Collaborative Multi-Mode Pruning for Vision-Language Models

Supplementary Material

In this document, we additionally provide the overall algorithm of our CoMP method in Sec. A, detailed descriptions of the datasets and evaluation metrics in Sec. B, more implementation details in Sec. C, detailed description of dimension mapping for parameter pruning in Sec. D, and additional experimental results in Sec. E. Specifically, we evaluate real-world inference latency in Sec. E.1, demonstrate contributions of difference modes in Sec. E.2, include more ablations on hyperparameters in Sec. E.3, multi-seed statistical results in Sec. E.4, video-text performance in Sec. E.5, further validation of orthogonality to single-mode pruning in Sec. E.6 and discussion on computational overhead in Sec. E.7. Besides, we provide supplementary observations on the inconsistency between parameter and token importance in Sec. E.8.

A. Overall Algorithm of CoMP

The overall workflow of our proposed CoMP is summarized in Algorithm 1. Lines $13 \sim 30$ describe the pipeline of inner loop as shown in Fig. 3a, pruning parameters and tokens based on importance scores calculated by the CIM module. Lines $8 \sim 44$ depict the pipeline of outer loop as displayed in Fig. 3a, by periodically shifting pruning modes with random exploration and historical information. Notably, $\mathcal{I}$ training steps are performed between mode shifting. In the parameter pruning mode, this interval is evenly divided into six sub-intervals $\mathcal{J} = \mathcal{I}/6$. For each of the first five sub-intervals, we accumulate parameter importance scores over the $\mathcal{J}$ steps to ensure training stability, and then uniformly decay the mask values M^p from 1 to 0, by following the UPop [16] framework. In the final $\mathcal{J}$ steps, only parameter updates are performed to enable model recovery.

B. Detailed Description of Datasets and Evaluation Metrics

For task-specific VLMs, we primarily evaluate our proposed method on four widely used public datasets, including NLVR2 [18], COCO [10], Flickr30K [25] and VQAv2 [6]. We also include experiments on the MSR-VTT [23] dataset in this document. For LLaVA, we further assess methods on six widely adopted image understanding benchmarks, including MME [5], MMBench [12], GQA [7], TextVQA [17], VQAv2 [6] and POPE [9]. In addition, the pruning is conducted through supervised fine-tuning (SFT) on the official 665K instruction data [11] (denoted as Mix665K).

NLVR2. The NLVR2 [18] dataset is designed for the advancement of joint reasoning over natural language and images. It contains 29,680 sentences and 127,502 real-world images, which are combined to form 107,292 examples. Each example consists of a caption text paired with two images, and the task is to determine whether the caption accurately describes both images. The overall examples are divided into training, development, public test and unreleased test sets, which contain 86,373, 6,982, 6,967 and 6,970 examples, respectively. We report the accuracy on the development set (Dev. Acc.) and the public test set (Test Acc.) as the evaluation metric.

COCO. The COCO [10] dataset is a widely used benchmark that facilitates multiple tasks by promoting the scene understanding capability. Each image is annotated with 5 captions. In our work, by following [8], we adopt the Karpathy split for both image-text retrieval and image captioning tasks, *i.e.* 113K, 5K and 5K images for training, validation and test, respectively. In the image-text retrieval task, as the goal is to retrieve the most relevant image/text given an input query, we report the top-1/-5 recall (R@1/5) as the evaluation metric. In the image captioning task, as this task aims to generate a descriptive sentence that accurately reflects the visual content of a given image, we report CIDEr [19] and SPICE [1] as the evaluation metrics. Concretely, CIDEr [19] evaluates the quality of the generated captions by measuring their consensus with multiple reference sentences. SPICE [1] simulates the human judgment process by comparing the semantic propositional content.

Flickr30K. The Flickr30K [25] dataset is originally designed for visual denotation. We also adopt the Karpathy split on the image-text retrieval task and report the top-1/-5 recall (R@1/5) metrics for evaluation. The training, validation and test sets consist of 29K, 1K and 1K images, respectively, where each image has 5 captions.

VQAv2. The VQAv2 [6] dataset is curated for the visual question answering task. Given an image and a question, this task encourages the model to jointly understand and reason over all information and generate a short answer. For the BLIP model, by following [8], we adopt the split consisting of 123K images and 658K questions for training, and 81K images and 448K questions for testing. We report accuracy on both the development (Test-dev) and standard (Test-std) splits of the test set. For the LLaVA model, by following [11], we conduct zero-shot evaluation and report accuracy on the development split of the test set. All metrics are obtained through the official evaluation website.

MSR-VTT. The MSR-VTT [23] dataset is a large-scale video description benchmark with comprehensive categories

Algorithm 1: Collaborative Multi-Mode Pruning (CoMP)

Input: Uncompressed VLM $\mathcal{F}(\cdot|\boldsymbol{W})$ in full parameters $\boldsymbol{W}$ with L layers, loss function $\mathcal{L}$, target FLOPs TF, pruning modes $\mathcal{B}$, probability for exploration ρ, interval steps $\mathcal{I}$ between mode shifting, training dataset $\mathcal{D}_{\text{train}}$, and validation dataset $\mathcal{D}_{\text{val}}$.

Output: Pruned model $\mathcal{F}(\cdot|\hat{\boldsymbol{W}}, \hat{\theta}_v^t, \hat{\theta}_l^t)$ with parameters $\hat{\boldsymbol{W}}$ and thresholds $\hat{\theta}_v^t, \hat{\theta}_l^t$ for token pruning.

Initialize model with parameter masks $\boldsymbol{M}^p$ and token thresholds θ_v^t, θ_l^t for pruning
$\mathcal{F} \leftarrow \mathcal{F}(\cdot|\boldsymbol{W}, \boldsymbol{M}^p, \theta_v^t, \theta_l^t)$
$\boldsymbol{M}^p \leftarrow \mathbf{1}$ # Initialize parameter pruning masks to 0
$\mathcal{J} \leftarrow \mathcal{I}/6$ # Interval between parameter pruning mask updates
Initialize the MPS module. Elements of Θ correspond to $\theta_v^p, \theta_l^p, \theta_c^p, \theta_v^t, \theta_l^t$ in order
$\Theta \leftarrow \{0,0,0,0,0\}, \mathcal{R} \leftarrow \{0,0,0,0,0\}, \mathcal{T} \leftarrow \{0,0,0,0,0\}, m \leftarrow 0, T \leftarrow 0$
$CA,\ CF \leftarrow \text{Evaluate}(\mathcal{F}, \mathcal{D}_{\text{val}})$ # Initialize accuracy CA and FLOPs CF
while $CF > TF$ **do**
  $\boldsymbol{S}'^p \leftarrow 0$ # Reset the accumulated parameter importance
  # Conduct the current pruning mode $\mathcal{B}_m$ by adjusting the threshold
  $\Theta_m \leftarrow \Theta_m + \Delta\Theta_m, T \leftarrow T + 1$
  # Optimize the model for $\mathcal{I}$ steps at current stage
  for $i \leftarrow 1$ **to** $\mathcal{I}$ **do**
    $(\boldsymbol{X}^0, Y) \leftarrow \text{Sample}(\mathcal{D}_{\text{train}})$ # Sample data from the training set
    for $l \leftarrow 1$ **to** L **do**
      # Run layer-wise forward propagation with partially masked parameters
      $\boldsymbol{X}^l \leftarrow \mathcal{F}^l(\boldsymbol{X}^{l-1}|\boldsymbol{W}^l, \boldsymbol{M}^p)$
      # Calculate the self-corrected token importance by CIM
      $\boldsymbol{S}^{t,l} \leftarrow \text{CIT}(\boldsymbol{X}^l)$
      # Perform layer-wise token pruning with $\theta^t \in \{\theta_v^t, \theta_l^t\}$
      $\boldsymbol{M}^{t,l} \leftarrow \mathbb{I}(\boldsymbol{S}^{t,l} > \theta^t), \boldsymbol{X}^l \leftarrow \boldsymbol{X}^l \odot \boldsymbol{M}^{t,l}$
    if $\mathcal{B}_m \in \{B_v^p, B_l^p, B_c^p\}$ **then**
      # Follow the pruning framework in [16]: accumulate importance every $\mathcal{J}$ steps and uniformly decay the mask to 0 for stability.
      $\boldsymbol{S}'^p \leftarrow \boldsymbol{S}'^p + \text{CIP}(\boldsymbol{S}^t, \boldsymbol{W}, \boldsymbol{X})$ # Calculate token-weighted parameter importance by CIM
      if $i \,\%\, \mathcal{J} = 0$ **then**
        $\boldsymbol{M}^p \leftarrow \mathbb{I}(\boldsymbol{S}'^p > \theta^p)$ # Select parameters to prune with $\theta^p \in \{\theta_v^p, \theta_l^p, \theta_c^p\}$
        # Decay the mask to 0 in $5\mathcal{J}$ steps; the last $\mathcal{J}$ steps perform recovery only
        $\boldsymbol{M}^p \leftarrow \boldsymbol{M}^p + (1 - i/(5\mathcal{J}))(1 - \boldsymbol{M}^p)$
        $\boldsymbol{S}'^p \leftarrow 0$ # Reset the accumulated parameter importance
    $\boldsymbol{W} \leftarrow \boldsymbol{W} - \nabla_{\boldsymbol{W}} \mathcal{L}(\boldsymbol{X}^L, Y)$ # Calculate loss and update parameters
  $LA \leftarrow CA, LF \leftarrow CF$ # Record accuracy and FLOPs at previous stage
  $CA,\ CF \leftarrow \text{Evaluate}(\mathcal{F}, \mathcal{D}_{\text{val}})$ # Re-evaluate current accuracy and FLOPs
  # Calculate the pruning cost at current stage
  $\Delta val\_acc \leftarrow LA - CA, \Delta FLOPs \leftarrow LF - CF,\ r \leftarrow \Delta val\_acc/\Delta FLOPs$
  $I_m \leftarrow T - \mathcal{T}_m$ # Get the interval since last execution of $\mathcal{B}_m$
  $\mathcal{R}_m \leftarrow \text{Get\_cost\_with\_history}(\mathcal{R}_m, r, I_m)$ # Update pruning cost with Eq. (10)
  $\mathcal{T}_m \leftarrow T$ # Update the pruning stage where $\mathcal{B}_m$ is last performed
  # Shift mode by random exploration and historical information
  if $\text{Sample\_from\_uniform}() < \rho$ **then**
    # Select a pruning mode randomly according to Eq. (9)
    $\rho_0, \rho_1, \rho_2, \rho_3, \rho_4 \leftarrow \text{Get\_probability}(T, \mathcal{T})$
    $m \leftarrow \text{Random\_choice}(\text{Indices}(\mathcal{B}), \text{prob} = \{\rho_0, \rho_1, \rho_2, \rho_3, \rho_4\})$
  else
    $m \leftarrow \arg\min(\mathcal{R})$ # Select a priority pruning mode according to pruning cost
$\hat{\boldsymbol{W}} \leftarrow \boldsymbol{W} \odot \boldsymbol{M}^p$ # Completely prune parameters according to the parameter mask
$\hat{\theta}_v^t \leftarrow \Theta_3, \hat{\theta}_l^t \leftarrow \Theta_4$
return $\mathcal{F}(\cdot|\hat{\boldsymbol{W}}, \hat{\theta}_v^t, \hat{\theta}_l^t)$

Table A. Training and testing configurations for pruning BLIP, CLIP and LLaVA models on various vision-language tasks. 'Reasoning', 'Retrieval', 'Captioning', 'VQA', 'Und' denote the visual reasoning task, the image-text retrieval task, the image captioning task, the visual question answering task and the image understanding task, respectively. $R_{I\to T}$ and $R_{T\to I}$ denote the recall for image-to-text and text-to-image retrieval, respectively. $-$Val_Loss denotes the negative of the loss on validation set. '-' indicates that the setting is not applied to the corresponding task.

Configurations	BLIP-Reasoning	BLIP-Retrieval		BLIP-Captioning	BLIP-VQA	CLIP-Retrieval		LLaVA-Und
	NLVR2 [18]	COCO [10]	Flickr30K [25]	COCO [10]	VQAv2 [6]	COCO [10]	Flickr30K [25]	Mix665K [11]
Train batch size	32	64	16	64	16	16	16	32
Test batch size	64	64	32	64	16	32	32	1
Train epochs	30	10	18	9	2	10	10	1
Learning rate	3e-6	1e-7	1e-7	1e-5	5e-6	5e-6	5e-6	2e-5
Step size $\mathcal{S}_P$	0.02	0.02	0.02	0.02	-	0.01	0.01	0.0005
Step size $\mathcal{S}_T$	0.4	0.4	0.4	0.2	-	0.2	0.2	0.02/0.01
Interval $\mathcal{I}$	300	300	300	300	-	300	150	18
Metric val_acc	Dev. Acc.	$(R_{I\to T}@1 + R_{T\to I}@1)/2$		(SPICE+CIDEr)/2	-	$(R_{I\to T}@1 + R_{T\to I}@1)/2$		$-$Val_Loss
Weight Decay	0.05					0.2		0

Table B. Configurations of the models for multiple tasks. * indicates two images share one vision transformer in a single forward propagation during inference.

Task	Input resolution	Vision Transformer					Language Transformer				
		Number	Layers	Width	Heads	Intermediate	Number	Layers	Width	Heads	Intermediate
BLIP-Reasoning	384×384	2*	12	768	12	3072	1	12	768	12	3072
BLIP-Retrieval	384×384	1	12	768	12	3072	1	12	768	12	3072
BLIP-Captioning	384×384	1	12	768	12	3072	1	12	768	12	3072
BLIP-VQA	480×480	1	12	768	12	3072	2	12	768	12	3072
CLIP-Retrieval	336×336	1	24	1024	16	4096	1	12	768	12	3072
LLaVA-Image Understanding	336×336	1	24	1024	16	4096	1	32	4096	32	11008

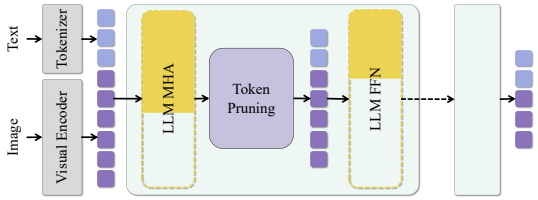

Figure A. Illustration of extending CoMP to the LLaVA-style architecture. We perform pruning in the LLM component, including pruning of vision tokens, language tokens and parameters, since this part accounts for more than 95% of the overall FLOPs.

and varied video content. It contains 10K video clips, each paired with 20 human-annotated textual descriptions. We follow the evaluation protocol of the BLIP model [8] to perform zero-shot video-text retrieval and report the top-1/-5 recall (R@1/5) metrics on the 1K test split.

MME. The MME [5] dataset assesses a model's perception and cognition capabilities. It consists of 14 sub-tasks with 2,000 questions for perception and 800 for cognition, where each image is associated with two binary (Yes/No) queries. By following existing works [20, 22, 24], we report the summation of scores on both capabilities. For the average score in Tab. 3, the summation is normalized by dividing by 2,800, which corresponds to the full score.

MMBench. The MMBench [12] dataset provides a balanced and comprehensive evaluation of model capabilities across three hierarchical levels. It consists of 3,217 multiple-choice questions, with the dataset split into development and test subsets at a 4:6 ratio. By following [11], we report the accuracy on the development set by submitting predictions to the official evaluation server.

GQA. The GQA [7] dataset targets visual understanding and reasoning capabilities in complex real-world scenarios. It contains binary and open queries across real-world reasoning, scene understanding, and compositional question answering. In line with the compared methods, we perform evaluation on the 'testdev' subset from [11], which contains 12,578 questions, and report the standard accuracy.

TextVQA. The TextVQA [17] dataset assesses a model's ability to understand and reason over text present within images. The benchmark requires the model to read textual content from images and answer associated open-ended questions. We report accuracy on the validation split, which comprises 5,000 image-question pairs.

POPE. The POPE [9] dataset evaluates the object hallucination of models. It is built upon 500 images selected from the COCO [10] dataset. For each image, a polling-based query with 6 questions is performed, organized into three subsets

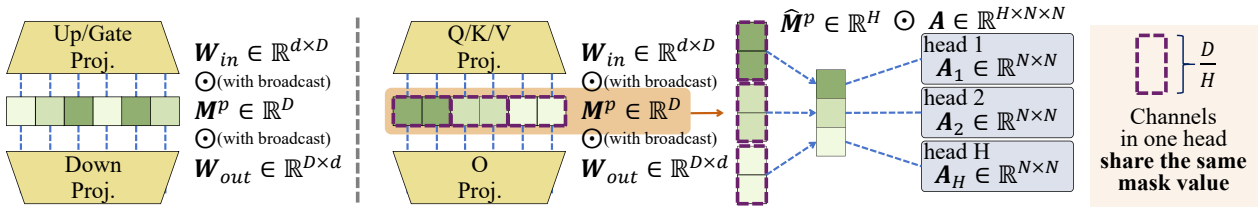

Figure B. Illustration of the dimension mapping for parameter pruning in FFN (Left) and MHA (Right).

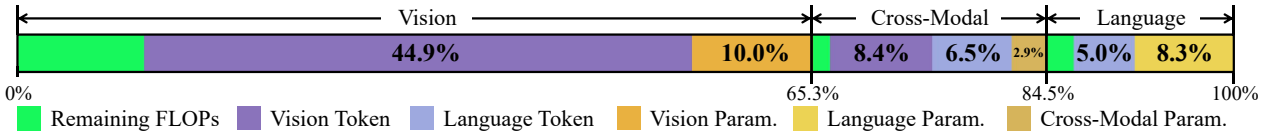

Figure C. FLOPs contributions of different pruning modes for the BLIP model, evaluated on the NLVR2 dataset at a pruning ratio of 0.85.

Table C. Comparison of GFLOPs, Latency (ms per image) and Speedup Ratio across various pruning methods on BLIP using NLVR2 dataset at a pruning ratio of 0.85. Best results are highlighted in **bold**.

Method	GFLOPs	Latency ↓	Speedup Ratio ↑
Uncompressed	132.54	4.85	1.00×
UPop [16]	20.01	2.61	1.86×
MADTP [2]	20.57	1.27	3.82×
CoMP (Ours)	20.26	**1.25**	**3.85**×

Table D. Ablation results of the random exploration ratio ρ on BLIP using NLVR2 dataset at a pruning ratio of 0.8. The default setting is underlined.

ρ	Dev. Acc.	Test Acc.	GFLOPs
0.1	78.70	79.08	26.00
0.2	79.23	79.62	25.97
0.3	79.50	79.76	25.95
0.4	78.79	79.55	26.68

based on different sampling strategies: random, common, and adversarial. By following [11], we report the average F1 score across all three subsets.

Mix665K. The Mix665K [11] data is a mixed collection derived from multiple benchmarks to comprehensively enhance the model's capabilities. It comprises 665K image–conversation pairs and serves as the official training set for LLaVA's supervised fine-tuning process.

C. Detailed Implementation Details

To evaluate the effectiveness of our method, we compress the BLIP [8] and CLIP [15] models fine-tuned for multiple downstream tasks, by following [16] and [2]. We further extend our methods to the LLaVA-v1.5-7B model. The training and testing configurations are summarized in Tab. A. We also report the GFLOPs/TFLOPs of the model during inference, of which the corresponding configurations are summarized in Tab. B.

Our collaborative pruning method builds upon the existing parameter and token pruning frameworks [2, 16, 22], and the multi-mode progressive pruning is achieved by adjusting the corresponding hyperparameters (*i.e.* to generate the $\Delta\Theta_m$ in Algorithm 1). Specifically, we utilize the parameter pruning framework from UPop [16] and, at each stage, increase its parameter reduction ratio by a step size of $\mathcal{S}_P$ for parameter pruning. Meanwhile for token pruning, we adopt the MADTP [2] framework for BLIP/CLIP, and progressively adjust the temperature it defines via a step size of $\mathcal{S}_T$. Notably, since MADTP is not suitable for LLaVA, we instead use the PDrop [22] framework, adjusting its token ratio hyperparameter for pruning. Given that language tokens are far fewer than vision tokens, we set $\mathcal{S}_T$ to 0.02/0.01 for vision/language tokens, respectively, as shown in Tab. A. Pruning is conducted simultaneously with model optimization. Throughout all 'Train epochs', we apply our proposed CoMP method to progressively prune the model until reaching the target FLOPs, after which we fix the pruning configuration and fine-tune the pruned model to recover its performance.

In addition, we set $\rho = 0.2$, $\tau = 5$ in Eq. (9) for random exploration, and $\lambda_0 = 0.4$, $I_{\max} = 5$ for historical information in the MPS module. These hyperparameters are fixed across all experiments, and ablation studies are provided in Sec. E.3.

For BLIP/CLIP, all experiments are conducted on 2 NVIDIA A800 GPUs. By following the training settings of [8] and [15], we adopt the AdamW [14] optimizer, along with a cosine learning rate scheduler [13] and random data augmentation [3]. We adopt the task-specific evaluation metric on the validation subset to compute val_acc, as defined in Eq. (8). For tasks with multiple metrics, we utilize a simple average of the primary metrics, with details provided in Tab. A. Alternatively, max-normalization may be preferable for metrics with varying scales. For LLaVA, given

Table E. Ablation results of the decay factor λ (parameterized by λ_0 and $I_{\max}$) on BLIP using NLVR2 dataset at a pruning ratio of 0.8. The default setting is underlined.

Setting			Dev.	Test	GFLOPs
λ_0	$I_{\max}$	Formulation of λ	Acc.	Acc.	
0.3	1	$\lambda=0.3$	79.07	79.03	26.51
0.4	1	$\lambda=0.4$	79.18	79.33	25.93
0.5	1	$\lambda=0.5$	78.75	79.02	25.91
0.4	2	$\lambda=\max(0.4-0.20(I_m-1),0)$	78.77	78.79	26.32
0.4	4	$\lambda=\max(0.4-0.10(I_m-1),0)$	79.35	79.45	26.31
0.4	5	$\lambda=\max(0.4-0.08(I_m-1),0)$	79.23	79.62	25.97
0.4	6	$\lambda=\max(0.4-0.07(I_m-1),0)$	79.25	79.61	26.19
0.4	8	$\lambda=\max(0.4-0.05(I_m-1),0)$	79.12	79.45	26.33

its architectural characteristics, we collaboratively prune the vision tokens, language tokens and parameters in the LLM component, as presented in Fig. A. A one-epoch SFT is performed on 8 NVIDIA A800 GPUs, with all settings consistent with the uncompressed model [11]. Due to the abscence of validation set, we uniformly sample 3K examples from the full 665K training set for validation and adopt the negative loss (denoted as '$-$Val_Loss') as a proxy for val_acc.

D. Dimension Mapping for Parameter Pruning

Fig. B illustrates the detailed dimension mapping process in parameter pruning. In the FFN module, all operations follow the standard per-channel definition described in Sec. 3. For the MHA module, pruning is performed in a head-wise manner, *i.e.*, all channels within a head are pruned simultaneously to preserve parallelism after compression. Taking the h-th head as an example, this requires that the mask values within the head remain identical, which can be formally expressed as:

$$\boldsymbol{M}^p_{(h-1)d_k+1} = \boldsymbol{M}^p_{(h-1)d_k+2} = ... = \boldsymbol{M}^p_{hd_k}. \tag{1}$$

Meanwhile, the prunable units are reduced from D channels to H heads. For each unit, its importance score is calculated as the average of the channel-level importance scores within the corresponding head:

$$\boldsymbol{S}'^p_h = \frac{1}{d_k} \sum_{i=(h-1)d_k+1}^{hd_k} \boldsymbol{S}'^p_{i,:}. \tag{2}$$

Furthermore, in our CIM module, the mask $\boldsymbol{M}^p$ is transferred from the channel dimension to the attention head dimension, resulting in $\hat{\boldsymbol{M}}^p$ as shown in Eq. (7), where $\hat{\boldsymbol{M}}^p_h = \boldsymbol{M}^p_{hd_k}$.

Table F. Ablation results of the interval steps $\mathcal{I}$ on BLIP using the NLVR2 dataset at a pruning ratio of 0.8. The default setting is underlined.

$\mathcal{I}$	Dev. Acc.	Test Acc.	GFLOPs
240	79.42	79.25	26.50
300	79.23	79.62	25.97
360	79.09	79.71	26.19

Table G. Ablation results of τ in Eq. (9) on BLIP using the NLVR2 dataset at a pruning ratio of 0.8. The default setting is underlined.

τ	Dev. Acc.	Test Acc.	GFLOPs
1	79.30	79.69	26.38
5	79.23	79.62	25.97
10	79.12	79.49	26.68

E. More Experimental Results and Analysis

E.1. Real-World Inference Latency

CoMP employs a structured pruning scheme, which is widely recognized for its ease of model deployment without requiring hardware-specific adaptations [16, 21]. To demonstrate the real-world efficiency of CoMP, we report the inference latency and speedup ratio on a single RTX 4090 GPU in Tab. C. On BLIP-NLVR2 with a FLOPs pruning ratio of 0.85, CoMP achieves a 3.85$\times$ speedup while maintaining state-of-the-art accuracy, as reported in Tab. 1.

E.2. Effect of Distinct Pruning Modes

To demonstrate the overall effect of collaborative pruning, Fig. C presents the FLOPs reduction contributed by the five pruning modes on BLIP-NLVR2 at a pruning ratio of 0.85. For instance, in the full model (*i.e.* 100% FLOPs), the vision branch accounts for 65.3% of the total computation, where vision parameter pruning and vision token pruning contribute FLOPs reductions of 10.0% and 44.9%, respectively.

E.3. Additional Ablations Results

On Random Exploration Ratio ρ. We ablate the random exploration ratio ρ in the MPS module on BLIP using the NLVR2 dataset at a pruning ratio of 0.8. As Tab. D shows, the best performance is achieved when ρ is set between 0.2 and 0.3. Intuitively, a small ρ constrains the model's exploration capability, potentially leading to stuck in suboptimal pruning modes. Conversely, an excessively large ρ may degrade the overall performance with unstable pruning process. The default value of 0.2 represents a balanced trade-off and consistently yields competitive results.

On Decay Factor λ. The design of λ in Eq. (10) aims to leverage historical information while appropriately discounting the outdated contributions. That means, if a pruning mode has not been performed for a long time, its past cost

Table H. Comparison of Dev./Test Acc. (%), R@1/5 (%) and GFLOPs by CoMP and MADTP on BLIP for NLVR2 visual reasoning and COCO image-text retrieval tasks. Reported as mean ± std over 5 seeds. The best results are highlighted in **bold**.

Method	NLVR2-Reasoning			COCO-Retrieval				
	Dev. Acc.	Test Acc.	GFLOPs	I→T R@1	I→T R@5	T→I R@1	T→I R@5	GFLOPs
MADTP [2]	77.16±0.57	77.64±0.47	26.77±0.23	74.38±0.32	91.50±0.24	56.28±0.33	80.74±0.13	30.04±0.44
CoMP (Ours)	**79.13**±0.24	**79.67**±0.28	26.33±0.38	**75.96**±0.27	**92.54**±0.13	**57.32**±0.31	**81.44**±0.30	29.24±0.32

Table I. Comparison of R@1/5 (%) and GFLOPs for BLIP on MSR-VTT dataset for the zero-shot video-text retrieval task. The best results are highlighted in **bold**.

Method	Pruning Mode	Pruning Ratio	V→T R@1	V→T R@5	T→V R@1	T→V R@5	GFLOPs
Uncompressed	/	/	35.8	59.7	43.3	65.5	733.4
UPop [16]	P	0.7	19.1	37.3	24.0	42.2	256.2
MADTP [2]	T	0.7	34.8	59.8	38.9	62.4	265.3
CoMP (Ours)	C	0.7	**35.4**	**60.3**	**39.5**	**63.1**	261.1

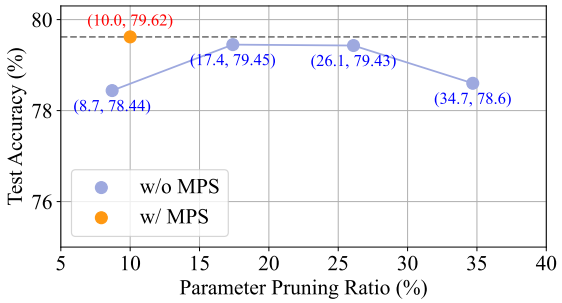

Figure D. Comparison of test accuracy with (w/) and without (w/o) the proposed MPS module on NLVR2 at an overall FLOPs pruning ratio of 0.8.

information should be diminished when guiding subsequent mode selection. To validate this, we conduct a two-step ablation study using BLIP on the NLVR2 dataset, as shown in Tab. E. Specifically, we first fix λ as a constant to introduce static historical weighting, where $\lambda = 0.4$ yields the best performance, and this value is adopted as the initial weight λ_0 in the decaying formulation. Next, we incorporate the linear decay mechanism as in Eq. (10), where historical information beyond $I_{\max}$ stages is no longer considered. Empirically, setting $I_{\max} = 5$ provides a favorable trade-off. These results suggest that incorporating historical information can stabilize the pruning process, but its effect should be limited to a reasonable temporal window, thus avoiding reliance on excessively old data.

On Interval Steps $\mathcal{I}$. The interval $\mathcal{I}$ between mode shifting directly determines the duration of the pruning process. As shown in Tab. F, our method is relatively insensitive to this hyperparameter. In practice, we set $\mathcal{I}$ empirically according to the model size, ensuring that the cost of pruning process does not exceed that of fine-tuning while still enabling sufficient recovery after conducting each pruning mode.

Table J. Comparison of Dev./Test Acc. (%) and GFLOPs by CoMP with different parameter pruning frameworks on BLIP using NLVR2 dataset. The best results are highlighted in **bold**.

Method	Pruning Ratio	Dev. Acc.	Test Acc.	GFLOPs
CoMP w/ UPop [16]	0.8	**79.23**	**79.62**	25.97
CoMP w/ Isomorphic Pruning [4]	0.8	79.09	**79.62**	26.53
CoMP w/ UPop [16]	0.85	75.81	76.08	20.26
CoMP w/ Isomorphic Pruning [4]	0.85	**77.23**	**77.75**	19.83

On Softmax Temperature τ. In Eq. (9), the temperature τ modulates the distribution over the stage intervals of modes since their last execution. As reported in Tab. G, our method exhibits low sensitivity to this . Accordingly, we fix $\tau = 5$ for all experiments in this work.

On Additional Effect of MPS. It's worth noting that the proposed MPS module not only facilitates shifting between pruning modes but also enables adaptively adjusting pruning ratios for parameter and token pruning, under a fixed overall budget of FLOPs. To further evaluate its effectiveness, we compare MPS to the baseline that manually adopts fixed parameter pruning ratios. As illustrated in Fig. D, within the same overall budget of FLOPs, MPS consistently outperforms the counterparts adopting distinct fixed pruning ratios, indicating its superior capability in adaptively adjusting pruning ratios for distinct pruning modes. The results also imply that in the context of multi-mode pruning, the ultimate performance is influenced not only by the global allocation of parameter and token pruning ratios, but also by the execution order of distinct pruning modes, for which our method provides a promising solution.

E.4. Statistical Significance

We evaluate the stability and statistical significance of our method using the BLIP model on the visual reasoning and image-text retrieval tasks. Specifically, we compare our proposed CoMP with the second-best baseline MADTP [2], where each method is evaluated over 5 runs with different random seeds. As shown in Tab. H, CoMP consistently achieves higher average performance (*e.g.* an average gain of 2.03% and 1.04% in NLVR2 Test Acc. and COCO im-

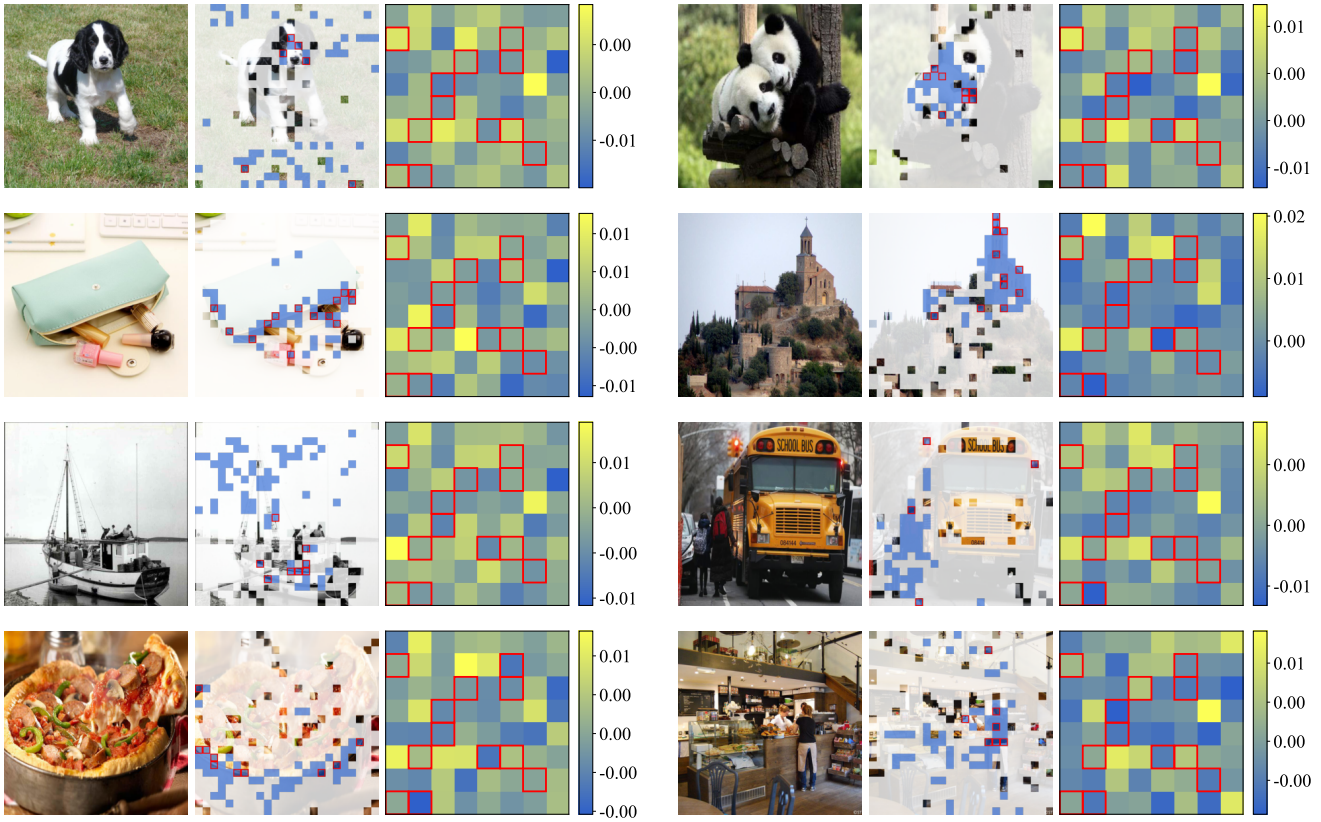

Figure E. Visualization of more examples illustrating inconsistency between parameter importance and token importance in the visual encoder of BLIP. Each example presents, from left to right: (1) the original image; (2) the remaining tokens after token pruning at $layer_{10}$, along with the tokens that contribute most to parameter importance (blue regions), where the overlapping tokens are indicated by red boxes; (3) the heatmap of parameter contributions to token importance at $layer_2$, with the least important parameters (to be pruned by parameter pruning) highlighted by red boxes.

age R@1, respectively) while exhibiting lower standard deviation across most metrics. Furthermore, paired significance tests indicate these gains are statistically significant ($p = 0.0004, 0.0025, 0.0076$ for NLVR2 Test Acc., COCO text R@1 and COCO image R@1, respectively).

E.5. Extended Evaluation on Video-based Tasks

We further validate the effectiveness of our method on video-based vision-language tasks. By following the settings of [8], we employ the BLIP models compressed on the COCO dataset, as reported in Tab. 2, to perform zero-shot video-text retrieval on the MSR-VTT [23] dataset. As shown in Tab. I, our CoMP still consistently outperforms the compared methods, demonstrating the effectiveness and generalization ability of our method on video benchmarks.

E.6. Orthogonality to Single-Mode Pruning

Notably, CoMP adaptively schedules multiple pruning modes to progressively achieve the target pruning ratio, optimizing the pruning process through cross-mode collaboration. This makes it orthogonal to existing importance criteria-based single-mode pruning frameworks. Experiments in Tab. 1 and Tab. 3 have shown its compatibility with token pruning frameworks like MADTP [2] and PDrop [22]. While for parameter pruning, we further incorporate Isomorphic Pruning [4], which is originally developed for vision models, into CoMP by adapting it to the vision-language setting. We evaluate this configuration on BLIP model using NLVR2 dataset, with all hyperparameters unchanged. As shown in Tab. J, at a pruning ratio of 0.8, CoMP combined with either parameter pruning framework achieves similar test accuracy (79.62%). However, at a higher ratio of 0.85, integrating Isomorphic Pruning leads to a notable accuracy gain of 1.67%, owing to its advantage in preserving structural uniformity and mitigating over-pruning. These results demonstrate the generalizability and flexibility of our CoMP framework, and underscores its strong potential to further reduce accuracy loss by combining with more advanced single-mode pruning techniques.

E.7. Discussion on Computational Overhead

The CoMP framework introduces additional computational overhead, primarily from the MPS module. Concretely, the mode shifting at each pruning stage requires to re-evaluate model performance on a validation set to compute the Δval_acc term in Eq. (8), incurring extra time cost. To reduce this cost, we adopt a reduced validation set: for small datasets like NLVR2 [18], we keep the full official validation set, while for the larger datasets like COCO [10], we uniformly sample a subset from the official validation set and fix it. As a result, each re-evaluation tasks around 1 minute, which is minor compared to the overall pruning and fine-tuning time. For instance, at a pruning ratio of 0.8, MADTP [2], UPop [16] and CoMP take 5.8h, 12.3h and 6.7h on two GPUs, respectively, on the BLIP-NLVR2. We consider this computational overhead generally affordable and acceptable, especially in light of the 2.01% and 12.13% improvements in test accuracy that CoMP achieves over these two counterparts.

E.8. More Examples of Importance Inconsistency

To further validate the observations introduced in Sec. 1, we provide additional visualizations illustrating the inconsistency between parameter importance metric and token importance metric. Concretely, we adopt the single-mode importance metrics from UPop [16] and MADTP [2]. For brevity in illustrating parameter behavior, we restrict the visualization to the grouped parameters in MHA, which serve as a representative subset. As shown in the middle column of each example in Fig. E, the tokens identified as important by token pruning rarely coincide with those that contribute most to parameter importance, exhibiting less than 30% overlap (*i.e*. highlighted by red boxes). Meanwhile, the right column of each example further demonstrates that parameters considered unimportant by parameter pruning (*i.e*. highlighted by red boxes) still exert substantial influence on the computation of token importance, as evidenced by high values in the heatmap. These findings indicate that the inconsistency between the two types of importance is widespread, for which our method, especially the CIM module, provides an effective solution for mitigating such cross-mode interference.

References

[1] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *Proceedings of the European Conference on Computer Vision*, pages 382–398, 2016. 1

[2] Jianjian Cao, Peng Ye, Shengze Li, Chong Yu, Yansong Tang, Jiwen Lu, and Tao Chen. Madtp: Multimodal alignment-guided dynamic token pruning for accelerating vision-language transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15710–15719, 2024. 4, 6, 7, 8

[3] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 702–703, 2020. 4

[4] Gongfan Fang, Xinyin Ma, Michael Bi Mi, and Xinchao Wang. Isomorphic pruning for vision models. In *Proceedings of the European Conference on Computer Vision*, pages 232–250, 2024. 6, 7

[5] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. In *Proceedings of the Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. 1, 3

[6] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6904–6913, 2017. 1, 3

[7] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6700–6709, 2019. 1, 3

[8] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *Proceedings of the International Conference on Machine Learning*, pages 12888–12900, 2022. 1, 3, 4, 7

[9] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 292–305, 2023. 1, 3

[10] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision*, pages 740–755, 2014. 1, 3, 8

[11] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306, 2024. 1, 3, 4, 5

[12] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *Proceedings of the European Conference on Computer Vision*, pages 216–233, 2024. 1, 3

[13] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 4

[14] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 4

[15] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry,

Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning*, pages 8748–8763, 2021. 4

[16] Dachuan Shi, Chaofan Tao, Ying Jin, Zhendong Yang, Chun Yuan, and Jiaqi Wang. Upop: Unified and progressive pruning for compressing vision-language transformers. In *Proceedings of the International Conference on Machine Learning*, pages 31292–31311, 2023. 1, 2, 4, 5, 6, 8

[17] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8317–8326, 2019. 1, 3

[18] Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. A corpus for reasoning about natural language grounded in photographs. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 6418–6428, 2019. 1, 3, 8

[19] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4566–4575, 2015. 1

[20] Zichen Wen, Yifeng Gao, Shaobo Wang, Junyuan Zhang, Qintong Zhang, Weijia Li, Conghui He, and Linfeng Zhang. Stop looking for "important tokens" in multimodal language models: Duplication matters more. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 9972–9991, 2025. 3

[21] Zimeng Wu, Jiaxin Chen, and Yunhong Wang. Samp: Subtask aware model pruning with layer-wise channel balancing for person search. In *Chinese Conference on Pattern Recognition and Computer Vision*, pages 199–211, 2023. 5

[22] Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, et al. Conical visual concentration for efficient large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14593–14603, 2025. 3, 4, 7

[23] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5288–5296, 2016. 1, 7

[24] Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. Visionzip: Longer is better but not necessary in vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19792–19802, 2025. 3

[25] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014. 1, 3