Supplementary Material for DropPos: Pre-Training Vision Transformers by Reconstructing Dropped Positions

In this supplementary material, we first provide mode implementation details for reproducibility in Sec. A. Next, in Sec. B, we evaluate the performance of the position reconstruction task using pre-trained models under different settings, and we provide more evidence to support the proposed three difficulties in Sec. 1.

A Implementation details

ViT architecture. We follow the standard vanilla ViT [8] architecture used in MAE [9] as the backbone, which is a stack of Transformer blocks [17]. Following MAE [9], we use the fixed 2D sine-cosine positional embeddings during pre-training. For the downstream classification task, we use features globally averaged from the encoder output for both end-to-end fine-tuning.

Effective training epochs. Following iBOT [24], we take the effective training epochs as the metric of the training schedule, due to extra computation costs brought by the multi-crop [2] augmentation, which is a widely used technique for contrastive methods. Specifically, the effective training epochs are defined as the actual pre-training epochs multiplied with a scaling factor r. For instance, DINO [3] is trained with 2 global 224×224 crops and 10 local 96×96 crops, and thus $r = 2+(96/224)^2\times 10 \approx 4$. More details and examples can be found in [24].

A.1 ImageNet classification

For all experiments in this paper, we take ImageNet-1K [16], which contains 1.3M images for 1K categories, as the pre-trained dataset. By default, we take ViT-B/16 [8] as the backbone and it is pre-trained 200 epochs followed by 100 epochs of end-to-end fine-tuning. Implementation details can be found in the following table. Most of the configurations are borrowed from MAE [9]. The linear learning rate scaling rule is adopted: $lr = lr_{\text{base}} \times \text{batch\_size} / 256$. For supervised training from scratch, we simply follow the fine-tuning setting without another tuning. For ViT-B/16, pre-training and fine-tuning are conducted with 64 and 32 2080Ti GPUs, respectively. For ViT-L/16, pre-training and fine-tuning are conducted with 32 and 16 Tesla V100 GPUs, respectively.

config	pre-training	fine-tuning
optimizer	AdamW	AdamW
base learning rate	1.5e-4	1e-3
weight decay	0.05	0.05
momentum	β_1, β_2 = 0.9, 0.95	β_1, β_2 = 0.9, 0.999
layer-wise lr decay	1.0	0.8
batch size	4096	1024
learning rate schedule	cosine decay	cosine decay
warmup epochs	10 (ViT-B/16), 40 (ViT-L/16)	5
training epochs	200	100 (ViT-B/16), 50 (ViT-L/16)
augmentation	RandomResizedCrop	RandAug (9, 0.5) [6]
label smoothing	-	0.1
mixup [22]	-	0.8
cutmix [21]	-	1.0
drop path [11]	-	0.1

A.2 COCO object detection and segmentation

We take Mask R-CNN [10] with FPN [14] as the object detector. Following [9] and [18], to obtain pyramid feature maps for matching the requirements of FPN [14], whose feature maps are all with a stride of 16, we equally divide the backbone into 4 subsets, each consisting of a last global-window block and several local-window blocks otherwise, and then apply convolutions to get the intermediate feature maps at different scales (stride 4, 8, 16, or 32).

We perform end-to-end fine-tuning on COCO [15] for 1× schedule with 1024×1024 resolution, where 88,750 iterations of training with a batch size of 16 are performed. We simply follow the configuration of ViTDet [13], where the learning rate is 3e-4 and decays at the 78,889-th and 85,463-th iteration by a factor of 10. Experiments are conducted on 8 Tesla V100 GPUs.

Table S1: Top-1 accuracy of position reconstruction using ViT-B/16 [8] pre-trained with **different mask ratio** γ. We underline the special parameter different from our default settings. Default settings are highlighted in color. "Avg. acc" is the *averaged* top-1 accuracy over 16 different cases. We evaluate the performance using the *same* pre-trained model under different γ and γ_{pos}. Larger γ and γ_{pos} indicates a more challenging task.

(a) Pre-training with $\gamma = 0$ (Avg. acc: 59.79).

γ \ γ_{pos}	0.25	0.50	0.75	0.95	avg.
0.00	99.37	99.26	99.15	98.34	99.03
0.25	87.97	87.45	86.20	70.79	83.10
0.50	55.37	55.11	49.84	22.96	45.82
0.75	12.99	14.85	12.85	4.21	11.23

(b) Pre-training with $\gamma = 0.25$ (Avg. acc: 79.62).

γ \ γ_{pos}	0.25	0.50	0.75	0.95	avg.
0.00	99.24	99.26	99.18	98.92	99.15
0.25	98.67	98.81	98.63	97.21	98.33
0.50	93.96	93.62	90.01	62.11	84.93
0.75	50.02	47.79	35.03	11.46	36.08

(c) Pre-training with $\gamma = 0.5$ (Avg. acc: 87.27).

γ \ γ_{pos}	0.25	0.50	0.75	0.95	avg.
0.00	99.33	99.23	99.13	98.85	99.14
0.25	99.05	98.95	98.78	98.20	98.75
0.50	94.60	95.94	94.28	83.31	92.03
0.75	78.77	72.75	59.54	25.59	59.16

(d) Pre-training with $\gamma = 0.75$ (Avg. acc: 87.83).

γ \ γ_{pos}	0.25	0.50	0.75	0.95	avg.
0.00	97.19	98.69	98.20	92.30	96.60
0.25	96.78	98.26	97.66	91.05	95.93
0.50	97.26	96.82	95.66	89.12	94.72
0.75	79.94	78.10	68.73	40.24	66.75

A.3 ADE20k semantic segmentation

We take UperNet [20] as the segmentation decoder following the code of [1, 5, 18]. Fine-tuning on ADE20k [23] for 80k iterations is performed. Specifically, each iteration consists of 16 images with 512×512 resolution. The AdamW optimizer is adopted with an initial learning rate of 7e-4 and a weight decay of 0.05 with ViT-B. We apply a polynomial learning rate schedule with the first warmup of 1500 iterations following common practice [18, 5, 1]. When fine-tuning using backbones pre-trained with different methods, we search for the optimal learning rate or simply follow their official implementation for a fair comparison. Specifically, the learning rate is 1e-4 for [4, 9, 12], 4e-4 for [7, 19], respectively. All experiments are conducted on 8 Tesla V100 GPUs.

B Performance of position reconstruction

In this section, we evaluate the performance of the position reconstruction task using pre-trained models under different settings. Specifically, we vary $\gamma \in \{0, 0.25, 0.5, 0.75\}$ and $\gamma_{\text{pos}} \in \{0.25, 0.5, 0.75, 0.95\}$ when measuring the position prediction accuracy. We report performance under different evaluation settings as well as the *averaged* accuracy among 16 different cases. From Tabs. S1 to S3, we find evidence to support the three difficulties for designing an appropriate position-related pretext task introduced in Sec. 1: ***(i)*** discrepancies between pre-training and fine-tuning, ***(ii)*** failing to learn highly semantic representations by solving this simple position reconstruction task, and ***(iii)*** difficult to decide which patch positions to reconstruct precisely.

We study the effectiveness of different values of γ during pre-training in Tab. S1. Interestingly, we find evidence for *failing to learn highly semantic representations by solving this simple position reconstruction task.* As illustrated by Tab. S1a, the pre-trained model performs *extremely well* when we set $\gamma = 0$ for evaluation but fails to keep this trend when we enlarge γ. This indicates that given the strength of ViTs in modeling long-range dependencies, they have easily solved this task in a superficial way, and thus pre-training with $\gamma = 0$ becomes trivial for ViTs. To this end, an appropriate γ is necessary to increase the difficulty of the pretext task and avoid trivial solutions.

We study the effectiveness of different values of γ_{pos} during pre-training in Tab. S2, and we find evidence for *discrepancies between pre-training and fine-tuning.* As shown by Tab. S2d, the model fails to reconstruct accurate positions given some visible anchors. This is because the model has *never* been exposed to any positional embeddings (PEs) during pre-training. Therefore, providing some anchors is necessary to address discrepancies. Also, it may help the model focus on modeling *relative* relationships instead of simply reconstructing absolute positions.

Table S2: Top-1 accuracy of position reconstruction using ViT-B/16 [8] pre-trained with **different positional mask ratio** γ_{pos}. We underline the special parameter different from our default settings.

(a) Pre-training with $\underline{\gamma_{\text{pos}} = 0.25}$ (Avg. acc: 65.19).

γ \ γ_{pos}	0.25	0.50	0.75	0.95
0.00	96.52	89.38	59.29	14.58
0.25	96.62	91.27	66.00	19.17
0.50	95.58	91.10	72.07	21.69
0.75	80.56	73.72	57.89	17.66
avg.	92.32	86.37	63.81	18.28

(b) Pre-training with $\underline{\gamma_{\text{pos}} = 0.5}$ (Avg. acc: 86.70).

γ \ γ_{pos}	0.25	0.50	0.75	0.95
0.00	98.98	98.81	98.24	92.80
0.25	98.59	98.34	97.51	89.73
0.50	96.63	95.85	93.40	74.38
0.75	81.94	76.70	64.87	30.44
avg.	94.04	92.43	88.51	71.84

(c) Pre-training with $\gamma_{\text{pos}} = 0.75$ (Avg. acc: 87.83).

γ \ γ_{pos}	0.25	0.50	0.75	0.95
0.00	97.19	98.69	98.20	92.30
0.25	96.78	98.26	97.66	91.05
0.50	97.26	96.82	95.66	89.12
0.75	79.94	78.10	68.73	40.24
avg.	92.79	92.97	90.06	78.18

(d) Pre-training with $\underline{\gamma_{\text{pos}} = 1}$ (Avg. acc: 19.44).

γ \ γ_{pos}	0.25	0.50	0.75	0.95
0.00	15.57	23.24	20.41	23.59
0.25	12.51	21.10	24.30	29.41
0.50	7.76	14.18	25.56	45.01
0.75	3.34	6.00	12.53	26.45
avg.	9.80	16.13	20.70	31.12

Table S3: Top-1 accuracy of position reconstruction using ViT-B/16 [8] pre-trained with **different (i)** σ **and (ii)** τ. We underline the special parameter different from our default settings.

(a) Pre-training with $\underline{\sigma = 0}$ (Avg. acc: 88.81).

γ \ γ_{pos}	0.25	0.50	0.75	0.95
0.00	98.48	98.74	98.29	94.37
0.25	98.06	98.31	97.76	93.48
0.50	96.06	96.08	94.48	85.49
0.75	82.19	78.64	69.39	41.23

(b) Pre-training with $\underline{\sigma = 1}$ (Avg. acc: 87.13).

γ \ γ_{pos}	0.25	0.50	0.75	0.95
0.00	97.03	98.50	97.93	91.40
0.25	96.63	98.04	97.33	89.69
0.50	94.30	95.58	93.68	81.47
0.75	79.14	77.06	67.43	38.89

(c) Pre-training with $\underline{\sigma = 2}$ (Avg. acc: 69.48).

γ \ γ_{pos}	0.25	0.50	0.75	0.95
0.00	78.45	78.81	78.19	72.33
0.25	78.03	78.41	77.68	70.49
0.50	76.14	76.27	74.53	63.52
0.75	63.75	61.21	53.30	30.47

(d) Pre-training with $\sigma = 1 \to 0$ (Avg. acc: 87.83).

γ \ γ_{pos}	0.25	0.50	0.75	0.95
0.00	97.19	98.69	98.20	92.30
0.25	96.78	98.26	97.66	91.05
0.50	97.26	96.82	95.66	89.12
0.75	79.94	78.10	68.73	40.24

(e) Pre-training with $\underline{\sigma = 2 \to 0}$ (Avg. acc: 87.65).

γ \ γ_{pos}	0.25	0.50	0.75	0.95
0.00	97.63	98.61	97.93	91.30
0.25	97.22	98.19	97.46	89.96
0.50	94.99	95.85	94.06	82.50
0.75	80.33	77.92	68.42	40.11

(f) Pre-training with $\underline{\tau = \infty}$ (Avg. acc: 88.66).

γ \ γ_{pos}	0.25	0.50	0.75	0.95
0.00	98.66	98.31	97.72	91.22
0.25	98.54	98.17	97.46	91.00
0.50	96.85	96.20	94.52	84.64
0.75	83.57	79.30	70.04	42.29

(g) Pre-training with $\tau = 0.1$ (Avg. acc: 87.83).

γ \ γ_{pos}	0.25	0.50	0.75	0.95
0.00	97.19	98.69	98.20	92.30
0.25	96.78	98.26	97.66	91.05
0.50	97.26	96.82	95.66	89.12
0.75	79.94	78.10	68.73	40.24

(h) Pre-training with $\underline{\tau = 0.5}$ (Avg. acc: 87.78).

γ \ γ_{pos}	0.25	0.50	0.75	0.95
0.00	97.68	97.94	97.82	91.78
0.25	97.83	97.91	97.53	91.01
0.50	96.44	96.06	94.52	84.49
0.75	80.16	77.16	66.01	40.18

We study the effectiveness of different values of γ_{pos} during pre-training in Tab. S2, and we find evidence for *hard to decide which patch positions to reconstruct precisely.* As shown by Tabs. S3a and S3f, the model achieves higher position prediction accuracy but performs worse on downstream tasks (please refer to Tabs. 3 and 4 for downstream performances). Therefore, to prevent being overwhelmed by this particular position reconstruction task, techniques for relaxing the patch-wise classification problem become necessary, *i.e.*, position smoothing, and attentive reconstruction.

References

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

[2] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:9912–9924, 2020.

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9650–9660, 2021.

[4] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9640–9649, 2021.

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

[6] 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 Workshop (CVPRW)*, pages 702–703, 2020.

[7] Xiaoyi Dong, Jianmin Bao, Ting Zhang, Dongdong Chen, Weiming Zhang, Lu Yuan, Dong Chen, Fang Wen, and Nenghai Yu. Bootstrapped masked autoencoders for vision bert pretraining. In *European Conference on Computer Vision (ECCV)*, pages 247–264. Springer, 2022.

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

[9] 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 (CVPR)*, pages 16000–16009, 2022.

[10] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2961–2969, 2017.

[11] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *European Conference on Computer Vision (ECCV)*, 2016.

[12] Gang Li, Heliang Zheng, Daqing Liu, Chaoyue Wang, Bing Su, and Changwen Zheng. Semmae: Semantic-guided masking for learning masked autoencoders. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[13] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. In *European Conference on Computer Vision (ECCV)*, pages 280–296. Springer, 2022.

[14] 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/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2117–2125, 2017.

[15] 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 (ECCV)*, pages 740–755. Springer, 2014.

[16] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115:211–252, 2015.

[17] 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 (NeurIPS)*, 30, 2017.

[18] Haochen Wang, Kaiyou Song, Junsong Fan, Yuxi Wang, Jin Xie, and Zhaoxiang Zhang. Hard patches mining for masked image modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[19] Haoqing Wang, Yehui Tang, Yunhe Wang, Jianyuan Guo, Zhi-Hong Deng, and Kai Han. Masked image modeling with local multi-scale reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

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

[21] 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 (ICCV)*, pages 6023–6032, 2019.

[22] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. 2018.

[23] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 633–641, 2017.

[24] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. Image bert pre-training with online tokenizer. In *International Conference on Learning Representations (ICLR)*, 2022.