Integrally Pre-Trained Transformer Pyramid Networks
Supplementary Materials

Yunjie Tian[1], Lingxi Xie[2], Zhaozhi Wang[1], Longhui Wei[2], Xiaopeng Zhang[2],
Jianbin Jiao[1], Yaowei Wang[3], Qi Tian[2], Qixiang Ye[1,3]
[1]UCAS [2]Huawei Inc. [3]Pengcheng Lab.
tianyunjie19@mails.ucas.ac.cn 198808xc@gmail.com wangzhaozhi22@mails.ucas.ac.cn
weilonghui1@huawei.com zxphistory@gmail.com yaoweiwang@bit.edu.cn
jiaojb@ucas.ac.cn tian.qi1@huawei.com qxye@ucas.ac.cn

1. More Details on Experiments

We provide more details on experiments. For pixel-supervised models, the pre-training and fine-tuning details are provided in Tables 7 and 8, respectively. For CLIP-supervised models, the pre-training and fine-tuning details are provided in Tables 9 and 10, respectively. During pre-training using CLIP, we add an early-stage supervision at $3/4$ of the third stage following BEiT-v2 [7].

Table 7. Hyperparameters for pre-training using image pixels as supervision on ImagetNet-1K.

Hyperparameters	base-scale	large-scale
Patch size	16	
Hidden size	512	768
Layers	3-3-24	2-2-40
FFN hidden size	2048	3072
Attention heads	8	12
Attention head size	64	
Input resolution	224×224	
Training epochs	400/1600	
Optimizer	AdamW	
Base learning rate	1.5e-4	
Weight decay	0.05	
Optimizer momentum	β_1,β_2 = 0.9,0.95	
Batch size	4096	
Learning rate schedule	cosine decay	
Warmup epochs	40	
Augmentation	RandomResizeCrop	
Absolute positional embedding	✓	
Relative positional embedding	✗	

Table 8. Hyperparameters for fine-tuning using image pixels as supervision on ImagetNet-1K.

Hyperparameters	base-scale	large-scale
Input resolution	224×224	
Training epochs	100	50
Optimizer	AdamW	
Base learning rate	5e-4	1e-3
Weight decay	0.05	
Layer decay	{.45,.5,.55}	{.5,.55,.6}
optimizer momentum	β_1,β_2 = 0.9,0.999	
Batch size	1024	
Learning rate schedule	cosine decay	
Warmup epochs	5	
Label smoothing	0.1	
Stoch. path	0.2	0.2
Dropout	✗	
Augmentation	RandAug (9,0.5)	
Mixup prob.	0.8	
Cutmix prob.	1.0	
Absolute positional embedding	✓	
Relative positional embedding	✓	

2. Handling Information Leak

When MIM-based methods are applied to pre-train iTPN, we encounter two kinds of information leak issues, namely, inter-layer and intra-layer information leak. Below, we elaborate them and describe the solutions.

The inter-layer information leak is related to the masking strategy used in this study, where the masking operation is applied across all the feature pyramid layers, *i.e.*, the hierarchical backbone and feature pyramid. As shown in Figure 5, independently and randomly masked tokens on differ-

Table 9. Hyperparameters for pre-training using CLIP models as supervision on ImagetNet-1K.

Hyperparameters	base-scale	large-scale
Patch size	16	14/16
Hidden size	512	768
Layers	3-3-24	2-2-40
FFN hidden size	2048	3072
Attention heads	8	12
Attention head size	64	
CLIP models	CLIP-B	CLIP-B/CLIP-L
Input resolution	224×224	
Lay Training epochs	300/800	300
Optimizer	AdamW	
Base learning rate	1.5e-3	
Minimal learning rate	1e-5	
Weight decay	0.05	
Optimizer momentum	β_1,β_2 = 0.9,0.98	
Batch size	2048	
Gradient clipping	3.0	
Drop path	0.1	0.2
Learning rate schedule	cosine decay	
Warmup epochs	10	
Augmentation	RandomResizeAndCrop	
Color jitter	0.4	
Absolute positional embedding	✓	
Relative positional embedding	✓	

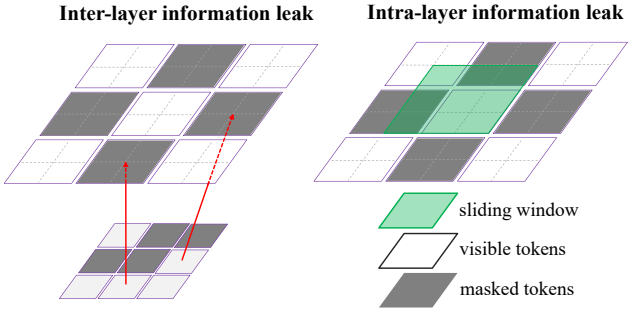

Figure 5. Information leak problems. **Left:** inter-layer information leak due from the normal tokens to masked tokens. **Right:** intra-layer information leak caused by spatial information interaction (such as convolutional operation) within the feature layer.

ent feature pyramid layers may cause the misalignment of masked tokens. Another kind of information leak is caused by the spatial overlapping of transformer pyramid layers, *i.e.*, the masked tokens in a pyramid layer can be easily reconstructed by that from adjacent layers, if those tokens are not masked. An intuitive solution is to spatially align the tokens to be masked across the feature pyramid. This simple operation achieves good performance as validated by experiments.

Table 10. Hyperparameters for fine-tuning using CLIP models as supervision on ImagetNet-1K.

Hyperparameters	base-scale	large-scale
Input resolution	224×224	
Training epochs	100	50
Optimizer	AdamW	
Base learning rate	5e-4	1e-3
Minimal learning rate	1e-6	
Layer decay	{.45,.5,.55}	{.5,.55,.6}
optimizer momentum	β_1,β_2 = 0.9,0.999	
Batch size	1024	
Learning rate schedule	cosine decay	
Warmup epochs	5	
Label smoothing	0.1	
Stoch. depth	0.2	0.3
Dropout	✗	
Gradient clipping	✗	
Weight decay	0.05	
Erasing prob.	0.25	
Augmentation	RandAug (9,0.5)	
Mixup prob.	0.8	
Cutmix prob.	1.0	
Absolute positional embedding	✓	
Relative positional embedding	✓	

The intra-layer information leak is caused by the spatial information interaction across a feature map, as convolutions or window attentions involve integrating information from masked and unmasked neighboring pixels [3]. These local operations collect information of masked tokens so that the MIM pre-training target degenerates. To solve intra-layer information, we propose to use channel-wise MLP (C-MLP) to replace all convolution and window attention operations. As C-MLP is only used to connect tokens across transformer layers and does not use any intra-layer connections, it does not involve spatial information interactions. All the required spatial information interactions are performed by the multi-head self-attention operations in deep transformer layers. This simple-yet-effective design not only solves intra-layer information leak, but also outperforms the window attention operations [6].

3. Generalizing to Plain Vision Transformers

In this part, we show that the proposed method also could be used on plain vision transformer (specifically ViT-B [2]). We up-sample the 4-th and 7-th layers from the backbone to $4\times$ and $2\times$ size features so that the hierarchical features are obtained to build the pyramid network. Other than that, all the rest modules of the architectures and settings are the

Table 11. Generalizing integral pre-training (iPT) to plain ViTs. All the results are reported with the same configurations by default. The numbers are in % for classification accuracy, box AP, and mIoU. The models are pre-trained for 400 epochs. For COCO, 1× Mask R-CNN is used and box AP is reported.

Method	epochs	ImageNet-1K	COCO	ADE20K
MAE	400	83.1	46.4	46.2
MAE	1600	83.6	48.4	48.1
iTPN	400	83.7	49.3	49.0

Table 12. Inference throughput (imgs/s) comparison with image size of 224 × 224. We test all the results using the same settings and the GPU is a V100-32G machine. The models we used report comparable throughput compared to vanilla ViT models.

Models	Base	Large
ViT	278.3	91.4
HiViT	264.1	86.0

Table 13. A model complexity comparison between FPN [5] and iTPN on object detection using Mask-RCNN [4] framework. We show that iTPN uses comparable model complexity (Params and FLOPs) and enjoys better results (1× training schedule here). We use the analysis tool provided by MMdetection [1] library to test the Params and FLOPs by using input size of 640 × 640.

Method	Pyramid	Params (M)	FLOPs (G)	AP
MAE-B	FPN [5]	115	389	48.4
iTPN-B	pyramid network	103	397	53.0
MAE-L	FPN [5]	338	756	54.0
iTPN-L	pyramid network	313	740	55.6

same to iTPN. We summarize the results on Table 11. As shown, iTPN-ViT 400e model surpasses both MAE 1600e and 400e models by a large margin, which verifies the effectiveness of integrally pre-training method.

4. Computational Efficiency

We provide the comparison on throughputs between vanilla ViT models and the proposed iTPNs (*i.e.*, HiViT models). As shown in Table 12, iTPN models report the comparable inference speeds for both base and large-scale models.

We then test the model complexity comparison in Table 13. We test the model complexity using the open analysis tool of MMdetection library [1] using the input size of 640 × 640. One can see that iTPN-base reports fewer model parameters (103M *v.s.* 115M) and comparable FLOPs (397G *v.s.* 389G) than ViT-base. For large-scale models, iTPN-large enjoys both of them: fewer model parameters (313M *v.s.* 338M) and lower FLOPs (740G *v.s.* 756G) compared to ViT-large.

References

[1] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 3

[2] 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*, 2021. 2

[3] Peng Gao, Teli Ma, Hongsheng Li, Jifeng Dai, and Yu Qiao. Convmae: Masked convolution meets masked autoencoders. *arXiv preprint arXiv:2205.03892*, 2022. 2

[4] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In *IEEE ICCV*, 2017. 3

[5] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *IEEE CVPR*, pages 936–944, 2017. 3

[6] 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 *IEEE ICCV*, pages 10012–10022, 2021. 2

[7] Zhiliang Peng, Li Dong, Hangbo Bao, Qixiang Ye, and Furu Wei. Beit v2: Masked image modeling with vector-quantized visual tokenizers. *arXiv preprint arXiv:2208.06366*, 2022. 1