Appendix

Zhanzhou Feng[1] Shiliang Zhang[1,2]
[1]National Key Laboratory for Multimedia Information Processing,
School of Computer Science, Peking University
[2]Peng Cheng Laboratory
fengzz@stu.pku.edu.cn, slzhang.jdl@pku.edu.cn

1. Masking strategy

The proposed method makes the masks evlove with the training process and combines the effects of grid-wise and part-wise masking by weighted adding the corresponding probability values. We elaborate method in Sec. 3. Further, this section provides a pseudocode implementation in Algorithm 1.

Algorithm 1 PyTorch pseudocode for masking strategy.

```
# x: input patches
# r: masking ratio
# S: parts partition for patches
# α: a hyper-parameter to balance between random mask
# and semantic-guided mask

def Mask(x, r, S, α):
    B, H, W, C = x.shape
    N = H * W

    # assign part-wise probability
    part_ini = Random(N)
    p_parts = torch.gather(part_noise_ini, index=S)

    # assign grid-wise probability
    grid_ini = Random(2, 2)
    p_grid = [grid_ini[(i//W)%2, i%2] for i in range(N)]

    # aggregate probabilities with a dynamic weight
    p = (1 - α) * p_grid + α * p_parts

    # decide mask or not according to the probability
    rank = torch.argsort(p)
    patch_location = torch.argsort(rank)
    ids_mask = rank[:, :N * r]
    x_masked = torch.gather(x, index=ids_mask)

    return x_masked, patch_location
```

The random sequences $part\_ini$ and $grid\_ini$ correspond to the δ in the article, *i.e.,* a set of random values to assign the probability P. For grid-wise masks, we assign identical probability values to the patches with the same relative position in the grid to ensure that these are preserved or masked simultaneously. For part-wise masks, we set the exact probability values for the patches with the same value in S (*i.e.*, patches belonging to the same part). After that, we add the p_parts and p_grid together and mask out the top $\lfloor N \times r \rfloor$ patches with the highest probability values.

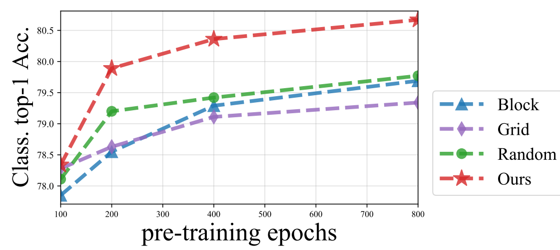

Figure 1. ImageNet-1K [1] top-1 classification performance.

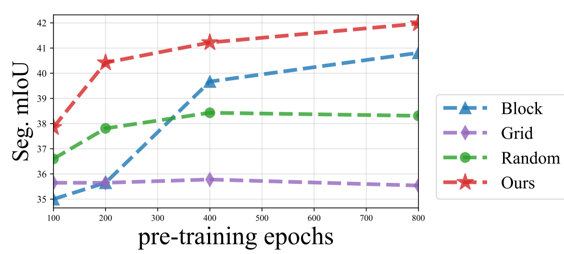

Figure 2. ADE20K [3] semantic segmentation mIoU curves.

2. Downstream performance

In this section we show the performance curves of the proposed method and three basic mask methods on downstream tasks, *i.e.* classification, segmentation and detection in 1, 2 and 3 respectively. It can be seen that the evolved method outperforms the static mask method in different pre-training epochs. Our methods can effectively converge with the limited training epochs, while with longer pre-training epochs, the method can further improve performance on various downstream tasks.

3. Compared with straightforward evolved baseline.

The partition is done with graph-cut in this paper. To validate the effectiveness of the proposed partition strategy, we replace our graph partition with random block mask-

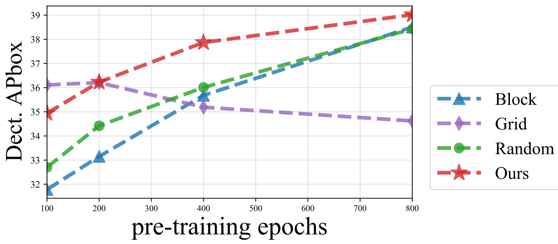

Figure 3. MSCOCO [2] detection AP-box performance.

ing as a baseline, *i.e.*, the mask evolves from grid masking to block masking. It achieves 79.18% (v.s. 79.89% of ours) ImageNet classification top-1 accuracy and 38.30% (v.s. 40.42% of ours) ADE20K [3] mIoU with 200 epochs pretraining.

References

[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 1

[2] 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. 2

[3] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017. 1, 2