Supplementary Material
PELA: Learning Parameter-Efficient Models with Low-Rank Approximation

Abstract

In this supplementary material, we mainly provide four categories of information.

- *A simple derivation of the compression ratio.*
- *More literature review about the vision transformers.*
- *Detailed experimental settings, especially the implementation details.*
- *More experimental results of our proposed method.*

1. Derivation of the Compression Ratio

Let us recall the low-rank approximation on one matrix multiplication operation,

$$\begin{aligned}\mathbf{W}^T\mathbf{x} &\approx (\mathbf{U}\mathbf{V}^T)^T\mathbf{x} \\ &= \mathbf{V}(\mathbf{U}^T\mathbf{x}),\end{aligned} \tag{1}$$

where the pre-trained weight matrix $\mathbf{W} \in \mathbb{R}^{d_{in}\times d_{out}}$ and $x \in \mathbb{R}^{d_{in}}$, $\mathbf{U} \in \mathbb{R}^{d_{in}\times d_{lr}}$ and $\mathbf{V} \in \mathbb{R}^{d_{out}\times d_{lr}}$ are low-rank matrices. In this context, the original operation takes $\mathcal{O}(d_{in} \times d_{out})$ to run, as compared to that of the right-hand side of the equal sign $\mathcal{O}((d_{in} + d_{out}) \times d_{lr})$. If we intend to use a compression ratio of κ, *i.e.*, compressing the original model $\kappa\times$, we have $d_{in} \times d_{out} \cong \kappa(d_{in} + d_{out}) \times d_{lr}$. Thereafter, we can easily obtain $d_{lr} = \frac{1}{\kappa}\frac{d_{in}\times d_{out}}{d_{in}+d_{out}}$.

We choose to use a universal κ for each matrix multiplication for simplicity. As a result, the total number of parameters can be easily approximated as $\frac{1}{\kappa}$. However, the matrix rank can be affected by several factors, especially the model depth. It is thus favorable to design an adaptive strategy for different matrix multiplication operations. We leave this exploration as future work.

2. Related Work on Vision Transformers

The past few years have witnessed the pervasive prosperity of Transformers in the language realm [6, 31]. This wave was first initiated to vision by Vision Transformer (ViT) [7], wherein each image is evenly split into patches, conforming to the inputs of a text-based Transformer model. Due to its superior performance, the following studies expanded this success from image classification [7, 29] to more challenging object detection [4, 8, 44], segmentation [35] and 3-D vision domains [22, 42].

One advantage of Transformers over the traditional CNNs is their weak inductive bias and capture of long-range dependencies from the self-attention operation [12, 23]. Though this merit is widely acknowledged by the existing literature, researchers have also endeavored to explore the viability of coupling Transformers and CNNs. A typical implementation is to resort to the convolutional embedding of patch tokens, followed by the self-attention action on the extracted features [32, 33]. In this way, the locality and global semantics are simultaneously modeled to yield improved results. Other common strategies in CNNs, such as hierarchical architecture [19] and pooling [41] are also extensively studied and demonstrate certain improvements in model performance, efficiency, and image throughput. In addition to the pure vision scope, some recent work has introduced vision Transformers to the vision-language tasks [25, 28]. For instance, [15, 17, 28] first pre-train a general model on large-scale image-text pair datasets [26, 27], and then transform it to downstream cross-modal retrieval [18], visual question answering [1], and visual entailment [36].

3. Experimental Setting

For all the experiments, we pre-trained and fine-tuned our model on four NVIDIA RTX A5000 GPUs. Due to resource constraints, we employed a smaller batch size for each respective baseline. We measured the number of model parameters and FLOPs with the open DeepSpeed toolkit[1]. In addition, we also leveraged public code frameworks, *i.e.*, HuggingFace[2] and MMCV[3], for simple computation of FLOPs. Pertaining to this experiment, the batch size for vision-only and vision-language models is set to 1 and 32, respectively.

3.1. Common Efficient Learning Baselines

We evaluated our PELA against four efficient baselines: **TinyBERT [14]** and **MaskAlign [38]** from the feature-

[1] https://www.deepspeed.ai/.
[2] https://huggingface.co/.
[3] https://mmcv.readthedocs.io/en/latest/.

based knowledge distillation group; **ToMe [2]** - a recent strong vision token pruning approach; and **LoRA [13]**, which is a widely used parameter-efficient transfer learning baseline. For TinyBERT, MaskAlign, and ToMe, we carefully tuned their hyperparameters so that the distilled model has a similar number of parameters or FLOPs to ours for a fair comparison.

TinyBERT [14] investigates various knowledge distillation techniques applied to the original BERT model [6]. These techniques involve aligning different components such as logits, embedding matrices, hidden states, and attention matrices. However, in our experiments, we focus on the alignment of hidden states and attention matrices, as the ViT models are unable to incorporate the other two techniques effectively. In particular, for the student model, take the compression ratio of 2 as an example, we kept six layers out of the original twelve layers of the ViT model.

MaskAlign [38] introduces a highly effective feature-based knowledge distillation approach. The key to MaskAlign is the Dynamic Alignment (DA) module, which specifically addresses the issue of input inconsistency between the student and teacher models. In our implementation, we follow a similar approach as TinyBERT to construct the student model.

ToMe [2] first identifies similar tokens and then merges them to reduce the number of vision tokens. We used their official public code to implement the proportional attention mechanism in the self-attention module of ViT. In our experiments, we carefully tuned the number of tokens reduced per layer r to make sure that the FLOPs are similar to our method. Additionally, following the practice of ToMe, we set "prop_attn" to be true to ensure that merged tokens can receive proportional attention.

Note that due to the hierarchical design of the Swin-Transformer architecture [19], it is not feasible to apply the token merging technique from ToMe [2], which randomly merges tokens. Additionally, the token reduction process is incompatible with the UperNet architecture, as it necessitates a fixed number of tokens in different layers. Consequently, we were unable to provide detailed comparisons with these methods.

LoRA [13] is a representative method in the realm of parameter-efficient transfer learning. It introduces two low-rank matrices to the original fixed large model and focuses on fine-tuning only these two matrices. By doing so, Lora achieves remarkable results across a diverse range of downstream fine-tuning tasks, and in some cases, even surpasses the performance of the conventional full fine-tuning strategy. To optimize efficiency, we maintain the rank of Lora at 32.

3.2. Vision-Only Downstream Tasks

Semantic Segmentation. The UperNet framework [34] is adopted upon the backbone for semantic segmentation following [19]. We fine-tuned our low-rank model on the ADE20k [43] dataset and reported the mIoU metric on the validation set.
Object Detection. We also tested the object detection performance of our method on the MSCOCO [18] dataset. In particular, we employed the Cascade Mask R-CNN [3, 11] framework with Swin-Base as the backbone due to the availability of source codes for a fair comparison.

3.3. Vision-Only Baseline Models

Pre-training. To pre-train our models, we maintained most of the settings from DeiT-Base [29], Swin-Base [19], and DeiT-III-Large [30], with the exception of the following hyper-parameters. Due to resource limitations, we reduced the number of pre-training epochs from 300 to 50, which already yields satisfactory performance. For DeiT-III-Large, we decreased the batch size to 32 images per GPU and adjusted the learning rate accordingly. Pertaining to the loss weights hyper-parameters α and β, we kept them as 1.0 and 10.0 for DeiT models, respectively; while for Swin-Base, we set both to 1.0.
Downstream Finetuning. To perform semantic segmentation, we utilized the MMSegmentation framework and employed the UperNet [34] architecture for accurate segmentation. The input resolution is set to 512×512. We used a batch size of 4 on 4 GPUs, resulting in an effective batch size of 16 for DeiT-Base and Swin-Base. While for DeiT-III-Large, the batch size is limited to 2 on 4 GPUs. We trained the model using the AdamW [20] optimizer for 160,000 steps.

Regarding object detection, we utilized the MMDetection framework and adopted the Cascade Mask RCNN [3, 11] as our detection head, which provides superior accuracy for detecting objects in complex scenes. The input resolution of each image is set to 1,024×1,024. We fine-tuned the model using a 1× schedule, consisting of 12 epochs in total. The learning rate was decayed by factors of 10 and 100 at the 8-th and 11-th epoch, respectively, to help the model converge more effectively.

3.4. Vision-Language Downstream Tasks

Image-Text Retrieval is composed of two sub-tasks: image-to-text retrieval (**TR**) and text-to-image retrieval (**IR**). We justified the model performance on Flickr30K [24] and MSCOCO dataset [18] using the recall metric **R**@n, *i.e.*, truncated top-n results is employed.
Visual Entailment (SNLI-VE) [37] predicts the relationship of an image-text pair with three classes: entailment, neutral, or contradictory. We followed previous literature [5, 17] to treat this task as three-way classification.

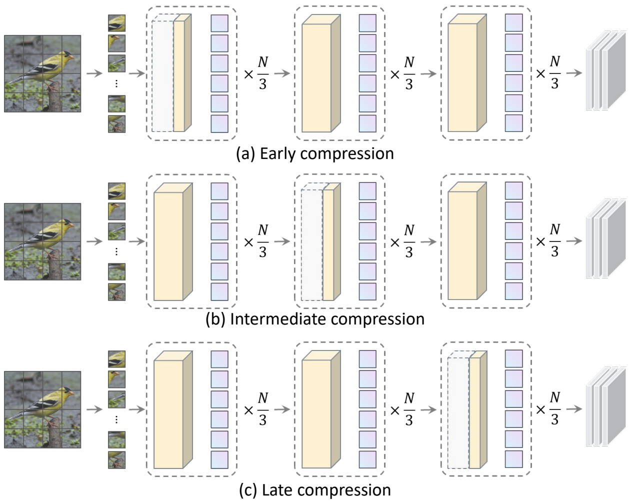

Figure 1. Illustration of applying low-rank approximation on different positions of a typical ViT model.

Visual Grounding (VG) localizes the accurate regions based on a textual query. We conducted experiments on the RefCOCO+ dataset [39], wherein no bounding box annotations are available (weakly-supervised setting). During inference, we extended Grad-CAM to obtain heatmaps and leveraged them to rank the detected proposals provided by [40]. Specifically, we first predicted the region referred to by the given query. We estimated the accuracy according to the intersection over union (IOU) ratio between the true and predicted bounding box, and reported this metric on three settings [39]: Val, TestA, and TestB.

Visual Question Answering (VQA). We used the popular VQA v2 dataset [9] and adopted accuracy as the key metric [10]. Due to the submission number limitation of the leaderboard website, we merely evaluated the baseline and our model and reported the final results once.

3.5. Vision-Language Baseline Model

Pre-training. We followed most of the settings with ALBEF [17]. Specifically, we pre-trained our model on four publicly available large-scale datasets: MSCOCO caption [18], Visual Genome [16], SBU [21] and Conceptual-Captions [27]. In total, there are 4M image-text pairs of these datasets.

We adopted the BERT-base model for text processing and ViT-base for visual feature extraction. We pre-trained the model for **10** epochs using a batch size of 128 on 4 GPUs. During pre-training, we took random image crops of resolution 256×256 as input, and also applied random augmentation to maintain visual feature diversity. For fine-tuning, we increased the image resolution to 384×384. In addition, the size of the queue used for image-text contrastive learning is set as 65,536 following ALBEF [17].

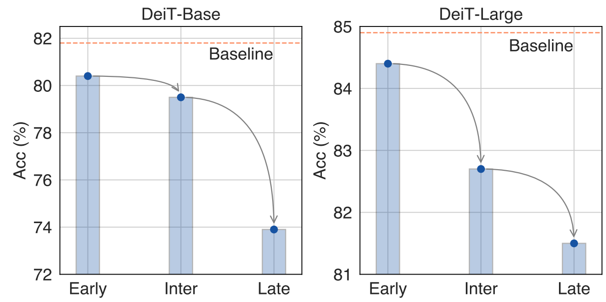

Figure 2. Model performance of different low-rank approximation positions on two DeiT models.

We also fixed the loss weights α and β as 0.1 and 1.0 during pre-training, respectively.

An interesting finding. It is worth noting that we found one special case when applying low-rank approximation on vision-language models. In general, the word embedding layers account for arguably the largest part of the parameters. Nevertheless, using low-rank approximation on the word embedding matrix leads to a computation-parameter dilemma. On the one hand, this operation will decompose the matrix with a large rank into two low-rank matrices, resulting in fewer parameters in a Transformer model. On the other hand, it cannot achieve efficient computation as the process from input tokens to word embedding is actually a look-up action rather than matrix multiplication! In view of this, we omitted the approximation on the word embedding layer in our implementation.

Down-stream fine-tuning. We strictly followed the experimental settings used by ALBEF and kept most of them untouched. We employed a smaller batch size and reproduced the results of the baseline model.

4. Experimental Results

Compression on different positions. We conducted experiments to test whether the low-rank approximation position affects the final model performance. To this end, we split a ViT model into three stages and applied low-rank approximation to every single stage (as shown in Fig. 1). The results are shown in Fig. 2. We can see that compressing deeper layers often results in worse model performance.

More knowledge distillation choice. We also studied other knowledge distillation choices during our trial-and-error stage. Specifically, we introduced the logit-based KD objective into our model and trained it with the other loss functions. From the results in Table 1, we can see that this loss does not lead to more performance improvements. We believe this is because the feat-based KD already aligns the feature distribution between the large pre-trained model and our low-rank model, thereby involving more objectives does not bring more benefits.

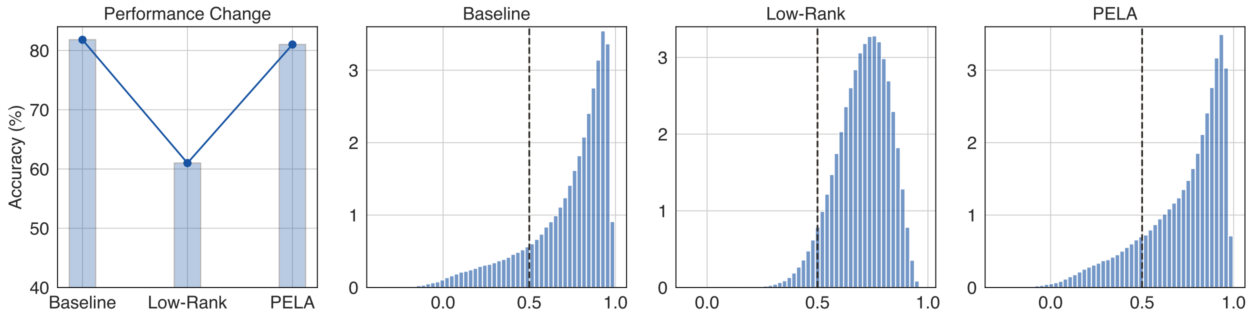

Figure 3. Performance comparison of three models and statistics of the instance-level feature similarity.

Table 1. Influence of the feat-based and logit-based knowledge distillation objectives.

Variant	DeiT-Base [29]
Original	80.25
+ feat-KD	80.96
+ logit-KD	80.85

Qualitative results. Fig. 3 demonstrates the performance comparison of the original model, the directly low-rank model, and our final PELA. We can observe that the compressed low-rank model does not effectively learn instance-level discriminative representation. One possible reason is that the learned features after low rank are confined in a narrow feature space (the similarity of the features is drastically increased as shown in the figure). After our PELA method, the feature similarity of each class becomes more consistent with that of the original model, thereby leading to improved model performance.

References

[1] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: visual question answering. In *ICCV*, pages 2425–2433. IEEE, 2015. 1

[2] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. In *ICLR*, 2023. 2

[3] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, pages 6154–6162. IEEE, 2018. 2

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229. Springer, 2020. 1

[5] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. UNITER: universal image-text representation learning. In *ECCV*, pages 104–120. Springer, 2020. 2

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186. ACL, 2019. 1, 2

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*. OpenReview.net, 2021. 1

[8] Yuxin Fang, Bencheng Liao, Xinggang Wang, Jiemin Fang, Jiyang Qi, Rui Wu, Jianwei Niu, and Wenyu Liu. You only look at one sequence: Rethinking transformer in vision through object detection. In *NeurIPS*, pages 26183–26197, 2021. 1

[9] Yash Goyal, Tejas Khot, Aishwarya Agrawal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. *IJCV*, 127(4):398–414, 2019. 3

[10] Yangyang Guo, Liqiang Nie, Zhiyong Cheng, Qi Tian, and Min Zhang. Loss re-scaling VQA: revisiting the language prior problem from a class-imbalance view. *TIP*, 31:227–238, 2022. 3

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969. IEEE, 2017. 2

[12] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross B. Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, pages 15979–15988. IEEE, 2022. 1

[13] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *ICLR*. OpenReview.net, 2022. 2

[14] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling BERT for natural language understanding. In *Findings of EMNLP*, pages 4163–4174. ACL, 2020. 1, 2

[15] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *ICML*, pages 5583–5594. PMLR, 2021. 1

[16] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and

Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 123(1): 32–73, 2017. 3

[17] Junnan Li, Ramprasaath R. Selvaraju, Akhilesh Gotmare, Shafiq R. Joty, Caiming Xiong, and Steven Chu-Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. In *NeurIPS*, pages 9694–9705, 2021. 1, 2, 3

[18] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *ECCV*, pages 740–755. Springer, 2014. 1, 2, 3

[19] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 9992–10002. IEEE, 2021. 1, 2

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

[21] Vicente Ordonez, Girish Kulkarni, and Tamara L. Berg. Im2text: Describing images using 1 million captioned photographs. In *NIPS*, pages 1143–1151, 2011. 3

[22] Chunghyun Park, Yoonwoo Jeong, Minsu Cho, and Jaesik Park. Fast point transformer. In *CVPR*, pages 16928–16937. IEEE, 2022. 1

[23] Namuk Park and Songkuk Kim. How do vision transformers work? In *ICLR*. OpenReview.net, 2022. 1

[24] Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *ICCV*, pages 2641–2649. IEEE, 2015. 2

[25] 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. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763. PMLR, 2021. 1

[26] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. LAION-400M: open dataset of clip-filtered 400 million image-text pairs. *CoRR*, abs/2111.02114, 2021. 1

[27] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, pages 2556–2565. ACL, 2018. 1, 3

[28] Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. FLAVA: A foundational language and vision alignment model. In *CVPR*, pages 15617–15629. IEEE, 2022. 1

[29] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, pages 10347–10357. PMLR, 2021. 1, 2, 4

[30] Hugo Touvron, Matthieu Cord, and Hervé Jégou. Deit iii: Revenge of the vit. In *ECCV*, pages 516–533. Springer, 2022. 2

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017. 1

[32] Cong Wang, Hongmin Xu, Xiong Zhang, Li Wang, Zhitong Zheng, and Haifeng Liu. Convolutional embedding makes hierarchical vision transformer stronger. In *ECCV*, pages 739–756. Springer, 2022. 1

[33] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *ICCV*, pages 22–31. IEEE, 2021. 1

[34] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, pages 418–434. Springer, 2018. 2

[35] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *NeurIPS*, pages 12077–12090, 2021. 1

[36] Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. Visual entailment: A novel task for fine-grained image understanding. *CoRR*, abs/1901.06706, 2019. 1

[37] Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. Visual entailment: A novel task for fine-grained image understanding. *CoRR*, abs/1901.06706, 2019. 2

[38] Hongwei Xue, Peng Gao, Hongyang Li, Yu Qiao, Hao Sun, Houqiang Li, and Jiebo Luo. Stare at what you see: Masked image modeling without reconstruction. In *CVPR*, pages 22732–22741. IEEE, 2023. 1, 2

[39] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C. Berg, and Tamara L. Berg. Modeling context in referring expressions. In *ECCV*, pages 69–85. Springer, 2016. 3

[40] Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L. Berg. Mattnet: Modular attention network for referring expression comprehension. In *CVPR*, pages 1307–1315. IEEE, 2018. 3

[41] Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. Metaformer is actually what you need for vision. In *CVPR*, pages 10809–10819. IEEE, 2022. 1

[42] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *CVPR*, pages 19291–19300. IEEE, 2022. 1

[43] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *IJCV*, 127: 302–321, 2019. 2

[44] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: deformable transformers for end-to-end object detection. In *ICLR*. OpenReview.net, 2021. 1