Masked Image Modeling with Local Multi-Scale Reconstruction

Haoqing Wang[1], Yehui Tang[1,2], Yunhe Wang[2,*], Jianyuan Guo[2], Zhi-Hong Deng[1,*], Kai Han[2]
[1]School of Intelligence Science and Technology, Peking University [2]Huawei Noah's Ark Lab
{wanghaoqing,yhtang,zhdeng}@pku.edu.cn, {yunhe.wang, kai.han}@huawei.com

	single-scale	multi-scale
global	83.0	82.8
local	83.0	83.3

(a) ViT-B

	single-scale	multi-scale
global	83.3	82.9
local	83.6	83.8

(b) Swin-B

Table 1. Decoupling local reconstruction and multi-scale supervisions. We report the top-1 fine-tuning accuracy on ImageNet-1K.

A. More experiments

In this section, we provide more experiments to support our work.

A.1. Decoupling local and multi-scale

To effectively guide the local layers, we propose multi-scale supervisions for multiple local reconstruction tasks. Here we decouple the local (or global) reconstruction and multi-scale (or single-scale) supervisions to further understand their relations. For global multi-scale reconstruction, we conduct at the top layer of encoder and use separate decoders to predict multiple supervisions of different scales. When the predictions have different scale with supervisions, we use deconvolution/pooling options to rescale them for matching supervisions. The results are shown in Table 1 and the pre-training length is 100 epochs. As we can see, global reconstruction prefers to single-scale supervisions, and using the supervisions of different scales to guide the same layer could make confusion. Conversely, local reconstruction prefers to multi-scale supervisions, and multiple local layers expect to learn the information of different scales. Local reconstruction can achieve better performance than the global one in most cases, and the gain increases when using multi-scale supervisions.

A.2. Comparison with feature fusion

To explicitly guide the lower layers, we conduct reconstruction task at multiple chosen local layers. The other method is fusing the features of multiple local layers to the top layer for global reconstruction [5]. It uses single-scale supervision for avoiding confusion. We compare our local reconstruction with this feature fusion method, and the results are shown in Table 2. Local reconstruction achieves consistently better performance than feature fusion on both columnar ViT [4] and pyramidal Swin [14]. For further exploration, we examine the gradient norm of each layer in the encoder during training process. Concretely, we load the checkpoint (state) of the median epoch in a complete training schedule and then calculate the gradient norm of parameters in each layer under this state. The results are shown in Fig. 1. For other middle epochs, we observe the same results. The lower layers have larger gradient norm than the upper ones due to the skip-connections in vision transformers. The skip-connections allow the lower layers to learn more quickly than the upper ones, which may be one reason for its significant effectiveness in various architectures [9, 19, 21]. Our local reconstruction can strengthen this characteristic and thus obtain better performance. Feature fusion essentially has the similar effect with the skip-connections. Besides, another advantage of our local construction is that it is compatible with multi-scale supervisions and thus can take advantage of richer information.

method	acc
global	83.0
global with fusion	83.0
local	83.3

(a) ViT-B

method	acc
global	83.3
global with fusion	83.5
local	83.8

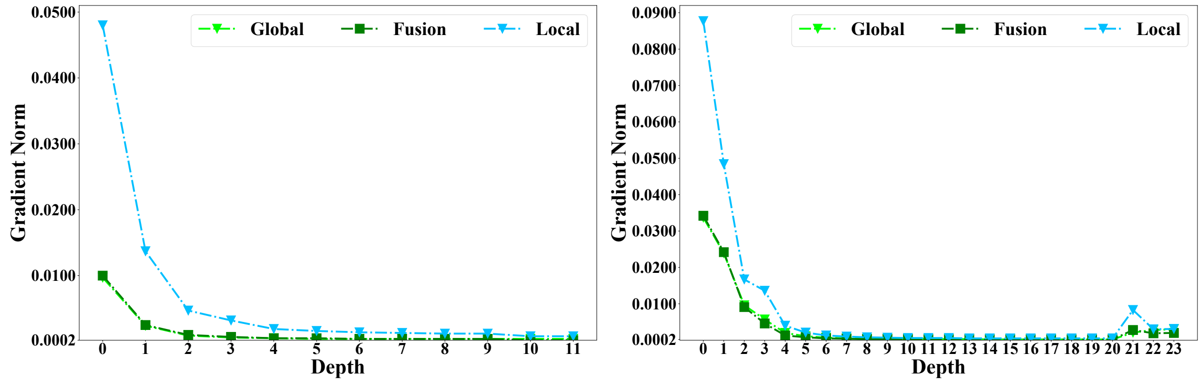
(b) Swin-B

Table 2. Top-1 fine-tuning accuracy on ImageNet-1K. We compare the global reconstruction, the global reconstruction with feature fusion and our local reconstruction.

A.3. Query-adaptive attention

In the main text, we use Normalized Mutual Information (NMI) [17] between query and key patches to examine how much the attention map depends on the query patch. Here we use another metric, the Kullback-Leibler divergence between the attention distributions of different query patches. Intuitively, when the attention map strongly depends on the query patch, the attention distributions of a pair of query patches should have large KL divergence. We

[*] Corresponding author.



(a) ViT-B

(b) Swin-B

Figure 1. Gradient norm of each layer in the encoder. We compare the global reconstruction, the global reconstruction with feature fusion and our local reconstruction, which are denoted as ‘Global’, ‘Fusion’ and ‘Local’ respectively.

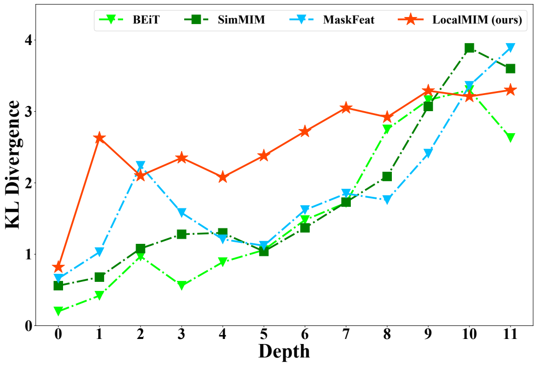


Figure 2. The KL divergence between attention distributions of different query patches at each layer of a pre-trained ViT-B backbone, averaged on all pairs of query patches.

calculate the average on all pairs of query patches at each layer and the results are shown in Fig. 2. As we expect, existing MIM models with global loss have small KL divergence at lower layers, which means the patches there have less query-adaptive attention. Relatively, the lower layers in our LocalMIM have larger KL divergence and the attention maps depend more strongly on the query patches.

B. GPU Hours

‘GPU Hours’ denotes the running time on single Tesla V100-32G GPU. For fair comparison, we estimate that of each model at the same machine with one Tesla V100-32G GPU, CUDA 10.2 and PyTorch 1.8. We pre-train each model for 10 epochs using its official released codes and default hyper-parameters, and then calculate the average running time per epoch. We find that each epoch takes similar time with each other during estimation, so pre-training 10 epochs is enough to estimate the GPU Hours per epoch. The batch size is an important factor that affects the running time, and we choose it from $\{32, 48, 64, 128, 256\}$ to take full advantage of GPU memory and computing capability. This estimation method avoids the interference of the communication time among multiple GPUs.

config	ViT	Swin
optimizer	AdamW [16]	
base learning rate	$2e^{-4}$	$1e^{-4}$
weight decay	0.05	
optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.95$	
batch size	2048 (B) / 4096 (L)	
learning rate schedule	cosine decay [15]	
augmentation	RandomResizedCrop	
input resolution	224×224	

Table 3. Pre-training setting on ImageNet-1K.

C. Implementation details

For ViT [4], we use the standard architecture with the sine-cosine positional embeddings and do not use relative positional encoding or layer scaling. For HOG feature, we set the number of orientation bins $\#bins = 18$ and the cell size is the same as the divided regions. We set the same weight to each local loss for simplicity. The pre-training and fine-tuning schedules mostly follow [7, 11].

Pre-training. The default setting is shown in Table 3. We use the simple data augmentation and do not use drop path or gradient clip. We use the linear learning rate scaling rule [6]: $lr = base\_lr \times batch\_size/256$. The warmup epoch [6] is set to 10 for pre-training 100 epochs, 40 for pre-training 400, 800 and 1600 epochs.

Fine-tuning on ImageNet-1K. The default fine-tuning setting is shown in Table 4. Most of the hyper-parameters are shared, except the peak learning rate, layer-wise learning rate decay and drop path rate, which are influenced by the

config	ViT		Swin	
	ViT-B	ViT-L	Swin-B	Swin-L
optimizer	AdamW			
peak learning rate	$\{2e^{-3}, 3e^{-3}, 4e^{-3}\}$		$\{3e^{-3}, 4e^{-3}, 5e^{-3}\}$	
weight decay	0.05			
optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.999$			
layer-wise lr decay [2]	$\{0.65, 0.75\}$		$\{0.80, 0.90\}$	
batch size	1024 (B) / 4096 (L)			
learning rate schedule	cosine decay			
fine-tuning epochs	100	50	100	100
warmup epochs	20	5	20	20
drop path [10]	0.1	0.2	0.1	0.3
augmentation	RandAug (9, 0.5) [3]			
label smoothing [18]	0.1			
mixup [23]	0.8			
cutmix [22]	1.0			
input resolution	224 × 224			

Table 4. Fine-tuning setting on ImageNet-1K.

backbones and the number of pre-training epochs.

Semantic segmentation on ADE20K. We use UperNet [20] with ViT-B backbone and follow the semantic segmentation code of [1, 7]. Concretely, we fine-tune end-to-end for 160K iterations using AdamW optimizer with the peak learning rate of $4e^{-4}$, weight decay of 0.05 and batch size of 16. The learning rate warmups with 1500 iterations and then decays with linear strategy. The model is trained with input resolution of 512×512 and uses bilinear positional embedding interpolate. We choose the out indices of feature maps as $[2, 4, 10, 12]$ and use FPN [12] to rescale them.

Object detection and segmentation on COCO. We fine-tune Mask R-CNN [8] on COCO [13] with Swin-B backbone. Following [11], we also use the code base and schedule from [14]. Concretely, the model is fine-tuned on COCO 2017 train split and evaluated on 2017 val split. We adopt the 3× fine-tuning schedule which trains the model for 36 epochs in total and decays the learning rate at the 27-th and 33-th epoch by a factor of 10. We use AdamW optimizer with the learning rate of $1e^{-4}$ and weight decay of 0.05.

References

[1] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. In *International Conference on Learning Representations*, 2022. 3

[2] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELECTRA: pre-training text encoders as discriminators rather than generators. In *ICLR*, 2020. 3

[3] 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. 3

[4] 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, 2

[5] Peng Gao, Teli Ma, Hongsheng Li, Jifeng Dai, and Yu Qiao. Convmae: Masked convolution meets masked autoencoders. *arXiv preprint arXiv:2205.03892*, 2022. 1

[6] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017. 2

[7] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022. 2, 3

[8] 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. 3

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[10] 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. 3

[11] Lang Huang, Shan You, Mingkai Zheng, Fei Wang, Chen Qian, and Toshihiko Yamasaki. Green hierarchical vision transformer for masked image modeling. *arXiv preprint arXiv:2205.13515*, 2022. 2, 3

[12] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 3

[13] 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 *European conference on computer vision*, pages 740–755. Springer, 2014. 3

[14] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 1, 3

[15] Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with warm restarts. In *ICLR*, 2017. 2

[16] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 2

[17] Alexander Strehl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, 3(Dec):583–617, 2002. 1

[18] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception archi-

tecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 3

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1

[20] 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. 3

[21] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10819–10829, 2022. 1

[22] 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. 3

[23] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 3