTrim	**81.89**	**82.72**	**77.25**	**80.92**	**80.90**	**92.1**	**97.9**	56.0
$2.0\times$ acceleration ratio								
FTKD	76.89	77.49	68.23	77.12	77.21	77.1	86.5	48.2
TRIPS† (Jiang et al., 2022)	80.42	81.35	75.92	80.65	80.47	90.4	96.9	47.1
SmartTrim	**82.02**	**81.97**	**77.13**	**80.67**	**80.86**	**91.6**	**97.8**	46.0
$2.5\times$ acceleration ratio								
FTKD	65.86	67.10	59.32	73.30	73.27	✗	✗	32.4
TRIPS† (Jiang et al., 2022)	77.90	78.91	72.50	79.80	79.60	86.9	94.6	32.8
SmartTrim	**81.18**	**81.55**	**76.60**	**80.53**	**80.57**	**89.8**	**96.8**	30.7

Table 1: Results of acceleration methods on various downstream vision-language tasks with different acceleration ratios. FLOPs are measured on VQA with the same hyper-parameters. † means the reimplementations by us. The marker ✗ indicates methods do not achieve promising results. The best results for each ratio are marked with **boldface**. The results are averaged over 3 runs with different seeds. For a fair comparison, we de-emphasize MiniVLM, DistillVLM, EfficientVLM (by using gray color) since they require additional pre-training and based on different backbones.

Methods	NLVR2		VQA	COCO FT			NoCaps ZS	
	dev	test-P	test-dev	B@4	C	S	C	S
BLIP (backbone) (Li et al., 2022)	82.57	82.53	78.2	39.9	133.3	23.8	109.3	14.7
$2.0\times$ acceleration ratio								
UPop (Shi et al., 2023)	80.33	81.13	76.3	-	128.9	23.3	-	-
SmartTrim	**82.24**	**82.83**	**78.0**	**39.3**	**130.8**	**23.4**	**106.4**	**14.6**
$4.0\times$ acceleration ratio								
UPop (Shi et al., 2023)	72.85	73.55	74.5	-	117.4	21.7	-	-
SmartTrim	**82.03**	**82.35**	**77.9**	**38.2**	**128.2**	**23.0**	**104.8**	**14.2**

Table 2: Results of acceleration methods with BLIP backbone on various vision-language tasks across different acceleration ratios. The results are averaged over 3 runs with different seeds. B@4: BLEU@4, C: CIDEr, S: SPICE.

which simultaneously prunes and retrains the backbone in a unified progressive pruning manner. For reference, we also present the results of efficient VLMs that need additional pre-training, including MiniVLM (Wang et al., 2020a), DistillVLM (Fang et al., 2021) and EfficientVLM (Wang et al., 2023a).

4.2. Experimental Results

Overall Performance We present the evaluation results based on the METER and BLIP architectures in Table 1 and Table 2, respectively. On the METER, SmartTrim effectively retains the performance of the original model (97.1% ~ 100.0%), while enjoying considerable speed-up, ranging from $1.5\times$ to $2.5\times$. To verify the generalizability of our approach, we also conduct an evaluation using BLIP as the backbone: SmartTrim achieves competitive results compared to the original model in ratios of $2\times$ and $4\times$. Compared to static acceleration baselines, SmartTrim significantly outperforms previous methods across various ratios and backbones, reflecting the effectiveness of our proposed adaptive pruning. Furthermore, we observe that MuE, a previous adaptive acceleration VLM, performs poorly on challenging VL tasks (*e.g.,* NLVR2 and VQA), which is due to its discarding of the entire layers of the model during inference. In contrast,

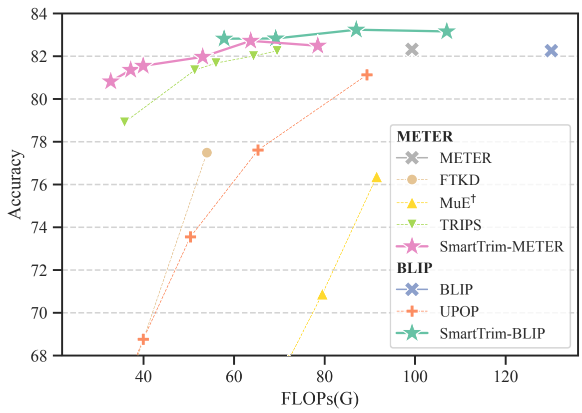

Figure 4: Pareto front of the efficiency-performance trade-offs of acceleration methods based on METER or BLIP backbones.

Models	Ratio	NLVR2 dev	NLVR2 test-P	VQA test-dev
UPop	$2\times$	80.33	81.13	76.3
	$4\times$	72.85	73.55	74.5
$\text{UPop}_{2\times}$ + SmartTrim	$4\times$	80.52	80.85	76.0

Table 3: Results of adopting the static acceleration model UPop as the backbone. We also provide the target acceleration ratio for each model.

our SmartTrim focuses on more fine-grained units and delivers promising results even when applied at higher acceleration ratios. In addition, SmartTrim achieves competitive performance compared to pretrained accelerated VLMs, further illustrating that our method is more economical.

Efficiency-Performance Trade-offs Figure 4 presents a Pareto front of efficiency-performance trade-offs of acceleration methods on NLVR2. We observe that SmartTrim consistently outperforms other acceleration methods, especially at higher ratios (~$3.0\times$). Surprisingly, SmartTrim performs even better than the original models with 21%~35% reduction in FLOPs, enjoying a "free lunch" in acceleration. We further evaluate the latency of METER, FTKD, TRIPS, and SmartTrim on the VQA dataset. The models are evaluated under the single-instance inference setting on the same CPU. The results are shown in Figure 5. We find that SmartTrim is significantly faster than the original model. Overall, SmartTrim achieves superior efficiency-performance trade-offs compared to the original models and previous acceleration methods.

Combining with Static Acceleration Approaches The proposed SmartTrim is orthogonal to static acceleration approaches. For further validation, we employ our approach on the static compressed model UPop, which statically prunes the parameters of the attention and FFN layers and achieves previous state-of-the-art performance on BLIP. The training recipe for SmartTrim is easily augmented to UPop without changing the original fine-tuning process. We utilize the UPop with the acceleration ratio $2\times$ as the backbone, and the results are presented in Table 3. Comparing with $\text{UPop}_{2\times}$, we observe that SmartTrim can preserve over 99% performance while enjoying faster inference. This indicates that our adaptive pruning can effectively complement static acceleration approaches to achieve faster inference and smaller sizes for VLMs. Moreover, SmartTrim significantly outperforms $\text{UPop}_{4\times}$, suggesting that combining SmartTrim with a static compression model may be better than directly training a smaller compression model, especially when aiming for higher speedup ratios.

Models	Image Resolution	VQA test-dev	FLOPs(G)
METER	288^2	76.78	48.3
SmartTrim	288^2	76.44	26.2
METER	384^2	77.43	88.5
SmartTrim	384^2	77.13	46.0

Table 4: Results of models fine-tuned with different image resolutions on the VQA dataset.

Fine-tuning with different resolutions Table 4 shows the VQA results of METER and SmartTrim on images of varying resolutions. Our approach reduces the computational overhead of the original model, while maintaining performance on input images of different resolutions. On METER models, increasing resolution improves results, but sacrifices efficiency, which poses a challenge in utilizing higher resolutions. However, at higher resolution (384^2), SmartTrim retains performance while being even faster than METER with lower resolution (288^2), suggesting that SmartTrim can effectively encode images of higher resolution to improve performance while minimizing computational demands.

5. Analysis

In this section, we conduct extensive experiments to analyze SmartTrim. All experiments are conducted on the METER backbone.

5.1. Ablation Study

Effect of Adaptive Trimmers We first investigate the effect of our adaptive pruning trimmers. For

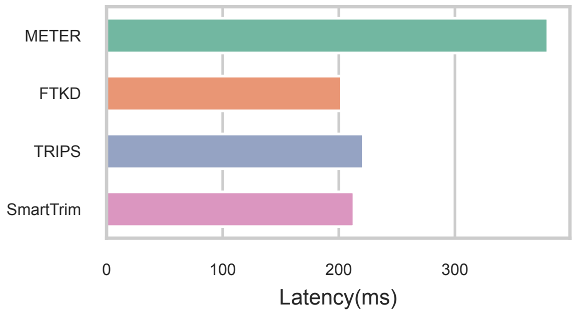

Figure 5: Averaged latency on the VQA dataset.

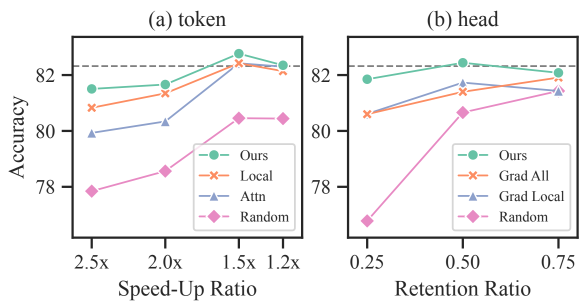

Figure 6: Comparison between different token (left) and head (right) pruning approaches on NLVR2. The dashed line denotes the performance of the original model.

simplicity, we only consider the pruning in cross-modal encoder. ❶ For *token pruning*, we consider a variant of adaptive pruning without cross-modal guidance (*Local*). Besides, we also include static pruning baselines: random pruning (*Random*) and attention score-based pruning (*Attn*; Jiang et al. (2022)). We present the NLVR2 performance trend with different speed-up ratios in Figure 6(a). We find that both adaptive pruning methods outperform static pruning methods at various ratios. Moreover, incorporating information from cross-modal interactions consistently improves performance, suggesting that cross-modal semantic guidance is critical to identifying more relevant tokens in different modalities. ❷ For *head pruning*, we compare with random pruning (*Random*), and gradient-based pruning variants (Michel et al., 2019) including retaining top-p heads in each module (*Grad Local*) or in the whole model (*Grad All*). As shown in Figure 6(b), our method significantly outperforms other baselines, especially in the low retention ratio regime ($0.25\times$), demonstrating the effectiveness of the proposed learned-based adaptive pruning mechanism. Another interesting phenomenon is that a slight pruning of tokens and heads can improve performance, which can be seen as a "free lunch" of sparsity and also presented in BERT (Hao et al., 2021) or ViT pruning (Chen et al., 2021).

Models	NLVR2 dev	NLVR2 test-P	VQA test-dev
SmartTrim$_{1.5\times}$	**81.89**	**82.72**	**77.25**
- Self-Distillation	81.58	82.50	77.06
- Curriculum Training	81.70	82.52	77.00
SmartTrim$_{2.0\times}$	**82.02**	81.97	**77.13**
- Self-Distillation	81.35	81.67	76.77
- Curriculum Training	81.58	**82.01**	76.35
SmartTrim$_{2.5\times}$	**81.18**	**81.55**	**76.60**
- Self-Distillation	80.51	81.30	75.79
- Curriculum Training	78.62	79.97	75.33

Table 5: Ablation studies of training strategies. Results are averaged over 3 runs.

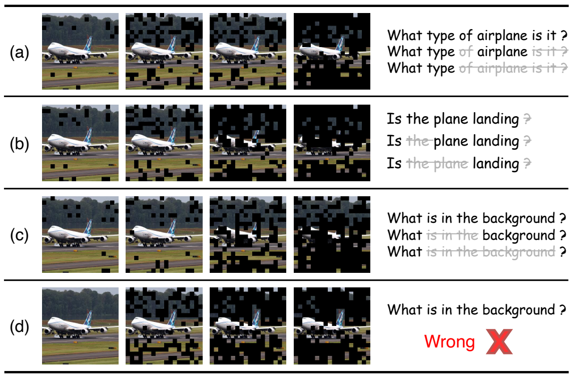

Figure 7: The visualizations of token trimming process on VQA. Image process order is shown from left to right and text is from top to bottom. (a)-(c) are obtained by our proposed XModal-aware token trimmer. (d) is from the local baseline that **without** cross-modal guidance, which finally yields a wrong answer.

Impact of Training Strategies We then analyze the impact of the proposed training strategies of SmartTrim. As shown in Table 5, we compare the proposed SmartTrim with variants without self-distillation or curriculum training on the NLVR2 and VQA datasets. From the results, we observe that both strategies improve performance at various acceleration ratios. At higher acceleration ratios, these strategies make training more stable, leading to a dramatic improvement.

5.2. Qualitative Analysis

Visualization of Token Trimming We visualize the token trimming procedure in Figure 7: (a)-(c) are from our XModel-aware token trimmer in SmartTrim while (d) is from the baseline without cross-modal guidance (*Local*). We observe that the XModal-aware trimmer gradually eliminates redundant tokens and finally focuses on informative ones. With the same input image, it can effectively identify

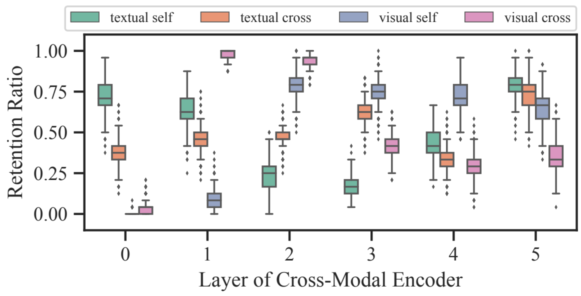

Figure 8: The head retention distribution of the model with 50% target budget.

patches relevant to different questions, thereby giving correct answers. However, the local baseline (Figure 7 (d)) only keeps the subject of the image (*plane*) but is irrelevant to the questions. See more results in Appendix D.

Distribution of Retained Attention Heads Figure 8 shows the distribution of the retention attention heads in SmartTrim with an overall target budget ratio of 50%. We observe significant variations in retention heads between different instances, and SmartTrim learns distinct trimming strategies for different attention modules.

Adaptive Computational Patterns We further analyze the computational distribution of SmartTrim to investigate adaptive patterns. We use a model with targeting on a 2 times acceleration budget [4] and show the visualization in Figure 1. As shown in Figure 1, we observe that SmartTrim can achieve an acceleration ranging from $1.5\times$ to $2.7\times$ on various instances. Furthermore, it learns to allocate more computations to instances that require complex cross-modal interactions and less to simple ones. These findings indicate that SmartTrim can adaptively allocate computational overhead across diverse inputs.

6. Related Work

6.1. Vision-Language Models

The Transformer-based vision-language model (VLM) has emerged as a dominant architecture for various vision-language tasks (Radford et al., 2021; Kim et al., 2021; Li et al., 2021; Bao et al., 2022; Wang et al., 2022b; Yu et al., 2022; Zeng et al., 2022; Xu et al., 2023; Li et al., 2023). Although they achieve satisfactory performance, the extensive amount of parameters inflicts an extravagant computational burden, impeding their scalability and application in the production environment.

[4]The resolution of input images is 288^2.

6.2. Transformer Acceleration

Extensive research aims at accelerating Transformer, which can be categorized into two streams: *Static* and *Adaptive* approaches (Xu et al., 2021).

Static Approaches yield accelerated models that remain static for all instances during inference after deployment. Prior work effectively accelerates uni-modal Transformers through various techniques, such as knowledge distillation (Hinton et al., 2015; Sanh et al., 2019; Sun et al., 2019; Jiao et al., 2020; Xu et al., 2020; Wang et al., 2020b), parameter pruning (Han et al., 2015; Michel et al., 2019; Wang et al., 2020c; Sanh et al., 2020; Hou et al., 2020; Fan et al., 2020; Xia et al., 2022), and static token reduction via pruning (Goyal et al., 2020; Chen et al., 2021; Rao et al., 2021; Tang et al., 2022; Liang et al., 2022; Xu et al., 2022) or merging (Ryoo et al., 2021; Bolya et al., 2023) less relevant tokens. Recently, a few static methods dedicated to VLMs have been proposed (Wang et al., 2020a, 2022c; Fang et al., 2021; Gan et al., 2022). EfficientVLM (Wang et al., 2023a) is trained under a framework of pre-training distillation followed by pruning. Shi et al. (2023) introduces a progressive search-and-prune method, which needs retraining to sustain performance. TRIPS (Jiang et al., 2022) proposes to eliminate visual tokens using textual information by pre-training, while they only focus on token reduction in the visual encoder and keep trimming ratios static for all instances. These methods require pre-training or iterative retraining to retain performance while being computationally expensive. Cao et al. (2023) introduces static token pruning and merging within the VLM cross-modal encoder. Overall, static acceleration fixes architecture regardless of large variations in the complexity of instances, limiting the capability of models.

Adaptive Approaches enable accelerated models to adjust the computation required based on inputs dynamically. Early exiting strategy has been applied to accelerate uni-modal Transformers by terminating inference at an early layer (Xin et al., 2020; Zhou et al., 2020). Another stream is adaptive token pruning (Ye et al., 2021; Pan et al., 2021; Kim et al., 2022; Guan et al., 2022; Yin et al., 2022; Meng et al., 2022; Kong et al., 2022; Zhou et al., 2023), which uses a policy network to gradually eliminate redundant tokens on a per-instance basis. However, employing these uni-modal approaches directly in multimodal scenarios is suboptimal, as they overlook the importance of cross-modal interactions. Tang et al. (2023) applies the early exiting technique based on layerwise similarities for an encoder-decoder-based VLM. However, the constraint of pruning all tokens at the same layer is aggressive, resulting in significant performance degradation on challenge VL tasks, as shown in our experiments. In contrast, SMARTTRIM focus on more fine-grained pruning units: token and attention heads, to achieve a better performance-efficiency trade-off.

7. Conclusion

In this work, we present SMARTTRIM, an adaptive pruning framework for efficient VLMs that dynamically adjusts the computation overhead in an input-dependent manner. By integrating token and head trimmers along with the backbone, SMARTTRIM prunes redundant tokens and heads during runtime based on the cross-modal information guidance and the pre-given budget. Extensive experiments across various architectures and datasets show that SMARTTRIM achieves better efficiency-performance trade-offs. We hope our endeavor will benefit end users by making multimodal systems more accessible.

Acknowledgements

We thank anonymous reviewers for their insightful feedback that helped improve the paper. The research is supported by the National Key Research and Development Project (2021YFF0901602), the National Science Foundation of China (U22B2059, 62276083), and Shenzhen Foundational Research Funding (JCYJ20200109113441941), Major Key Project of PCL (PCL2021A06). The first two authors are core contributors of the work. Ming Liu is the corresponding author.

8. Bibliographical References

Harsh Agrawal, Peter Anderson, Karan Desai, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. nocaps: novel object captioning at scale. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 8947–8956. IEEE.

Hangbo Bao, Wenhui Wang, Li Dong, Qiang Liu, Owais Khan Mohammed, Kriti Aggarwal, Subhojit Som, Songhao Piao, and Furu Wei. 2022. Vlmo: Unified vision-language pre-training with mixture-of-modality-experts. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, volume 382 of *ACM International Conference Proceeding Series*, pages 41–48. ACM.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. 2023. Token merging: Your vit but faster. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Qingqing Cao, Bhargavi Paranjape, and Hannaneh Hajishirzi. 2023. Pumer: Pruning and merging tokens for efficient vision language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 12890–12903. Association for Computational Linguistics.

Tianlong Chen, Yu Cheng, Zhe Gan, Lu Yuan, Lei Zhang, and Zhangyang Wang. 2021. Chasing sparsity in vision transformers: An end-to-end exploration. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 19974–19988.

Xi Chen, Xiao Wang, Soravit Changpinyo, A. J. Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Nan Ding, Keran Rong, Hassan Akbari, Gaurav Mishra, Linting Xue, Ashish V. Thapliyal, James Bradbury, and Weicheng Kuo. 2023. Pali: A jointly-scaled multilingual language-image model. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Zi-Yi Dou, Yichong Xu, Zhe Gan, Jianfeng Wang, Shuohang Wang, Lijuan Wang, Chenguang Zhu, Pengchuan Zhang, Lu Yuan, Nanyun Peng, Zicheng Liu, and Michael Zeng. 2022. An empirical study of training end-to-end vision-and-language transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 18145–18155. IEEE.

Angela Fan, Edouard Grave, and Armand Joulin. 2020. Reducing transformer depth on demand with structured dropout. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Zhiyuan Fang, Jianfeng Wang, Xiaowei Hu, Lijuan Wang, Yezhou Yang, and Zicheng Liu. 2021. Compressing visual-linguistic model via knowledge distillation. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 1408–1418. IEEE.

Zhe Gan, Yen-Chun Chen, Linjie Li, Tianlong Chen, Yu Cheng, Shuohang Wang, Jingjing Liu, Lijuan Wang, and Zicheng Liu. 2022. Playing lottery tickets with vision and language. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 652–660. AAAI Press.

Saurabh Goyal, Anamitra Roy Choudhury, Saurabh Raje, Venkatesan T. Chakaravarthy, Yogish Sabharwal, and Ashish Verma. 2020. Power-bert: Accelerating BERT inference via progressive word-vector elimination. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 3690–3699. PMLR.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 6325–6334. IEEE Computer Society.

Yue Guan, Zhengyi Li, Jingwen Leng, Zhouhan Lin, and Minyi Guo. 2022. Transkimmer: Transformer learns to layer-wise skim. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 7275–7286. Association for Computational Linguistics.

Song Han, Jeff Pool, John Tran, and William J. Dally. 2015. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 1135–1143.

Yaru Hao, Li Dong, Furu Wei, and Ke Xu. 2021. Self-attention attribution: Interpreting information interactions inside transformer. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 12963–12971. AAAI Press.

Dan Hendrycks and Kevin Gimpel. 2016. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *CoRR*, abs/1606.08415.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531.

Lu Hou, Zhiqi Huang, Lifeng Shang, Xin Jiang, Xiao Chen, and Qun Liu. 2020. Dynabert: Dynamic BERT with adaptive width and depth. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Chaoya Jiang, Haiyang Xu, Chenliang Li, Ming Yan, Wei Ye, Shikun Zhang, Bin Bi, and Songfang Huang. 2022. TRIPS: efficient vision-and-language pre-training with text-relevant image patch selection. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 4084–4096. Association for Computational Linguistics.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. Tinybert: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 4163–4174. Association for Computational Linguistics.

Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 3128–3137. IEEE Computer Society.

Yigitcan Kaya, Sanghyun Hong, and Tudor Dumitras. 2019. Shallow-deep networks: Understanding and mitigating network overthinking. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 3301–3310. PMLR.

Sehoon Kim, Sheng Shen, David Thorsley, Amir Gholami, Woosuk Kwon, Joseph Hassoun, and Kurt Keutzer. 2022. Learned token pruning for transformers. In *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, pages 784–794. ACM.

Wonjae Kim, Bokyung Son, and Ildoo Kim. 2021. Vilt: Vision-and-language transformer without convolution or region supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5583–5594. PMLR.

Zhenglun Kong, Peiyan Dong, Xiaolong Ma, Xin Meng, Wei Niu, Mengshu Sun, Xuan Shen, Geng Yuan, Bin Ren, Hao Tang, Minghai Qin, and Yanzhi Wang. 2022. Spvit: Enabling faster vision transformers via latency-aware soft token pruning. In *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XI*, volume 13671 of *Lecture Notes in Computer Science*, pages 620–640. Springer.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. 2023. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 19730–19742. PMLR.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi. 2022. BLIP: bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 12888–12900. PMLR.

Junnan Li, Ramprasaath R. Selvaraju, Akhilesh Gotmare, Shafiq R. Joty, Caiming Xiong, and Steven Chu-Hong Hoi. 2021. Align before fuse: Vision and language representation learning with momentum distillation. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 9694–9705.

Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. 2022. Evit: Expediting vision transformers via token reorganizations. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: common objects in context. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, volume 8693 of *Lecture Notes in Computer Science*, pages 740–755. Springer.

Weijie Liu, Peng Zhou, Zhiruo Wang, Zhe Zhao, Haotang Deng, and Qi Ju. 2020. Fastbert: a self-distilling BERT with adaptive inference time. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6035–6044. Association for Computational Linguistics.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13–23.

Lingchen Meng, Hengduo Li, Bor-Chun Chen, Shiyi Lan, Zuxuan Wu, Yu-Gang Jiang, and Ser-Nam Lim. 2022. Adavit: Adaptive vision transformers for efficient image recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 12299–12308. IEEE.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 14014–14024.

Ali Modarressi, Hosein Mohebbi, and Mohammad Taher Pilehvar. 2022. Adapler: Speeding up inference by adaptive length reduction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 1–15. Association for Computational Linguistics.

Bowen Pan, Rameswar Panda, Yifan Jiang, Zhangyang Wang, Rogério Feris, and Aude Oliva.

2021. Ia-red^2: Interpretability-aware redundancy reduction for vision transformers. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 24898–24911.

Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. 2015. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 2641–2649. IEEE Computer Society.

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. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. 2021. Dynamicvit: Efficient vision transformers with dynamic token sparsification. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 13937–13949.

Michael S. Ryoo, A. J. Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. 2021. Tokenlearner: Adaptive space-time tokenization for videos. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 12786–12797.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Victor Sanh, Thomas Wolf, and Alexander M. Rush. 2020. Movement pruning: Adaptive sparsity by fine-tuning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Dachuan Shi, Chaofan Tao, Ying Jin, Zhendong Yang, Chun Yuan, and Jiaqi Wang. 2023. Upop: Unified and progressive pruning for compressing vision-language transformers. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 31292–31311. PMLR.

Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. 2019. A corpus for reasoning about natural language grounded in photographs. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 6418–6428. Association for Computational Linguistics.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. Patient knowledge distillation for BERT model compression. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 4322–4331. Association for Computational Linguistics.

Shengkun Tang, Yaqing Wang, Zhenglun Kong, Tianchi Zhang, Yao Li, Caiwen Ding, Yanzhi Wang, Yi Liang, and Dongkuan Xu. 2023. You need multiple exiting: Dynamic early exiting for accelerating unified vision language model. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 10781–10791. IEEE.

Yehui Tang, Kai Han, Yunhe Wang, Chang Xu, Jianyuan Guo, Chao Xu, and Dacheng Tao. 2022. Patch slimming for efficient vision transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 12155–12164. IEEE.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Hanrui Wang, Zhekai Zhang, and Song Han. 2021. Spatten: Efficient sparse attention architecture with cascade token and head pruning. In *IEEE International Symposium on High-Performance Computer Architecture, HPCA 2021, Seoul, South Korea, February 27 - March 3, 2021*, pages 97–110. IEEE.

Jianfeng Wang, Xiaowei Hu, Pengchuan Zhang, Xiujun Li, Lijuan Wang, Lei Zhang, Jianfeng Gao, and Zicheng Liu. 2020a. Minivlm: A smaller and faster vision-language model. *CoRR*, abs/2012.06946.

Peihao Wang, Wenqing Zheng, Tianlong Chen, and Zhangyang Wang. 2022a. Anti-oversmoothing in deep vision transformers via the fourier domain analysis: From theory to practice. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. 2022b. OFA: unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 23318–23340. PMLR.

Tiannan Wang, Wangchunshu Zhou, Yan Zeng, and Xinsong Zhang. 2023a. Efficientvlm: Fast and accurate vision-language models via knowledge distillation and modal-adaptive pruning. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 13899–13913. Association for Computational Linguistics.

Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, and Furu Wei. 2023b. Image as a foreign language: BEIT pretraining for vision and vision-language tasks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 19175–19186. IEEE.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020b. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Zekun Wang, Wenhui Wang, Haichao Zhu, Ming Liu, Bing Qin, and Furu Wei. 2022c. Distilled dual-encoder model for vision-language understanding. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 8901–8913. Association for Computational Linguistics.

Ziheng Wang, Jeremy Wohlwend, and Tao Lei. 2020c. Structured pruning of large language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6151–6162. Association for Computational Linguistics.

Mengzhou Xia, Zexuan Zhong, and Danqi Chen. 2022. Structured pruning learns compact and accurate models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 1513–1528. Association for Computational Linguistics.

Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. 2019. Visual entailment: A novel task for fine-grained image understanding. *CoRR*, abs/1901.06706.

Ji Xin, Raphael Tang, Jaejun Lee, Yaoliang Yu, and Jimmy Lin. 2020. Deebert: Dynamic early exiting for accelerating BERT inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 2246–2251. Association for Computational Linguistics.

Canwen Xu, Wangchunshu Zhou, Tao Ge, Furu Wei, and Ming Zhou. 2020. Bert-of-theseus: Compressing BERT by progressive module replacing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 7859–7869. Association for Computational Linguistics.

Jingjing Xu, Wangchunshu Zhou, Zhiyi Fu, Hao Zhou, and Lei Li. 2021. A survey on green deep learning. *CoRR*, abs/2111.05193.

Xiao Xu, Chenfei Wu, Shachar Rosenman, Vasudev Lal, Wanxiang Che, and Nan Duan. 2023. Bridgetower: Building bridges between encoders in vision-language representation learning. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 10637–10647. AAAI Press.

Yifan Xu, Zhijie Zhang, Mengdan Zhang, Kekai Sheng, Ke Li, Weiming Dong, Liqing Zhang, Changsheng Xu, and Xing Sun. 2022. Evo-vit: Slow-fast token evolution for dynamic vision transformer. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth*

Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022, pages 2964–2972. AAAI Press.

Deming Ye, Yankai Lin, Yufei Huang, and Maosong Sun. 2021. TR-BERT: dynamic token reduction for accelerating BERT inference. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 5798–5809. Association for Computational Linguistics.

Hongxu Yin, Arash Vahdat, Jose M. Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. 2022. A-vit: Adaptive tokens for efficient vision transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 10799–10808. IEEE.

Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. 2022. Coca: Contrastive captioners are image-text foundation models. *Trans. Mach. Learn. Res.*, 2022.

Yan Zeng, Xinsong Zhang, and Hang Li. 2022. Multi-grained vision language pre-training: Aligning texts with visual concepts. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 25994–26009. PMLR.

Wangchunshu Zhou, Yuchen Eleanor Jiang, Ryan Cotterell, and Mrinmaya Sachan. 2023. Efficient prompting via dynamic in-context learning. *CoRR*, abs/2305.11170.

Wangchunshu Zhou, Canwen Xu, Tao Ge, Julian J. McAuley, Ke Xu, and Furu Wei. 2020. BERT loses patience: Fast and robust inference with early exit. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

A. Details of Similarity Calculation

To measure the redundancy in token representations and attention heads of VLMs, we calculate the average cosine similarity between token representations and attention maps at each layer following previous work (Goyal et al., 2020; Wang et al., 2022a).

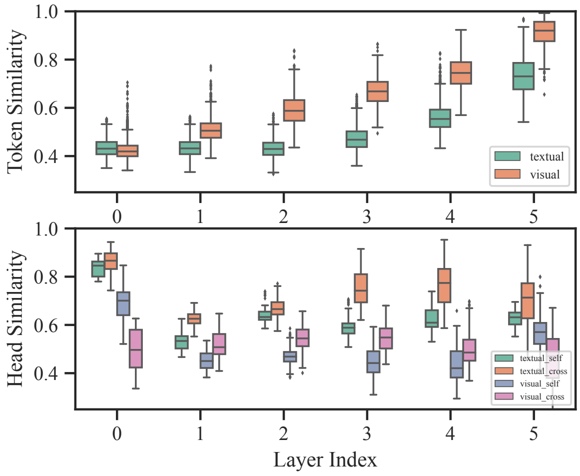

Figure 9: The similarity visualizations of the cross-modal encoder in METER fine-tuned on NLVR2.

Token Similarity Given the corresponding token representations $\boldsymbol{X} \in \mathbb{R}^{N \times D}$, the averaged token representations similarity is computed by:

$$S_T = \frac{2}{N(N-1)} \sum_{i=1}^{N} \sum_{j=i+1}^{N} \frac{\boldsymbol{X}_i \cdot \boldsymbol{X}_j}{\|\boldsymbol{X}_i\|_2 \, \|\boldsymbol{X}_j\|_2}$$

Head Similarity We use the similar metric to compute head similarity for attention maps. Given the attention map $\boldsymbol{A} \in \mathbb{R}^{H \times N \times N}$ with H heads, the averaged cosine similarity between different heads is calculated as:

$$S_A = \frac{2}{H(H-1)N} \sum_{i=1}^{H} \sum_{j=i+1}^{H} \sum_{k=1}^{N} \frac{\boldsymbol{A}_i^k \cdot \boldsymbol{A}_j^k}{\|\boldsymbol{A}_i^k\|_2 \|\boldsymbol{A}_j^k\|_2}$$

where $\boldsymbol{A}_i^k$ denotes the k-th token's attention distribution in the i-th head.

More Visualization We also present the visualizations of different modules in VLMs on NLVR2 and VQA tasks in Figures 9, 10, and 11. Similar to Figure 3, significant redundancy can be observed in both token representations and attention heads within the VLM modules on various tasks.

B. Details of Downstream Tasks

Natural Language for Visual Reasoning (NLVR2 (Suhr et al., 2019)) is a visual reasoning task that aims to determine whether a textual statement describes a pair of images. For METER-based models, we construct two pairs of image-text, each consisting of the image and a textual statement. For models based on BLIP, we directly feed the two images and the text to the encoder.

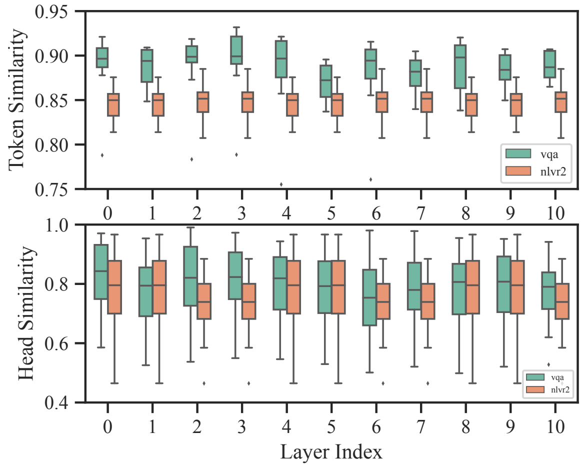

Figure 10: The similarity visualizations of the textual encoder in METER fine-tuned on VQA and NLVR2.

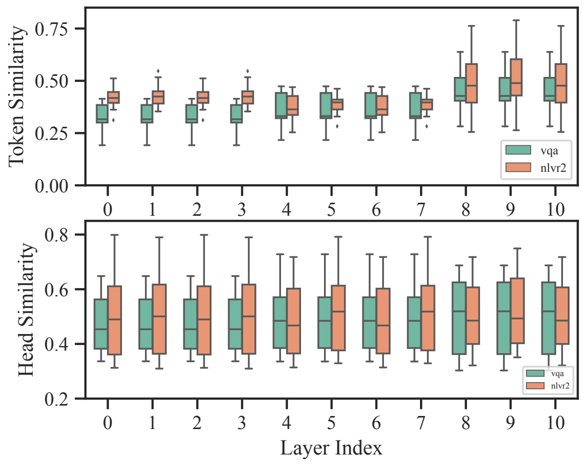

Figure 11: The similarity visualizations of the visual encoder in METER fine-tuned on VQA and NLVR2.

Visual Question Answering (VQA v2 (Goyal et al., 2017)) requires the model to answer questions based on the input image. For METER-based models, we formulate the problem as a classification task with 3,129 answer candidates. For BLIP-based models, we consider it as an answer generation task and use the decoder to rank the candidate answers during inference.

Visual Entailment (SNLI-VE (Xie et al., 2019)) is a three-way classification dataset, aiming to predict the relationship between an image and a text hypothesis: *entailment*, *natural*, and *contradiction*.

Image-Text Retrieval (ITR) We evaluate image-to-text retrieval (TR) and text-to-image retrieval (IR) on Flickr30K (Plummer et al., 2015) with the standard split (Karpathy and Fei-Fei, 2015).

Hyperparameters	NLVR2	VQAv2	SNLI-VE	Flickr30K
Epochs	10	10	5	10
Batch Size	256	512	64	512
Initial Learning Rate	1e-5	5e-6	2e-6	5e-6
Learning Rate Decay	Linear Scheduler			
Dropout	0.1			
Weight Decay	0.01			
Warmup Ratio	0.1			
AdamW β	(0.9, 0.999)			
Data Augmentation	RandomAugment			
Image Resolution	288^2			

Table 6: Hyperparameters for fine-tuning SMARTTRIM-METER on various downstream VL tasks.

Hyperparameters	NLVR2	VQAv2	Captioning
Epochs	15	10	5
Batch Size	256		
Initial Learning Rate	3e-5	2e-5	1e-5
Learning Rate Decay	Cosine Scheduler		
Weight Decay	0.05		
AdamW β	(0.9, 0.999)		
Data Augmentation	RandomAugment		
Image Resolution	384^2	480^2	384^2

Table 7: Hyperparameters for fine-tuning SMARTTRIM-BLIP on various downstream VL tasks.

Image Captioning The image is given to the encoder and the decoder will generate the corresponding caption with a text prompt "a picture of" following Li et al. (2022). In this work, we optimize only the cross-entropy loss during fine-tuning. Our experiments are conducted on COCO (Lin et al., 2014), and the evaluation is performed on both the COCO test set and the NoCaps (Agrawal et al., 2019) validation set (zero-shot transfer).

C. Implementation Details

C.1. Hyperparameter Settings

The MLP network in our token and head trimmers consists of two linear layers with GeLU activation (Hendrycks and Gimpel, 2016). To reduce the computations, we set $D' = D/12$. Fine-tuning hyperparameters on METER are given in Table 6, mainly following the defaults in Dou et al. (2022). Fine-tuning hyperparameters on BLIP are given in Table 7, mainly following the defaults in Li et al. (2022). We perform token adaptive pruning in the visual encoder/cross-modal encoder and head adaptive pruning in the cross-modal encoder. For efficiency evaluation, we use *torchprofile* to measure FLOPs. As for the latency, we evaluate on an Intel Xeon E5-466 2640 v4 CPU.

C.2. Details of Re-implemented Baselines

For FTKD, we initiate the student model following Sun et al. (2019) to directly use the first k layers of the original model ($k \in \{4, 6\}$ for the visual encoder, $k \in \{2, 3\}$ for the cross-modal encoder). In our experiments, we find that this initialization strategy is considerably better than the other methods. Then, we fine-tune the student model by logit/hidden representation/attention distillation objectives the same as Jiao et al. (2020). For MuE, we fine-tune the METER according to Tang et al. (2023), and perform grid search from 0.85 to 0.99, an interval of 0.01, for the similarity thresholds of the visual and cross-modal encoder. For TRIPS, we follow the original setting in Jiang et al. (2022) to fine-tune the METER backbone. We exhaustively search for optimal settings and hyperparameters for the re-implemented baselines.

C.3. Details of Baselines for Trimming Ablation

Here we provide details of baselines in the trimming ablation.

Token Trimming For the *local* baseline, we remove the cross-modal awareness score when calculating the token importance. The *random* baseline randomly prunes tokens during both training and inference. Following previous work (Goyal et al., 2020; Liang et al., 2022; Jiang et al., 2022), the *Attn* baseline adopts the token attention value as the importance score and uses top-*k* operation to select retained tokens, discarding the remaining ones. For a fair comparison, we ensure that all baselines incur the same computational overhead as our method. In addition, we conduct an exhaustive search to determine the optimal hyperparameters for each baseline. This meticulous approach ensures the comparability of our method with other methods.

Head Trimming For a given retention ratio $p\%$, the random baseline randomly retains $p\%$ of heads in each attention module. Gradient-based head pruning (Michel et al., 2019) first computes loss on pseudo-labels and then prunes attention heads with the importance score obtained by Taylor expansion. With given input x, importance score of head h is defined as:

$$\boldsymbol{I}_h = E_x \left| A_h^T \frac{\partial \mathcal{L}(x)}{\partial A_h} \right|$$

Where $\mathcal{L}$ is the loss function, and A_h is the context layer of head h. For the gradient-based baseline, we introduce two variants: (1) *Grad Local*, which retains the top-$p\%$ heads in each attention module, (2) *Grad All*, which maintains the top-$p\%$ heads of the entire model. We apply these methods on the METER cross-modal encoder.

D. More Visualization Examples of Token Trimming

To demonstrate the ability to understand cross-modal interactions of our approach, we show more visualization results of our XModal-aware token trimmer in Figure 12. We can see that the final retained image patches are highly relevant to the textual questions. The question words (e.g., *what*) are critical in VQA because they are highly correlated with the category (numbers, yes/no or others) of correct answers. Therefore, we observe that function words (e.g., *of,the*) are gradually removed while critical tokens such as question words are retained.

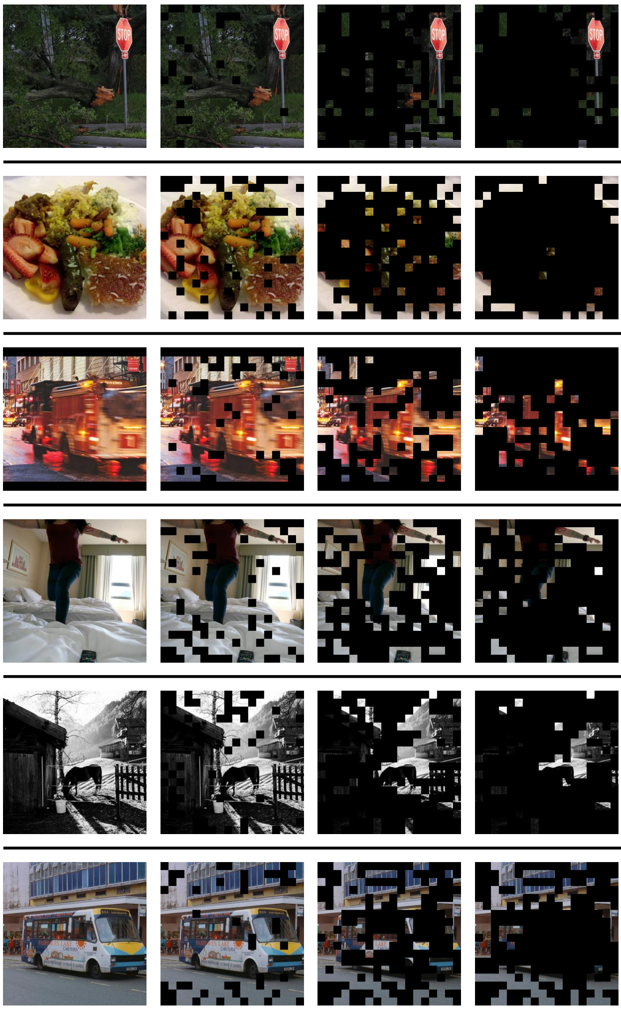

Figure 12: More visualization results by SmartTrim.