A1. Experimental Settings for Ablation

This section describes the experimental setups for ablation, including models of SwinV2-T, SwinV2-S, and SwinV2-B, as well as tasks of ImageNet-1K image classification, COCO object detection, and ADE20K semantic segmentation.

A1.1. ImageNet-1K Pre-Training

All ablation studies use the ImageNet-1K image classification task for pre-training. We use an input image size (window size) of 256×256 (8×8)[1]. Following [12], we use an AdamW [13] optimizer with 300 epochs using a cosine decay learning rate scheduler with 20 epochs of linear warm-up. A batch size of 1024, an initial learning rate of 1×10^{-3}, a weight decay of 0.05, and gradient clipping with a max norm of 5.0 are used. Augmentation and regularization strategies include RandAugment [7], Mixup [16], Cutmix [15], random erasing [17] and stochastic depth [11]. An increasing degree of stochastic depth augmentation is employed for larger models, i.e., $0.2, 0.3, 0.5$ for tiny, small, and base models, respectively.

A1.2. Fine-tuning on Down-stream Tasks

ImageNet-1K image classification For ImageNet-1K image classification experiments, we conduct a fine-tuning step if the input image resolution is larger than that in the pre-training step. The fine-tuning lasts for 30 epochs, with an AdamW [13] optimizer, a cosine decay learning rate scheduler with an initial learning rate of 4×10^{-5}, a weight decay of 1×10^{-8}, and the same data augmentation and regularizations as those in the first stage.

COCO object detection We use the cascade mask R-CNN [3, 10] implemented in mmdetection [5] for object detection. In training, a multi-scale augmentation strategy [5] is adopted, with the shorter side between 480 and 800 and the longer side of 1333. The window size is set as 16×16. Other details are: an AdamW [13] optimizer with an initial learning rate of 1×10^{-4}, a weight decay of 0.05, a batch size of 16, and a 3× scheduler.

ADE20K semantic segmentation The image size (window size) we use is 512×512 (16×16). In training, we employ an AdamW optimizer [13] with an initial learning rate of 4×10^{-5}, a weight decay of 0.05, a learning rate scheduler that uses linear learning rate decay and a linear warm-up of 1,500 iterations. Models are trained with batch size of 16 for 160K iterations. We follow the mmsegmentation codebase [6] to adopt augmentations of random horizontal flipping, random re-scaling within ratio range [0.5, 2.0] and a random photometric distortion. Stochastic depth with a ratio of 0.3 is applied for all models. All experiments use a layer-wise learning rate decay [1] of 0.95.

[1] Most of our experiments use the window size of an even number so that the window shifting offset is divisible by the window size. Nevertheless, a window size of an odd number also works well, just like the case in the original Swin Transformer (7×7).

A2. Experimental Settings for System-Level Comparison

A2.1. SwinV2-B and SwinV2-L Settings

Tables 2, 3, and 4 show the results for SwinV2-B and SwinV2-L. For these experiments, we first perform ImageNet-22K pre-training and then fine-tune the pre-trained models on downstream recognition tasks.

ImageNet-22K pre-training Both models use an input image size (window size) of 192×192 (12×12). We employ an AdamW optimizer [13] for 90 epochs using a cosine decayed learning rate scheduler with 5-epoch linear warm-up. We use a batch size of 4096, an initial learning rate of 0.001, a weight decay of 0.1, and gradient clipping with a max norm of 5.0. Augmentation and regularization strategies include RandAugment [7], Mixup [16], Cutmix [15], random erasing [17] and stochastic depth [11] with ratio of 0.2.

ImageNet-1K image classification We consider input image sizes of 256×256 and 384×384. The training length is set as 30 epochs, with a batch size of 1024, a cosine decay learning rate scheduler with an initial learning rate of 4×10^{-5}, and a weight decay of 1×10^{-8}. The ImageNet-1K classification weights are also initialized from the corresponding ones in the ImageNet-22K model.

COCO object detection We adopt HTC++ [4, 12] for experiments. In data pre-processing, Instaboost [8], a multi-scale training [9] with an input image size of 1536×1536, a window size of 32×32, and a random scale between $[0.1, 2.0]$ are used. An AdamW optimizer [13] with an initial learning rate of 4×10^{-4} on batch size of 64, a weight decay of 0.05, and a 3× scheduler are used. The backbone learning rate is set as $0.1\times$ of the head learning rate. In inference, soft-NMS [2] is used. Both single-scale and multi-scale test results are reported.

ADE20K semantic segmentation The input image size (window size) is set to 640×640 (40×40). We employ an AdamW [13] optimizer with an initial learning rate of 6×10^{-5}, a weight decay of 0.05, and a linear decayed learning rate scheduler with 375-iteration linear warm-up. The model is trained with batch size of 64 for 40K iterations. We follow the default settings in mmsegmentation [6] for data augmentation, including random horizontal flipping, random re-scaling within ratio range $[0.5, 2.0]$ and random photometric distortion. A stochastic depth with ratio of 0.3 is applied.

A2.2. SwinV2-G Settings

ImageNet-22K-ext dataset collection The ImageNet-22K-ext dataset is collected by querying class names on a public search engine of BING. To get more images, we expand the class queries by prompts such as "a photo of", the super-class such as "a type of dog", or a detailed description such as "any bird associated with night ...". There is no human re-labelling process, and so the labels are very noisy. The newly collected extensions also have a class imbalance issue, like that of the original ImageNet-22K dataset, but is lighter. By using this noisy dataset for pre-training, we observed comparable top-1 accuracy on ImageNet-1K than that using the original ImageNet-22K dataset on a Swin-L, and higher accuracy (about 1%) on a Swin-H model.

Stage-1 self-supervised pre-training The model is first pre-trained 20 epochs on the ImageNet-22K-ext dataset (70 million images) using a self-supervised learning approach [14]. To reduce the overhead of experimentation, we used a smaller image size of 192×192. The model was trained using the AdamW [13] optimizer, which has a 30,000-step linear warm-up and follows a cosine decayed learning rate scheduler. We use gradient clipping with a batch size of 9216, an initial learning rate of 1.4×10^{-3}, a weight attenuation of 0.1, and a maximum norm of 100.0. In data augmentation, we adopt a light version: random resize cropping with scale range [0.67,1], an aspect ratio of [3/4, 4/3], followed by random flip and color normalization steps.

Stage-2 supervised pre-training The model is further pre-trained using the class labels on the ImageNet-22K-ext dataset. We employ an AdamW optimizer [13] for 30 epochs, using a cosine decayed learning rate scheduler with 20,000 steps of linear warm-up. We use a batch size of 9216, an initial learning rate of 1.4×10^{-3}, a layer-wise learning rate decay of 0.87, a weight decay of 0.1, and gradient clipping with a max norm of 100.0. Augmentation and regularization strategies include RandAugment [7], random erasing [17] and a stochastic depth [11] ratio of 0.3.

Fine-tuning on ImageNet-1K image classification We experimented with an input image size of 640×640. The AdamW optimizer [13] is employed for 10 epochs, using a cosine decayed learning rate scheduler and a 2-epoch linear warm-up. We use a batch size of 576, an initial learning rate of 2.1×10^{-5}, a weight decay of 0.1, and gradient clipping with a max norm of 100.0. Augmentation and regularization strategies include RandAugment [7], random erasing [17] and a stochastic depth [11] ratio of 0.5.

In evaluation, we test top-1 accuracy for both ImageNet-1K V1 and V2.

Fine-tuning on COCO object detection We conduct an intermediate fine-tuning phase using the Objects-365 V2 dataset. In this phase, we remove the mask branch of the HTC++ framework [4, 12] because there are no mask annotations on this dataset. The input image resolution and window size are set as $[800, 1024]$ and 32×32, respectively. In training, we use an AdamW optimizer [13] with an initial learning rate of 1.2×10^{-3}, a weight decay of 0.05, and a batch size of 96. The training length is set to 67,500 steps.

We then fine-tune the HTC++ model on the COCO dataset, with the mask branch randomly initialized and other model weights loaded from the Objects-365-V2 pre-trained model. In this training phase, the input image resolution is set to 1536×1536, with a multi-scale ratio of $[0.1, 2.0]$. The window size is set 32×32. We use the AdamW optimizer [13] for 45,000 steps, with an initial learning rate of 6×10^{-4}, a weight decay of 0.05, and a batch size of 96.

In testing, Soft-NMS [2] is used. Both window sizes of 32×32 and 48×48 are considered.

Fine-tuning on ADE20K semantic segmentation We set the input image size (window size) as 640×640 (40×40). An AdamW optimizer [13] is employed, with an initial learning rate of 4×10^{-5}, a weight decay of 0.05, a linear decayed learning rate scheduler with 80K iterations, a batch size of 32, and a linear warm-up of 750 iterations. For augmentations, we follow the default settings in mmsegmentation to include random horizontal flipping, random re-scaling within ratio range $[0.5, 2.0]$ and random photometric distortion. The stochastic depth ratio is set as 0.4.

Fine-tuning on Kinetics-400 video action recognition We adopt a 2-stage fine-tuning process. In the first stage, an input resolution of 256×256×8 with 16×16×8 window size is used. We employ the AdamW optimizer for 20 epochs using a cosine decayed learning rate scheduler with 2.5-epoch linear warm-up. Other training hyper-parameters include: a batch-size of 80, an initial learning rate of 3.6×10^{-4}, and a weight decay of 0.1.

In the second stage, we further fine-tune the model with a larger input video resolution (320×320×8, window size 20×20×8). We use the AdamW optimizer for 5 epochs and a cosine decayed learning rate scheduler with 1-epoch linear warm-up. We set the batch size to 64, the initial learning rate of 5×10^{-5}, and a weight decay of 0.1.

A3. Learnt Relative Position Bias by Different Approaches

Figure 1 visualizes the relative position bias matrices ($\hat{B} \in \mathbb{R}^{(2M-1)\times(2M-1)}$) learnt using different bias computation approaches on a SwinV2-T model. The bias matrices of the 3 heads in the first block are visualized. The left shows the bias matrices learnt by using an input image size of 256×256 and a window size of 8×8. The right shows the bias matrices after fine-tuning on a larger input image resolution of 512×512 and a larger window size of 16×16. It turns out that the bias matrices learnt by two CPB (continuous position bias) approaches are smoother than that learnt by P-RPB (parameterized relative position bias). Figure 2 shows more examples for the last block of this model.

References

[1] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers, 2021. 1

[2] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S. Davis. Soft-nms – improving object detection with one line of code. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 1, 2

[3] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6154–6162, 2018. 1

[4] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, et al. Hybrid task cascade for instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4974–4983, 2019. 1, 2

[5] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 1

[6] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation, 2020. 1

[7] 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. 1, 2

[8] Hao-Shu Fang, Jianhua Sun, Runzhong Wang, Minghao Gou, Yong-Lu Li, and Cewu Lu. Instaboost: Boosting instance segmentation via probability map guided copy-pasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 682–691, 2019. 1

[9] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. Nas-fpn: Learning scalable feature pyramid architecture for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7036–7045, 2019. 1

[10] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1

[11] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *European conference on computer vision*, pages 646–661. Springer, 2016. 1, 2

[12] 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, 2021. 1, 2

[13] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 1, 2

[14] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *Tech report*, 2022. 2

[15] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6023–6032, 2019. 1

[16] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 1

[17] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13001–13008, 2020. 1, 2

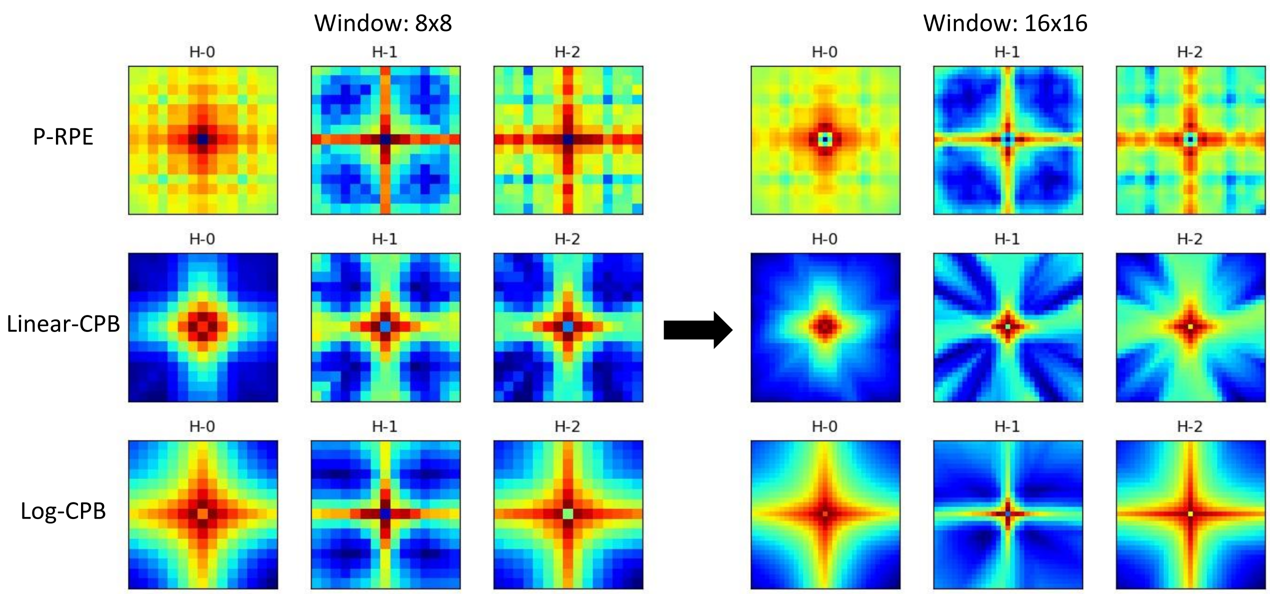

Figure 1. Visualization of the learnt relative position bias matrices by different approaches, using a SwinV2-T model and the 3 heads in the first block. Left: the bias matrices after pre-training on a 256×256 image and an 8×8 window; Right: the bias matrices after fine-tuning, using a 512×512 image size and a 16×16 window size. H-x indicates the x-th head.

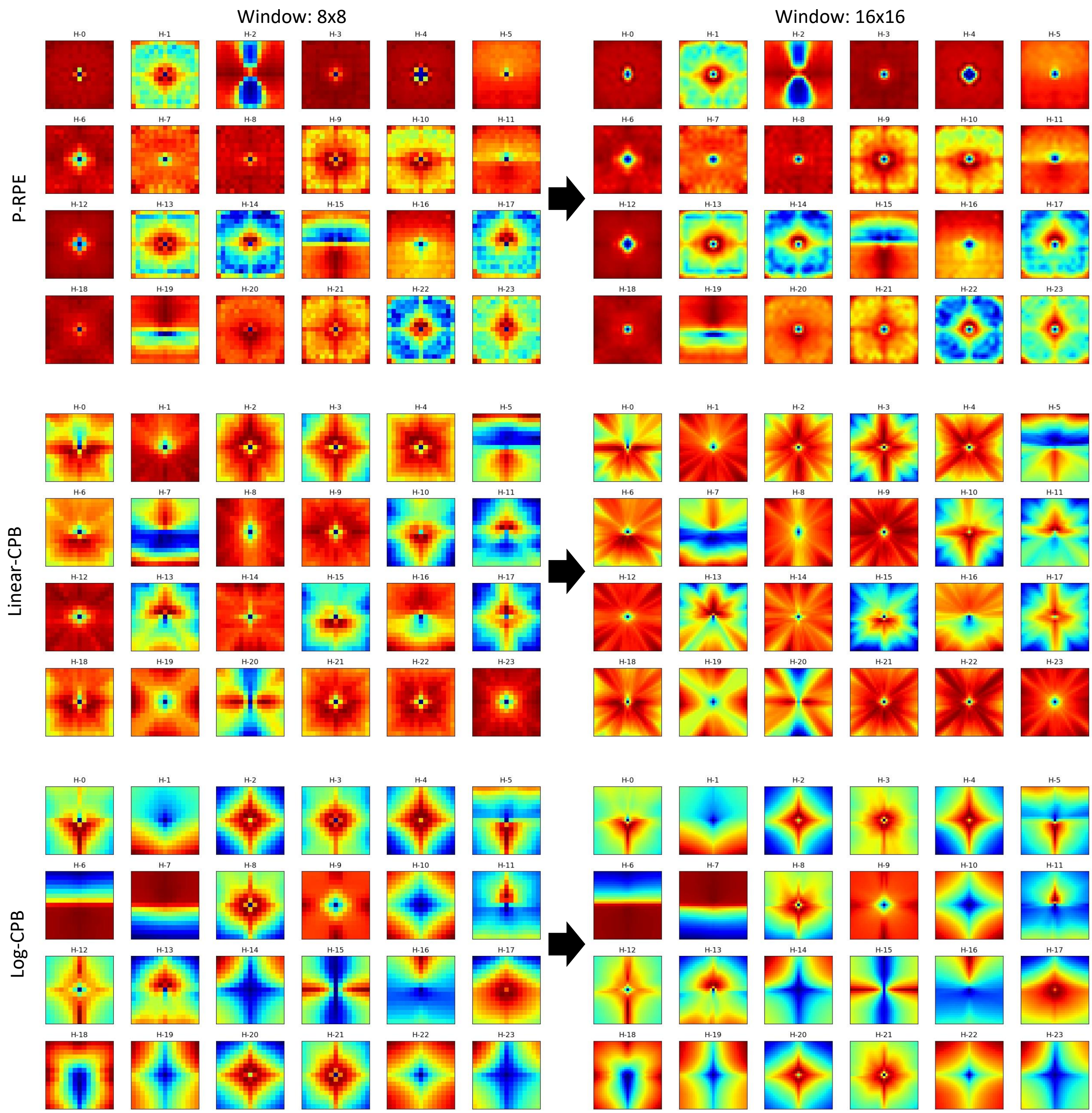

Figure 2. Visualization of the learnt relative position bias matrices by different approaches, using a SwinV2-T model and the 24 heads in the last block. Left: the bias matrices after pre-training on a 256×256 image and an 8×8 window; Right: the bias matrices after fine-tuning using a 512×512 image size and a 16×16 window size. H-x indicates the x-th head.