MADTP: Multimodal Alignment-Guided Dynamic Token Pruning for Accelerating Vision-Language Transformer

Supplementary Material

A. Dataset and evalution metrics

We have conducted extensive experiments to evaluate our MADTP framework, utilizing four diverse multimodal datasets, namely NLVR2 [12], COCO [6], Flickr30k [13], and VQA v2.0 [3]. These datasets encompass a wide range of tasks and challenges, allowing us to assess the effectiveness of the proposed framework comprehensively. More details are shown below.

A.1. NLVR2

The NLVR2 [12] dataset is curated to advance research in computer vision and natural language processing for visual reasoning tasks. Its main objective is to enable models to determine if two images share common objects or scenes using provided natural language descriptions. With 107,292 examples of human-written English sentences grounded in pairs of photographs, NLVR2 offers linguistic diversity and visually complex images. The dataset is divided into subsets: the training set contains 86,373 examples, the development set consists of 6,982 examples, Test-P comprises 6,967 examples, and Test-U includes 6,970 examples. The primary evaluation metric is Accuracy (Acc), reflecting the proportion of correctly predicted image pairs. These evaluation metrics aid researchers in assessing model performance, facilitating comparisons and guiding improvements.

A.2. COCO

The COCO [6] dataset is a valuable resource for both image-text retrieval and image caption tasks, containing a vast amount of annotated data. It includes 82,783 training images with 413,915 captions, 40,504 validation images with 202,520 captions, and 40,775 testing images with 379,249 captions. For the image-text retrieval task, Recall@k serves as a useful evaluation metric. It quantifies the proportion of relevant results that are correctly retrieved within the top-k ranked items. This metric is valuable for assessing the model's ability to recall relevant captions when given an image query and vice versa. For the image caption task, evaluation metrics such as CIDEr and SPICE are commonly used. CIDEr (Consensus-based Image Description Evaluation) leverages consensus-based scoring by comparing generated captions to multiple reference captions, providing a measure of the quality of the generated captions. SPICE (Semantic Propositional Image Caption Evaluation) considers the semantic structure of the captions by evaluating their ability to describe the image content accurately.

A.3. Flickr30k

The Flickr30k [13] dataset is widely utilized for image caption and image-text retrieval tasks, providing a substantial collection of images with associated captions. It consists of three distinct subsets: a training set comprising 29,000 images and 145,000 captions, a validation set containing 1,000 images and 5,000 captions, and a test set with 1,000 images and 5,000 captions. This dataset provides researchers with a diverse range of images and associated textual descriptions, enabling the development and evaluation of models for various image understanding tasks. In the experiments of this paper, we focus on evaluating the performance of the MADTP compressed models for the image-text retrieval task using the Flickr30k dataset. To ensure consistency with evaluation practices used in the COCO [6] dataset, we employed the same Recall@k metric as the final evaluation metric.

A.4. VQA 2.0

The VQA 2.0 [3] dataset serves as a widely adopted resource for Visual Question Answering (VQA) task, where models are tasked with answering questions related to images. It is an extended version of the original VQA dataset, addressing its limitations and providing a more comprehensive evaluation setup. The dataset is derived from the COCO [6] dataset and is divided into three main subsets: training, validation, and testing. The training set consists of approximately 82,783 images with 443,757 associated questions. The validation set contains around 40,504 images with 214,354 questions, while the testing set comprises about 81,434 images with 447,793 questions. Notably, the testing set is further divided into two distinct subsets: test-dev and test-std. The test-dev subset is designated for model development and fine-tuning purposes, while the test-std subset is reserved for official evaluation and facilitates performance comparisons. Evaluation of models on the VQA 2.0 dataset employs various metrics. The primary metric is Accuracy (Acc), which measures the proportion of correctly answered questions. Additionally, the dataset provides per-question-type and per-answer-type accuracy metrics, allowing for a more detailed analysis of model performance across different question and answer categories.

A.5. GFLOPs

GFLOPs (Giga Floating Point Operations per Second) is a widely adopted metric for quantifying the computational costs of computer systems, particularly in the fields of deep

learning and artificial intelligence. It measures the number of floating-point operations that a system can perform in one second, with "Giga" representing one billion (10^9) operations. In this paper, the GFLOPs can vary for different inputs due to the instance-level dynamic pruning scheme employed by our MADTP. Therefore, in our experiments, we opted to calculate the averaged GFLOPs over the entire dataset to effectively measure the computational overhead of the compressed model.

B. Implementation details

In our experiments, we employ the MADTP framework to compress Vision-Language Transformers, specifically the CLIP [9] and BLIP [5] models. These models are initialized with pretrained weights obtained from the official implementation of [10]. Table 1 and Table 2 present detailed hyperparameter settings for each model during the compression training process. Further, Table 3 details the architecture configures of the Vision-Language Transformers used in different multimodal models. In our experimental setup, we train the models using 8 A100 GPUs, with a fixed batch size of 32. Note that, unlike the two-stage approach employed in Upop [10], our method is a one-stage approach that eliminates the search stage, resulting in a significant reduction in training time. The MADTP framework exhibits fast convergence, often achieving promising results within just 1-2 epochs. For example, in the case of BLIP-VQA, impressive performance is observed after only 3 epochs of training. In terms of specific hyperparameters, the number of learnable tokens is consistently set to 100, and the channel dimension is set to 768 across different models. Additionally, the hyperparameter α in the loss function is consistently set to 0.1. To enable parallel training, we incorporated the "max-keep" operation within each mini-batch to retain crucial tokens. We will release the code, allowing others to build upon our work.

Hyperparameters	CLIP [9]	
	COCO [6]	Flickr30K [13]
Optimizer	AdamW [8]	
AdamW β	(0.9, 0.999)	
Weight decay	0.2	
Batch size	32	
Train epochs	5	10
Train LR	1e-5	
Learnable token numbers K	100	
Learnable token dimensions d_k	768	
Loss weight α	0.1	
Prune operation	max-keep	
Train LR schedule	CosineLRScheduler [7]	
Data augmentation	RandomAugment [2]	

Table 1. Training hyperparameters for compressing CLIP-based models on both COCO and Flickr30K datasets.

C. Supplementary Experiments and Analyses

C.1. Comparison with Token Pruning

In this study, we conduct a comparative analysis between our MADTP and some recent token pruning techniques, including CrossGET [11] and ELIP [4]. However, it should be noted that these methods have not been formally published and are currently only available on the arXiv website. Hence, we do not include them in our main paper.

Detailed comparisons are shown in Table 4 and Table 5. Specifically, CrossGET [11] introduces the use of cross tokens as guidance for both modalities and employs the single-modality token merge method [1] for accelerating VLTs. On the other hand, ELIP [4] proposes a vision token pruning and merging method that removes less influential tokens based on the supervision of language outputs. Both of these methods overlook the significance of modality alignment guidance in the multimodal token pruning process. Additionally, they belong to the category of static token pruning, which cannot achieve adaptive dynamic compression for Vision-Language Transformers. In contrast, our MADTP method introduces the Multi-modality Alignment Guidance (MAG) module, which enables modality alignment guidance during VLT compression. Further, we design the Dynamic Token Pruning (DTP) module, which can achieve both input instance- and layer-wise compression of VLTs. Due to the differences in experimental settings and challenges related to code release, we focus on comparing the final compression results with these two methods. The experimental results clearly show that our MADTP achieves superior compression performance compared to CrossGET [11] and ELIP [4], which provide strong evidence for the effectiveness of our approach.

C.2. Orthogonality with Parameter Pruning

In this section, we conduct experiments to validate the orthogonality of our MADTP framework with parameter pruning techniques. The detailed results are presented in Table 6. Here are the specifics of the experimental setup: we firstly apply a parameter pruning approach [10] to the BLIP model, using a compression ratio of 0.15 on the NLVR2 dataset as the initial compression step. Subsequently, we further accelerate the compressed model using our MADTP with a reduce ratio of 0.3. The objective of this additional pruning step is to dynamically eliminate non-critical tokens, thereby further enhancing model efficiency. The thorough experimental results confirm the orthogonality of our MADTP framework with parameter pruning approaches. In detail, after applying our MADTP method, the model exhibits a 0.26% increase in accuracy on the dev set and a 0.17% increase on the test set. The GFLOPs of the compressed model decrease by 20.8%, indicating a substantial reduction in computational costs. Remarkably, de-

Hyperparameters	BLIP-NLVR [5]	BLIP-Caption [5]	BLIP-VQA [5]	BLIP-Retrieval [5]	
	NLVR2 [12]	COCO [6]	VQAv2 [3]	COCO [6]	Flickr30K [13]
Optimizer	AdamW [8]				
AdamW β	(0.9, 0.999)				
Weight decay	0.05				
Batch size	32				
Train epochs	15	5	3	5	10
Train LR	3e-6	1e-5	2e-5	1e-6	1e-7
Learnable token numbers K	100				
Learnable token dimensions d_k	768				
Loss weight α	0.1				
Prune operation	max-keep				
Train LR schedule	CosineLRScheduler [7]				
Data augmentation	RandomAugment [2]				

Table 2. Training hyperparameters for compressing BLIP-based models on five kinds of datasets.

Model	Input resolution	Vision Transformer				Language Transformer			
		number	layers	width	heads	number	layers	width	heads
BLIP-NLVR [5]	384×384	2*	12	768	12	1	12	768	12
BLIP-Caption [5]	384×384	1	12	768	12	1	12	768	12
BLIP-VQA [5]	480×480	1	12	768	12	2	12	768	12
BLIP-Retrieval [5]	384×384	2	12	768	12	2	12	768	12
CLIP [9]	336×336	2	24	1024	16	2	12	768	12

Table 3. Architecture configures of all models used in our experiments. The superscript * indicates 2 Transformers share parameters.

Approach	Image→Text			Text→Image			GFLOPs
	R@1	R@5	R@10	R@1	R@5	R@10	
ToMe‡ [11]	90.8	99.2	99.5	78.1	95.3	97.7	-
CrossGET [11]	92.1	**99.7**	99.8	79.6	**97.5**	98.0	-
UPop [10]	93.2	99.4	99.8	80.5	95.4	97.6	201.1
MADTP (Ours)	**93.9**	99.5	**99.8**	**83.3**	97.0	**98.5**	178.8

Table 4. Performance comparisons of different methods when compressing CLIP on the Flickr30K dataset of the Image-Text Retrieval task. The R@1, R@5, and R@10 are the higher the better. The best results are in bold. The symbol ‡ represents the model implementation is derived from CrossGET [11].

Approach	Flickr30K							COCO						
	Image→Text			Text→Image			GFLOPs	Image→Text			Text→Image			GFLOPs
	R@1	R@5	R@10	R@1	R@5	R@10		R@1	R@5	R@10	R@1	R@5	R@10	
EViT† [4]	87.3	98.5	99.4	75.1	93.5	96.4	48.0	66.8	88.9	93.9	50.8	77.9	86.3	48.0
ToMe† [4]	91.5	98.8	99.4	80.5	95.6	97.9	69.8	71.5	91.6	95.9	55.3	81.2	88.7	69.8
ELIP [4]	92.2	99.1	99.7	80.3	96.0	98.0	93.4	72.0	91.9	95.9	56.3	81.2	88.7	93.4
UPop [10]	94.0	99.5	99.7	82.0	95.8	97.6	91.0	77.4	93.4	97.0	59.8	83.1	89.8	88.3
MADTP (Ours)	**95.1**	**99.5**	**99.7**	**82.3**	**96.2**	**98.0**	74.5	**79.1**	**94.2**	**97.2**	**60.3**	**83.6**	**89.9**	87.4

Table 5. Performance comparisons of different methods when compressing BLIP on the Flickr30K and COCO datasets of the Image-Text Retrieval task. The R@1, R@5, and R@10 are the higher the better. The best results are in bold. The symbol † represents the model implementation is derived from ELIP [4].

spite these improvements, the model's parameters only increases by a mere 0.4%. Therefore, combining both pruning schemes in a joint compression strategy yields outstanding compression results. Our future work involves integrating a parameter pruning scheme into the proposed MADTP framework for comprehensive VLT compression.

Approach	Reduce ratio (Params)	Reduce ratio (GFLOPs)	Dev Acc	Test Acc	Params	GFLOPs
Uncompressed	-	-	82.48	83.08	259.45	132.54
Parameter pruning [10]	0.15	-	81.54	82.35	**219**	117.32
Parameter pruning [10] + **MADTP**	0.15	0.3	**81.80**	**82.52**	220↑0.4%	**92.75** ↓20.8%

Table 6. The orthogonality of our MADTP framework with parameter pruning techniques. Compress BLIP on the NLVR2 dataset for visual reasoning task. Reduce ratio (Params) represents the proportion of model parameter compression, and Reduce ratio (GFLOPs) denotes the compression ratio of model computational costs. The experimental results demonstrate that combining our approach with parameter pruning techniques yields superior compression performance.

Approach	Modality	Dev Acc	Test Acc	GFLOPs
Uncompressed	-	82.48	83.08	132.54
STP	vision only	80.04	80.50	67.69
	language only	74.67	75.01	129.54
	vision and language	78.08	77.61	68.31
MADTP	vision only	**82.27**	82.45	66.41
	language only	77.33	77.58	128.98
	vision and language	81.97	**82.85**	**66.16**

Table 7. Ablation studies of MADTP on different modalities.

C.3. Compression on different modalities

We also perform ablation studies on applying the proposed MADTP method to compress different modalities for VLTs, and the detailed results can be found in Table 7. Due to the varying importance of different modalities in accomplishing the final task and the different computational costs associated with each modality branch, individually compressing different modalities has a significant impact on the overall performance of the compressed model. In our experiment, we separately compressed various modal branches of the BLIP model on the NLVR2 dataset, including the only vision branch, only language branch, and the combined vision and language branch. The experimental results indicate that the visual branch has higher token redundancy, allowing for significant reductions in computational costs through token pruning. Conversely, the text branch has lower computational cost and is essential for multimodal tasks. Thus, compressing the text branch has a more substantial impact on model performance, albeit with minimal decrease in GFLOPs. These observations aligns with the finding of the CrossGET [11] method. However, our MADTP method additionally accounts for modality alignment and integrates an adaptive token pruning mechanism, facilitating collaborative compression of both modalities and achieving superior compression results.

C.4. Effect of Hyperparameters

In this section, we conduct additional ablation studies to validate the hyperparameters that affect the performance of MADTP. Firstly, we extend our analysis about the Token Importance Scores(TIS), as shown in Table 8. Furthermore,

Components of MADTP		Dev Acc	Test Acc	GFLOPs
TIS	only w/S_{self}	81.49	82.13	70.46
	only w/S_{token}	80.68	81.00	66.74
	only w/S_{cls}	81.62	82.25	69.67
	S_{self} & S_{token}	81.79	82.32	67.08
	S_{self} & S_{cls}	81.40	82.35	70.67
	S_{token} & S_{cls}	81.76	82.41	66.19
	S_{self} & S_{token} & S_{cls}	**81.97**	**82.85**	**66.16**

Table 8. Results of compressing the BLIP model on the NLVR2 dataset with different token importance scores.

Setting	Batch size	Temperature	Test Acc	GFLOPs
Baseline	16	1.26	82.35	67.62
Inference	1	1.26	77.90	38.46
		0.44	81.86	67.04
	4	1.26	81.04	52.13
		0.89	82.20	66.97
	32	1.26	**82.36**	75.08
		1.43	82.08	68.37

Table 9. The performance of the 0.5 compressed BLIP model on NLVR2 dataset when using different batch sizes during inference. Our baseline model is trained with a batch size of 16, and the GFLOPs with different batch sizes can be adjusted by controlling the temperature to maintain consistency with the baseline.

Batch size	Sorted	Dev Acc	Test Acc	GFLOPs
1	N	76.96	77.90	38.46
	Y	-	77.74↓	38.50
4	N	80.48	81.04	52.13
	Y	-	81.16↑	53.36
16	N	81.64	82.35	67.62
	Y	-	82.59↑	67.61
32	N	81.96	82.36	75.08
	Y	-	82.74↑	73.78

Table 10. Performance of the 0.5 compressed BLIP model on the NLVR2 dataset when using different instance order. Sorted Y means we first sort the instances according to their difficulty and then use the compressed model for inference.

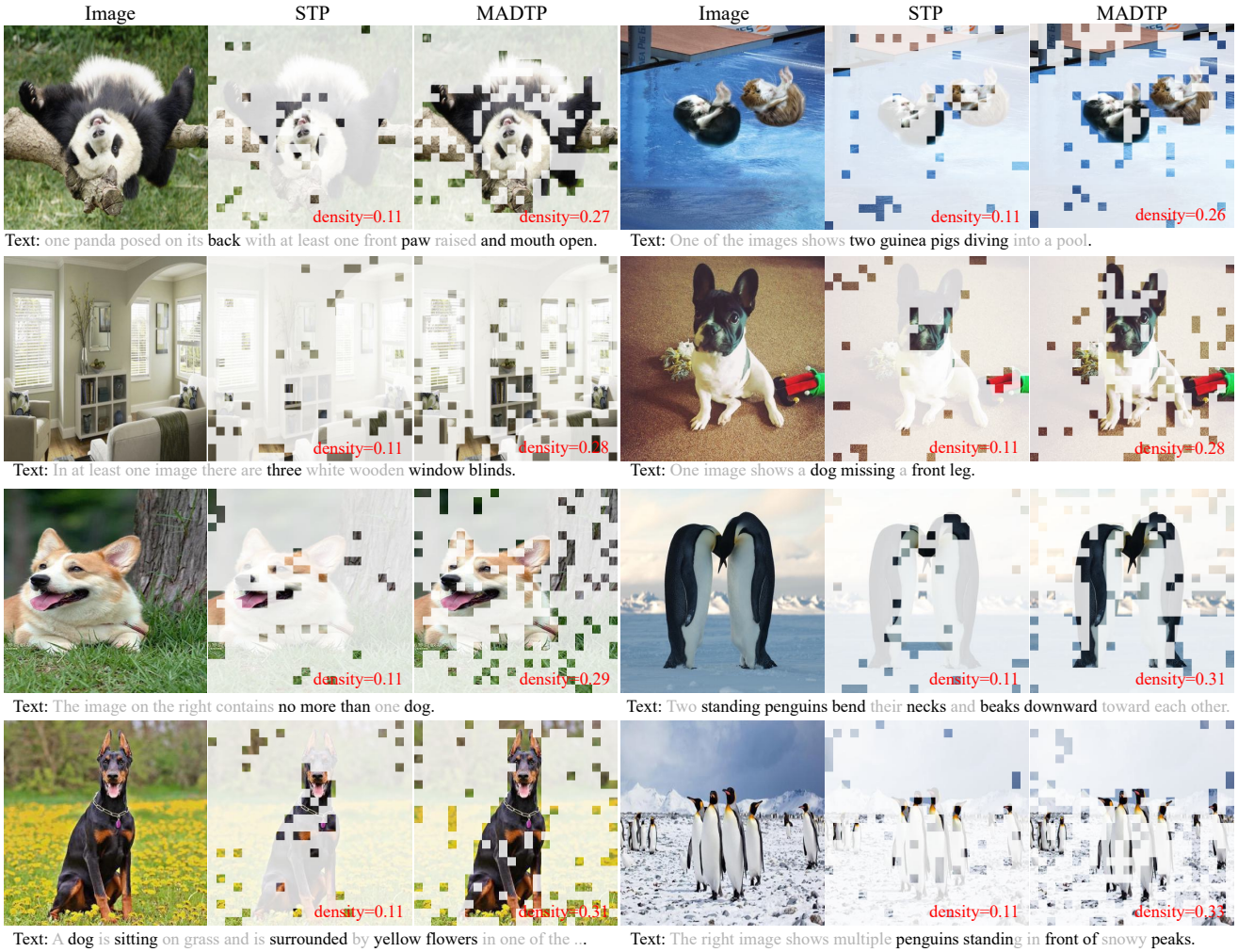

Figure 1. Visualization comparisons of token pruning results between STP and MADTP, providing strong evidence that our approach emphasizes modality correlation, effectively avoids pruning crucial tokens and dynamically adjusts pruning ratio according to inputs.

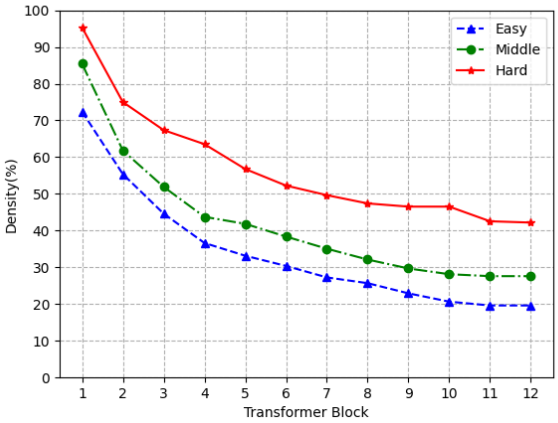

Figure 2. Comparisons of MADTP token pruning in each transformer block for samples of different instance complexity levels, including Easy, Middle, and Hard samples. The density represents the ratio of retained tokens to the total number of original tokens.

we discover that our MADTP is significantly influenced by the batch size during the inference stage, as demonstrated in Table 9. The reason behind this observation is that we adopt the max-keep pruning strategy in the token pruning process, which selects the maximum number of tokens to be retained across input instances in a mini-batch. Therefore, when using a smaller batch size for model inference, the GFLOPs significantly decrease, leading to a decline in performance. However, by adjusting the temperature parameter T, we can increase the GFLOPs with the smaller batch size to match the baseline model, thereby restoring the performance. This experiment proves the strong correlation between the compressed model's performance and GFLOPs. In addition, as shown in Table 10, we observe that sorting the input instances based on their difficulty during inference leads to improved performance. This finding suggests that applying the max-keep strategy to sorted input instances can further enhance compressed models' performance.

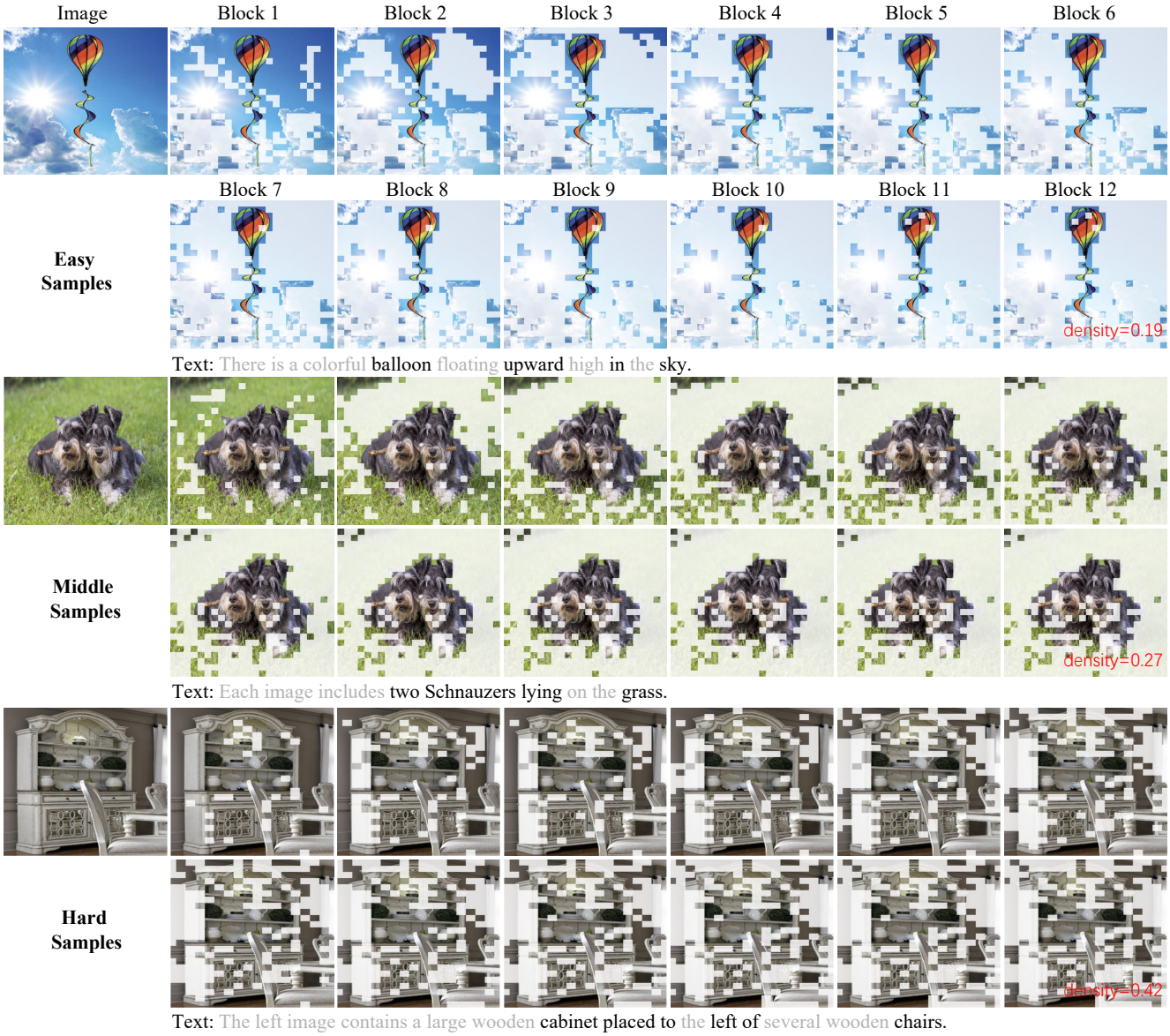

Figure 3. Visualization of the compressed results of MADTP on samples with different levels of instances complexity, including Easy, Middle, and Hard samples. The density represents the ratio of retained tokens to the total number of original tokens.

C.5. Visualization of MADTP

In this section, we visualize the token pruning results of the proposed MADTP framework using a compressed BLIP model with a reduction ratio of 0.5 on the NLVR2 dataset. In Fig. 1, we present an extended visualization comparison between Static Token Pruning (STP) and our MADTP approach. It is evident that our MADTP emphasizes the correlation between modalities and successfully avoids pruning critical tokens. Additionally, we further visualize MADTP token pruning in each transformer block for samples with different instance complexity levels, including Easy, Middle, and Hard samples. Fig. 2 illustrates the token density in the visual branch of VLTs at each transformer block, while Fig. 3 showcases the specific positions of token pruning in each block. These visualizations demonstrate the adaptive dynamic compression capability of the proposed MADTP framework for different input instances. Finally, we show additional visualizations of token compression using the MADTP framework for easy and hard samples in Fig. 4 and Fig. 5. These visualizations further validate the effectiveness of MADTP in dynamically compressing tokens for Vision-Language Transformers.

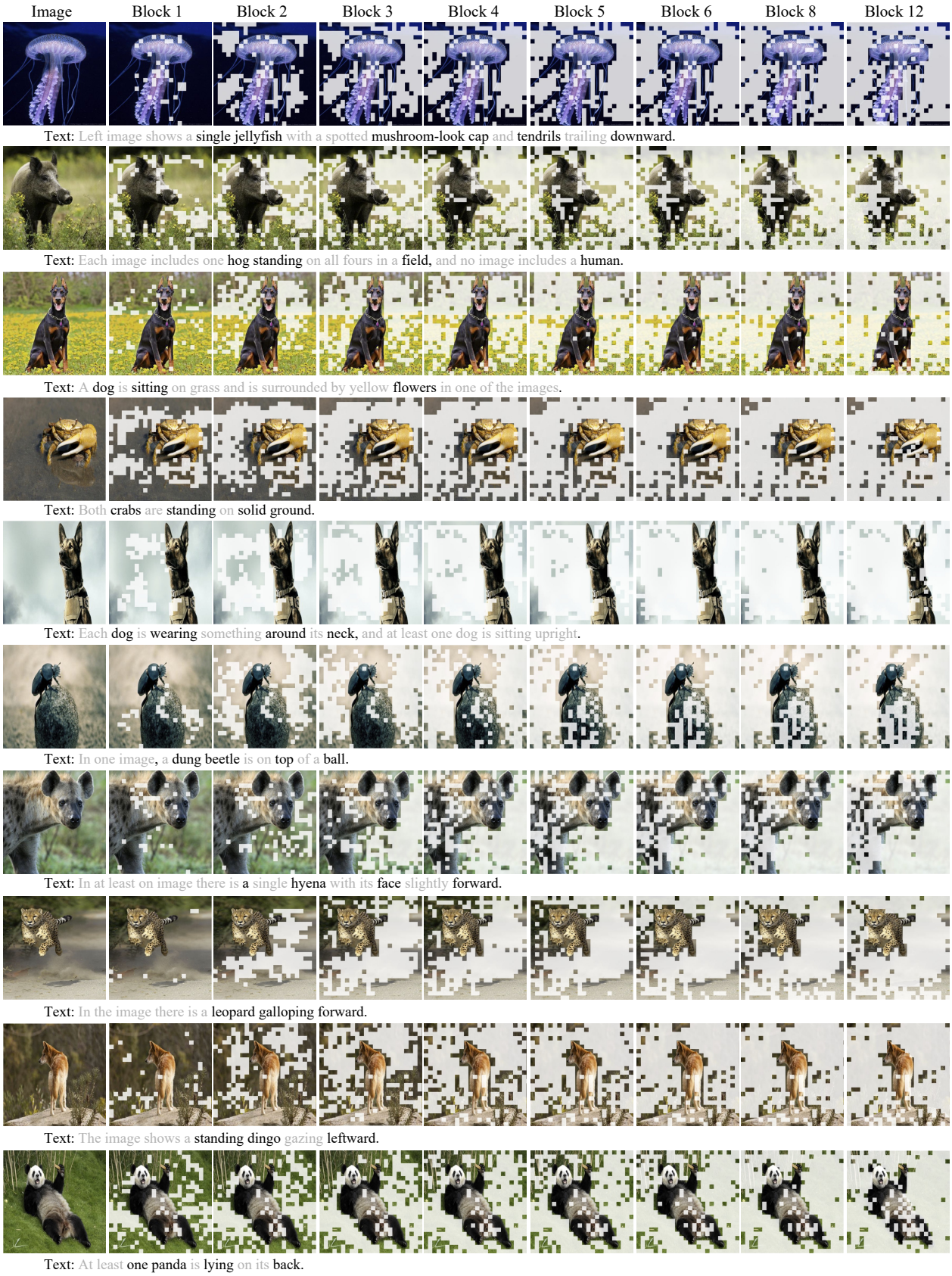

Figure 4. Visualization of our MADTP's compressed BLIP results on **Easy Samples** from the NLVR2 dataset at each transformer block.

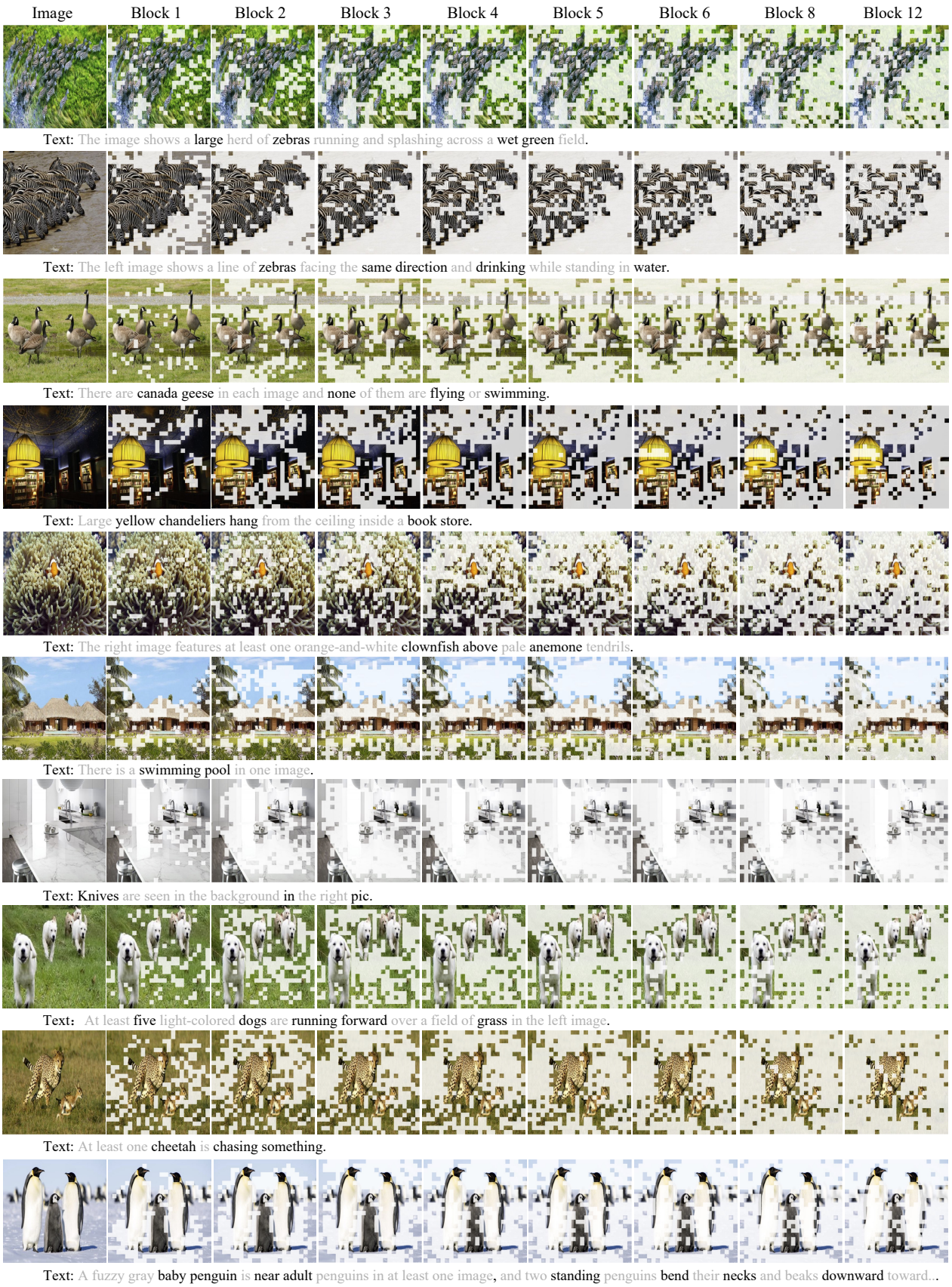

Figure 5. Visualization of our MADTP's compressed BLIP results on **Hard Samples** from the NLVR2 dataset at each transformer block.

References

[1] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. *arXiv preprint arXiv:2210.09461*, 2022. 2

[2] 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 workshops*, pages 702–703, 2020. 2, 3

[3] 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 conference on computer vision and pattern recognition*, pages 6904–6913, 2017. 1, 3

[4] Yangyang Guo, Haoyu Zhang, Liqiang Nie, Yongkang Wong, and Mohan Kankanhalli. Elip: Efficient language-image pre-training with fewer vision tokens. *arXiv preprint arXiv:2309.16738*, 2023. 2, 3

[5] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022. 2, 3

[6] 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 *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 1, 2, 3

[7] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 2, 3

[8] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 2, 3

[9] Alec Radford, JongWook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *Cornell University - arXiv,Cornell University - arXiv*, 2021. 2, 3

[10] 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 40th International Conference on Machine Learning*, pages 31292–31311. PMLR, 2023. 2, 3, 4

[11] Dachuan Shi, Chaofan Tao, Anyi Rao, Zhendong Yang, Chun Yuan, and Jiaqi Wang. Crossget: Cross-guided ensemble of tokens for accelerating vision-language transformers. *arXiv preprint arXiv:2305.17455*, 2023. 2, 3, 4

[12] Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. A corpus for reasoning about natural language grounded in photographs. *arXiv preprint arXiv:1811.00491*, 2018. 1, 3

[13] 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, 2, 3