Supplementary Materials for *SimMIM*: A Simple Framework for Masked Image Modeling

Zhenda Xie[1*] Zheng Zhang[2*] Yue Cao[2*]
Yutong Lin[3] Jianmin Bao[2] Zhuliang Yao[1] Qi Dai[2] Han Hu[2*]
[1]Tsinghua University [2]Microsoft Research Asia [3]Xi'an Jiaotong University
{t-zhxie,zhez,yuecao,t-yutonglin,jianmin.bao,t-zhuyao,qid,hanhu}@microsoft.com

Model	Swin-B	Swin-L	SwinV2-H
Base Channel	128	192	352
Depths	{2,2,18,2}	{2,2,18,2}	{2,2,18,2}
Params	88M	197M	658M
Pre-training			
Input Size	192	192	192
Window Size	6	12	12
FLOPs	11.3G	26.0G	86.2G
Fine-tuning			
Input Size	224	224	224
Window Size	7	14	14
FLOPs	15.4G	35.8G	118.1G

Table 1. Detailed architecture specifications.

1. Detailed Architectures

The detailed architecture specifications are shown in Table 1, where an input image size of 192×192 is used for pre-training and 224×224 is used in fine-tuning.

2. The Effect of Learning Rate Schedulers

In our ablation study, we follow common practice [1, 3] to use a *cosine* learning rate scheduler. In our scaling up experiments, we adopt a *step* learning rate scheduler to reduce experimental overheads of potentially studying the effects of different training lengths.

In this section, we investigate the effects of different schedulers on fine-tuning accuracy. Both schedulers adopt 10-epoch linear warm-up. For the *step* learning rate scheduler, the base learning rate is set as 8e-4, and is decayed by a factor of 10 at 90% and 95% of the total training length. For this comparison, we follow the default settings used in ablation, except that the scheduler is changed. As shown in Table 2, the *step* scheduler performs marginally better than the *cosine* scheduler, by +0.1% using a 100-epoch pre-training, and by +0.3% using a longer 300-epoch training procedure.

*Equal. Zhenda, Yutong, Zhuliang are long-term interns at MSRA.

lr scheduler	100 epochs	300 epochs
cosine	82.8	83.0
step	82.9	83.3

Table 2. The effects of different learning rate schedulers.

3. Results on Downstream Tasks

In this section, we add more results on several downstream tasks, including iNaturalist (iNat) 2018 classification, COCO object detection and ADE20K semantic segmentation.

3.1. Detailed Settings

iNaturalist 2018 classification iNaturalist [10] 2018 is a long-tail image classification dataset with more than 8,000 categories. It includes 437,513 training images and 24,426 validation images. We fine-tune the pre-trained models using an AdamW optimizer by 100 epochs. The fine-tuning hyper-parameters are: a batch size of 2048, a base learning rate of 1.6e-2, a weight decay of 0.05, $\beta_1 = 0.9$, $\beta_2 = 0.999$, a stochastic depth [6] ratio of 0.1, and a layer-wise learning rate decay of 0.9. We follow the same data augmentation strategies used in [1], including RandAug [2], Mixup [13], Cutmix [12], label smoothing [9], and random erasing [14].

COCO object detection A Mask-RCNN [5] framework is adopted and all models are trained with a $3\times$ schedule (36 epochs). We utilize an AdamW [7] optimizer with a learning rate of 6e-5, a weight decay of 0.05, and a batch size of 32. Following [4], we employ a large jittering augmentation (1024×1024 resolution, scale range [0.1, 2.0]). The window size for Swin-B is set to 7 and that for Swin-L and SwinV2-H models is 14.

ADE20K semantic segmentation Following [8], An UPerNet framework [11] is used following [8]. We

Head	ImageNet Top-1 Acc	iNat-2018 Top-1 Acc	COCO mAP^{box}	ADE20K mIoU
Linear	82.8	75.2	49.9	50.0
2-layer MLP	82.8	75.0	50.1	49.9
inverse Swin-T	82.4	74.9	49.8	49.4
inverse Swin-B	82.5	75.0	49.8	49.0

Table 3. More ablation studies on prediction head designing using iNat-2018, COCO and ADE20K.

Loss	Pred. Resol.	ImageNet Top-1 Acc	iNat-2018 Top-1 Acc	COCO mAP^{box}	ADE20K mIoU
8-bin	48^2	82.7	75.3	50.0	49.7
256-bin	48^2	82.3	74.6	49.7	49.3
iGPT	48^2	82.4	75.0	49.6	49.1
BEiT	-	82.7	75.2	50.1	48.8
ℓ_1	192^2	82.8	75.2	49.9	50.0

Table 4. More ablation studies on prediction targets using iNat-2018, COCO and ADE20K.

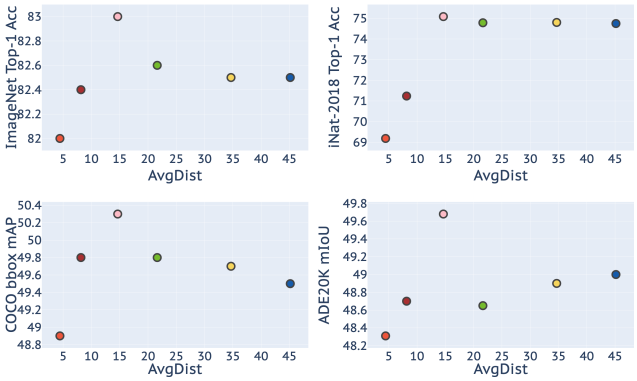

Figure 1. *AvgDist* (averaged distance of masked pixels to the nearest visible pixels) w.r.t. performance on ImageNet, iNat-2018, COCO and ADE20K.

use an AdamW [7] optimizer using the following hyper-parameters: a weight decay of 0.05, a batch size of 32, a layer-wise decay rate of 0.9, and a learning rate searching from 1e-4 and 3e-4. All models are trained for 80K iterations with an input resolution of 512×512 and a window size of 20. In inference, a multi-scale test using resolutions that are [0.75, 0.875, 1.0, 1.125, 1.25]× of 512×2048 is employed.

For ADE20K experiments, we initialized the segmentation models using model weights after supervised fine-tuning on ImageNet-1K, because its performance is superior to using the self-supervised pre-trained weights directly.

Backbone	Sup. COCO mAP^{box}	Sup. ADE20K mIoU	Ours COCO mAP^{box}	Ours ADE20K mIoU
Swin-B	50.2	50.4	52.3	52.8
Swin-L	50.9	50.0	53.8	53.5
SwinV2-H	50.2	49.8	54.4	54.2

Table 5. Scaling experiments with Swin on COCO and ADE20K.

3.2. Ablation Studies

Table 3 and 4 ablates the designs in SimMIM on the above additional down-stream tasks. We also copy the results of ImageNet-1K from the main body to these tables for reference.

Table 3 indicates that a lighter head (linear, 2-layer) is consistently better than the heavier heads (e.g. inverse Swin-T) on most tasks: +0.4% on ImageNet-1K, +0.3% on iNat-2018, and +0.6 on ADE20K. Table 4 suggests that our presented regression based prediction target (ℓ_1) could achieve on par or better performance than the well designed classification based ones.

We also use these additional down-stream tasks to verify different masking strategies, as shown in Figure 1. It turns out that the observations in Figure 3 of the main paper also hold: 1) the *AvgDist* measure is a good indicator for the learning effectiveness of masked image modeling; 2) an *AvgDist* of 15 is empirically good for masked image modeling.

3.3. Scaling Experiments

Table 5 shows the scaling performance using COCO object detection and ADE20K semantic segmentation. On Swin-B, Swin-L, and SwinV2-H, SimMIM achieves +2.1 / +2.9 / +4.2 mAP^{box} and +2.4 / +3.5 / +4.4 mIoU higher accuracy than its supervised counterparts, respectively. It indicates the broad effectiveness of the SimMIM approach. It also suggests that larger models benefit more from this approach.

4. More Results on Channel-wise Bin Color Discretization

Table 6 shows more results of using *channel-wise bin color discretization* as the prediction target, by varying bin numbers and prediction resolutions. We notice that the best accuracy for different bin numbers are achieved at different prediction resolutions: the 2-bin and 4-bin targets reach the best accuracy at a resolution of 192^2, and all other bin numbers reach the best accuracy at a low prediction resolution of 6^2. These results imply a moderately fine-grained target is encouraged for this classification based approach.

Pred. Resolution	Bin Num. (Top-1 acc %)					
	2	4	8	16	32	256
6^2	82.5	82.7	**82.8**	**82.9**	**82.8**	**82.4**
48^2	82.5	82.8	82.7	82.6	82.5	82.3
192^2	**82.7**	**82.9**	82.7	82.7	N/A	N/A

Table 6. More results of using *channel-wise bin color discretization* as the prediction target, by varying bin numbers and prediction resolutions. Swin-B and 100-epoch pre-training are used.

5. SimMIM with ConvNets

With the remarkable performance of SimMIM on Vision Transformers, we want to verify its effectiveness on versatile architectures. Here we adopt ResNet-50×4 as the base architecture. The overall training setup remains the same as that of Swin-Base. We use masked tokens to replace the original features after the stem of a 3×3 convolution of $stride = 2$ followed by a 2×2 max-pooling operator.

On ResNet-50×4, SimMIM achieves 81.6% top-1 accuracy on ImageNet-1K validation set using 300-epoch pre-training and 100-epoch fine-tuning, outperforming the supervised counterpart by +0.9% (vs. 80.7%). This indicates the generality of SimMIM.

References

[1] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021. 1

[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. 1

[3] 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 *International Conference on Learning Representations*, 2021. 1

[4] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D. Cubuk, Quoc V. Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2918–2928, June 2021. 1

[5] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017. 1

[6] 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

[7] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1, 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. *arXiv preprint arXiv:2103.14030*, 2021. 1

[9] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 1

[10] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8769–8778, 2018. 1

[11] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 418–434, 2018. 1

[12] 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

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

[14] 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