Vision Transformer with Deformable Attention
Supplementary Materials

1. DAT and Deformable DETR

In this section, we provide a detailed comparision between our proposed deformable attention and the direct adaptation from the deformable convolution [4], which is also known as the multiscale deformable attention in Deformable DETR [20].

First, our deformable attention serves as a feature extractor in the vision backbones while the one in Deformable DETR which replaces the vanilla attention in DETR [1] with a linear deformable attention, plays the role of the detection head. Second, the m-th head of query q in the attention in Deformable DETR with M heads in a single scale is formulated as

$$z_q^{(m)} = \sum_{k=1}^{K} A_{qk}^{(m)} W_v \phi(x; p_q + \Delta p_{qk}^{(m)}), \tag{1}$$

where K key points are sampled from the input features, mapped by W_v and then aggregated by attention weights $A_{qk}^{(m)}$. Compared to our deformable attention (Eq.(9) in the paper), this attention weights is learned from x by a linear projection, *i.e.* $A_{qk}^{(m)} = \sigma(W_{\text{att}} x)$, where $W_{\text{att}} \in \mathbb{R}^{C \times MK}$ is the weight matrix to predict the attention weights of each key k and head m, after which a softmax function σ is applied to the dimensions of K keys to normalize the attention score. In fact, the attention weights are predicted directly by queries instead of measuring the similarities between queries and keys. If we change the σ function to a sigmoid, this will be a variant of modulated deformable convolution [19], hence this deformable attention is more similar to convolution rather than attention.

Third, the deformable attention in Deformable DETR is not compatible to the dot-product attention for its enormous memory consumption mentioned in Sec.3.2 in the paper. Therefore, the linear predicted attention is used to avoid computing dot products and a smaller number of keys $K = 4$ is also adopted to reduce the memory cost.

To experimentally validate our claim, we replace our deformable attention modules in DAT with the modules in [20] to verify that the naive adaptation is inferior for vision backbone. The comparison results are shown in Table 1. To obtain a Deformable DETR under low memory budget, we reduce the number of keys to 16. Comparing the first row and the fourth row, our model achieves 1.2% better performance with similar memory cost. Comparing the third and last row, we can see that the D-DETR attention with the same number of keys as DAT consumes 2.6× memory and 1.3× FLOPs, while the performances are still lower than DAT.

Attn	Stage 3 #Key	Stage 4 #Key	FLOPs	#Param	Memory	IN-1K Acc.
D-DETR	16	16	4.44G	27.95M	13.9GB	80.6
D-DETR	49	49	4.83G	31.15M	18.8GB	80.7
D-DETR	196	49	6.16G	37.26M	37.9GB	79.2
DAT	49	49	4.38G	28.32M	12.5GB	81.8
DAT	196	49	4.59G	28.32M	14.4GB	**82.0**

Table 1. Comparisons of the deformable attention in DAT with that in [20] under different compuational budgets. The GPU memory cost is measured in a forward pass with a batch size of 64.

2. Adding Convolutions to DAT

Recent works [2,5,12,13] have proved that adopting convolution layers in the Vision Transformer architecture can further improve model performances. For example, using convolutional patch embedding can generally boost model performances by $0.5\% \sim 1.0\%$ on ImageNet classification tasks. It is worth noticing that our proposed DAT can readily combine with these techniques, while we maintain the convolution-free architecture in the main paper to perform fair comparison with baselines.

To fully explore the capacity of DAT, we substitute the patch embedding layers in the original model with strided and overlapped convolutions. The comparison results are shown in Table 2, where baseline models have similar modifications. It is shown that our model with additional convolution modules achieve 0.7% improvement comparing to the original version, and consistently outperform other baselines.

ImageNet-1K Classification			
Method	FLOPs	#Param	Top-1 Acc.
CvT-13 [13]	4.5G	20M	81.6
CoAt-Lite Small [14]	4.0G	20M	81.9
CeiT-S [15]	4.8G	24M	82.0
PVTv2-B2 [12]	4.0G	25M	82.0
CoAt Small [14]	12.6G	22M	82.1
RegionViT-S [2]	5.3G	31M	82.5
DAT-T	4.6G	28M	82.0
DAT-T*	4.8G	30M	**82.7**

Table 2. Comparisons of DAT with other vision transformer backbones on FLOPS, parameters, accuracy on the ImageNet-1K classification task. DAT-T refers to the original version. DAT-T* refers to the model with convolutional patch embeddings.

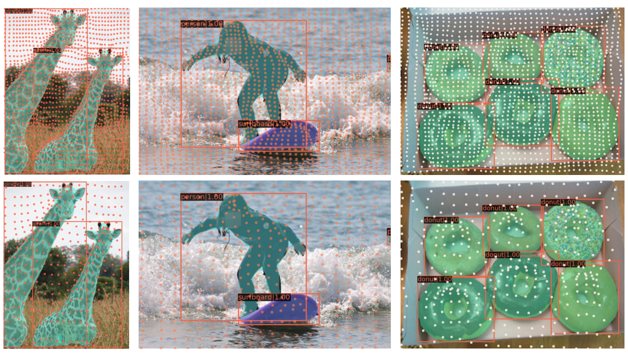

Figure 1. Visualizations on COCO [7] of learned sampling locations in deformable attention at Stage 3 (first row) and Stage 4 (second row) of DAT. The orange and yellow points show one group of deformed points. The detection bounding boxes and segmentation masks are also presented to indicate the targets.

3. More Visualizations

We visualize examples of learned deformed locations in our DAT to verify the effectiveness of our method. As illustrated in Figure 1, the sampling points are depicted on the top of the object detection boxes and instance segmentation masks, from which we can see that the points are shifted to the target objects. In the left column, the deformed points are contracted to two target giraffes, while other points are keeping a nearly uniform grid with small offsets. In the middle column, the deformed points distribute densely among the person's body and the surfing board both in the two stages. The right column shows the deformed points focus well to each of the six donuts, which shows our model has the ability to better model geometric shapes even with multiple targets. The above visualizations demonstrate that DAT learns meaningful offsets to sample better keys for attention to improve the performances on various vision tasks.

We also provide visualization results of the attention map given specific query tokens, and compare with Swin Transformer [8] in Figure 2. We show key tokens with the highest

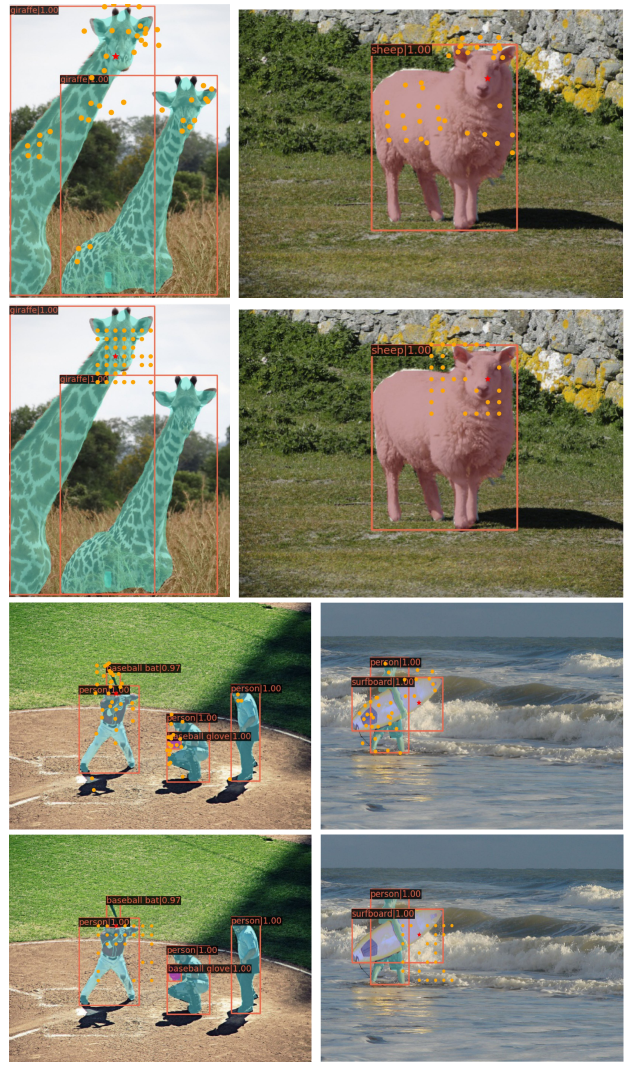

Figure 2. Visualizations on COCO [7] validation set. The red star denotes a query point, the orange dots are the keys with higher attention scores in the last layer. The images in the first and third rows depict our DAT attention and Swin Transformers' [8] are shown in the second and fourth rows. The detection bounding boxes and segmentation masks are also presented to indicate the targets.

attention values. It can be observed that our model focus on the more relevent part. As a showcase, our model allocates most attention to foreground objects, *e.g.*, both gireffas in the first row. On the other hand, the region of interests in Swin Transformer is comparably local and fail to distinguish foreground from background, which is depicted in the last surfboard.

4. Training Details of DAT

We use AdamW [9] optimizer to train our models for 300 epochs with a cosine learning rate decay. The basic learning rate for a batch size of 1024 is set to 1×10^{-3}, and then linearly scaled w.r.t. the batch size. To stabilize training procedures, we schedule a linear warm-up of learning rate from 1×10^{-6} to the basic learning rate, and for a better convergence the cosine decay rule is applied to gradually decrease the learning rate to 1×10^{-7} during training. We follow DeiT [11] to set the advanced data augmentation, including RandAugment [3], Mixup [17] and CutMix [16] to avoid overfitting. In addition, stochastic depth [6] and weight decay of 0.05 are also applied, in which the stochastic depth degree is chosen 0.2, 0.3 and 0.5 for the tiny, small and base model, respectively. We do not adopt EMA [10], random erasing [18] and the vanilla drop out, which does not improve the training of Vision Transformers, as verified in [8, 11]. In terms of larger resolution finetuning, we finetune our DAT-B using AdamW optimizer with a cosine scheduled learning rate 4×10^{-6} for 30 epochs. We set the stochastic depth rate to 0.5 and lower the weight decay to 1×10^{-8} to keep the regularization.

References

[1] 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

[2] Chun-Fu Chen, Rameswar Panda, and Quanfu Fan. Regionvit: Regional-to-local attention for vision transformers. *arXiv preprint arXiv:2106.02689*, 2021. 1, 2

[3] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPRW*, pages 702–703, 2020. 3

[4] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, pages 764–773, 2017. 1

[5] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows, 2021. 1

[6] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *ECCV*, pages 646–661. Springer, 2016. 3

[7] 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 *ECCV*, pages 740–755. Springer, 2014. 2

[8] 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. *ICCV*, 2021. 2, 3

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

[10] Boris T Polyak and Anatoli B Juditsky. Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization*, 30(4):838–855, 1992. 3

[11] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers & distillation through attention. In *ICML*, volume 139, pages 10347–10357, July 2021. 3

[12] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvtv2: Improved baselines with pyramid vision transformer. *arXiv preprint arXiv:2106.13797*, 2021. 1, 2

[13] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. *arXiv preprint arXiv:2103.15808*, 2021. 1, 2

[14] Weijian Xu, Yifan Xu, Tyler Chang, and Zhuowen Tu. Co-scale conv-attentional image transformers. *arXiv preprint arXiv:2104.06399*, 2021. 2

[15] Kun Yuan, Shaopeng Guo, Ziwei Liu, Aojun Zhou, Fengwei Yu, and Wei Wu. Incorporating convolution designs into visual transformers. *arXiv preprint arXiv:2103.11816*, 2021. 2

[16] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, pages 6023–6032, 2019. 3

[17] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 3

[18] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *AAAI*, volume 34, pages 13001–13008, 2020. 3

[19] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *CVPR*, pages 9308–9316, 2019. 1

[20] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 1