What to Hide from Your Students: Attention-Guided Masked Image Modeling
Supplementary Material

Ioannis Kakogeorgiou[1], Spyros Gidaris[2], Bill Psomas[1], Yannis Avrithis[3,4], Andrei Bursuc[2], Konstantinos Karantzalos[1], and Nikos Komodakis[5,6]

[1]National Technical University of Athens [2]valeo.ai
[3]Institute of Advanced Research in Artificial Intelligence (IARAI) [4]Athena RC
[5]University of Crete [6] IACM-Forth

Table of Contents

What to Hide from Your Students: Attention-Guided Masked Image Modeling

Supplementary Material

Ioannis Kakogeorgiou[1], Spyros Gidaris[2], Bill Psomas[1], Yannis Avrithis[3,4], Andrei Bursuc[2], Konstantinos Karantzalos[1], and Nikos Komodakis[5,6]

[1]National Technical University of Athens [2]valeo.ai
[3]Institute of Advanced Research in Artificial Intelligence (IARAI) [4]Athena RC
[5]University of Crete [6] IACM-Forth

A More Experiments

We provide more benchmarks (subsection A.1), more ablations (subsection A.2), and more visualizations (subsection A.3).

A.1 More Benchmarks

How Does AttMask Affect the Patch Features? In contrast with the DINO objective that is applied only on the output [CLS] token embeddings, the MIM objective is directly applied to the output features of the patch tokens. Table A1 shows that using *global average pooling* (GAP) over patch features instead of the [CLS] token embeddings, AttMask outperforms baseline iBOT [25] by 9.0% k-NN accuracy. This indicates that AttMask leads to a more challenging MIM objective, which in turn forces the ViT to learn more discriminative patch features.

Table A1. k-NN top-1 accuracy on ImageNet-1k validation using global average pooling (GAP) over patch features *vs*. the [CLS] token embeddings. Models are pre-trained on 100% of ImageNet-1k for 100 epochs

	CLS	GAP
iBOT	71.5	49.0
iBOT + AttMask	72.5	58.0
Gain	+1.0	+9.0

Does AttMask Lead to Better Exploitation of Non-Salient Parts? We examine the performance of the models pre-trained on 100% of ImageNet-1k on a more challenging ImageNet-1k validation set. In particular, we gradually mask

Table A2. Linear probing top-1 accuracy on a more challenging *masked version* of ImageNet-1k validation set. Salient parts are gradually masked using the attention maps of the official pre-trained DINO ViT-Base model and setting the corresponding masked pixel values to zero (black). Models pre-trained on 100% of ImageNet-1k for 100 epochs

Mask Ratio (%)	0	10	30	50	70
iBOT	74.4	64.8	47.6	31.4	17.0
iBOT + AttMask	75.7	66.9	50.0	34.2	20.5
Gain	+1.3	+2.1	+2.4	+2.8	+3.5

Table A3. *Scene classification* measuring accuracy (%) using linear probing on Places205 [23]. Models pre-trained on 100% ImageNet-1k training set for 100 epochs

	iBOT	iBOT+AttMask
Places205	55.9	**56.7**

the salient parts using the attention maps of the official pre-trained DINO ViT-Base model and setting the corresponding masked pixel values to zero. Our assumption is that a more robust model should be less sensitive when salient parts of an object are missing. In Table A2, we observe that as more parts of the images are hidden, a larger gain occurs by using AttMask with iBOT. This indicates that AttMask leads to less sensitive models that exploit better the non-salient parts or even background context.

Downstream Tasks using Linear Probing. We experiment on *scene classification* on Places205 [23], measuring classification accuracy, using linear probing evaluation on models pre-trained on 100% of ImageNet-1k for 100 epochs. In Table A3, we observe that AttMask improves scores by 0.8% accuracy.

Training for More Epochs. We train iBOT with AttMask on 100% of ImageNet-1k for 300 epochs. AttMask not only accelerates the learning process and has better performance on data-limited regimes as explained in the main paper, but as we see in Table A4(a), even when trained for many epochs and with many data, it still brings an improvement of 0.4% k-NN and 0.1% linear probing over baseline iBOT [25]. Also, AttMask outperforms all other state-of-the-art frameworks on linear probing evaluation on ImageNet-1k validation set. We highlight that MST [10] employs an additional CNN decoder, while AttMask achieves improved linear probing performance with fewer learnable parameters.

We argue that the higher improvement of AttMask k-NN compared with linear probing indicates higher quality of learned embeddings, since linear prob-

Table A4. Top-1 accuracy on ImageNet validation set. (a) k-NN and linear probing using the full ImageNet training set; (b) k-NN using only $\nu \in \{1, 5, 10, 20\}$ examples per class. Pre-training on 100% ImageNet-1k for 300 epochs

METHOD	(a) FULL		(b) FEW EXAMPLES			
	k-NN	LINEAR	$\nu = 1$	5	10	20
SimCLR [5]	-	69.0				
BYOL [6]	66.6	71.4				
MoBY [21]	-	72.8				
DINO [4]	72.8	76.1				
MST [10]	**75.0**	76.9				
iBOT [25]	74.6	77.4	38.9	54.1	58.5	61.9
iBOT+AttMask (Ours)	**75.0**	**77.5**	**40.4**	**55.5**	**59.9**	**63.1**

Table A5. Top-1 accuracy on ImageNet validation set after supervised fine-tuning for 100 epochs on ImageNet-1k training set. Models pre-trained on 100% ImageNet-1k training set for 300 epochs

	iBOT	iBOT+AttMask
Fine-tuning on ImageNet	81.1	**81.3**

ing amounts to supervised classification on higher-dimensional embeddings[1] and on the same dataset that was used for self-supervised pre-training. To validate this, we experiment with a more challenging variant of k-NN where only $\nu \in \{1, 5, 10, 20\}$ examples per class of the training set are used. Table A4(b) shows that using AttMask for self-supervised pre-training and then using only simple k-NN classifier with only one example per class, achieves an accuracy improvement of 1.5% compared with the default iBOT. This highlights the superiority of AttMask in low-shot learning regimes, which are of great practical interest.

Full fine-tuning on ImageNet-1k. For iBOT and iBOT+AttMask pre-trained on ImageNet-1k for 300 epochs, we also experiment with further supervised fine-tuning on ImageNet-1k, training for 100 epochs. We report results in Table A5. AttMask improves iBOT by 0.2% (81.1% → 81.3%), providing a better network initialization for supervised finetuning.

A.2 More Ablations

MIM Loss Weight. The overall loss of iBOT [25] is a weighted sum of L_{MIM} (3), with weight λ, and L_{G} (4) + L_{LC} (A1) (DINO), with weight 1. Table

[1] We remind that, following the evaluation setups of DINO [4] for ViT-S, for linear probing we use the concatenated features from the last 4 layers of ViT while for k-NN the feature from only the last layer. So, linear probing uses 4 times higher-dimensional features

Table A6. k-NN top-1 accuracy on ImageNet-1k validation *vs.* MIM Loss Weight λ, while the weight of DINO loss is fixed to 1. Pre-training on 20% of ImageNet-1k for 100 epochs

MIM Loss Weight λ	0.0	0.5	1.0	2.0
iBOT	43.4	46.5	46.7	41.9
iBOT+AttMask	43.5	47.3	**49.7**	48.3
Gain	+0.1	+0.8	+3.0	+6.4

A6 shows that AttMask is superior to the default block-wise random masking of iBOT in all cases, while the default $\lambda = 1$ works best for both and yields the greatest gain of 3% k-NN accuracy for AttMask. In particular, increasing the weight of the MIM loss leads to a larger gain in k-NN accuracy. This shows that AttMask boosts the MIM task.

Table A7. *AttMask-High* vs. *random masking strategies*: k-NN top-1 accuracy on ImageNet-1k validation for iBOT pre-training on 20% of ImageNet-1k for different mask ratio r. †: default iBOT masking strategy from BEiT [1]

Mask Ratio r (%)	10-30	10-50	10-70	30
Random Block-Wise	46.5	46.7†	47.1	46.9
Random	47.6	47.8	47.8	48.2
AttMask-High	49.5	**49.7**	48.5	49.1

Masking strategy and mask ratio. We ablate both the masking strategy (random block-wise, random or AttMask-High) and the mask ratio r in Table A7. AttMask-High with 10-50 mask ratio gives the best results.

A.3 More Visualizations

Visualization of Attention Maps. In Figure A1, we utilized the pre-trained models on 20% of ImageNet and observe that, when training iBOT with the default block-wise random masking strategy, there is at least one head (in blue) that attends the background to a great extent. By contrast, with our AttMask, all heads mostly attend salient objects or object parts. It appears that by focusing on reconstructing highly-attended masked tokens, the network learns to focus more on foreground objects.

Visualization of Masking Examples. We illustrate the effect of mask ratio r (%) to various masking strategies in Figure A2 and Figure A3. While random Block-Wise and Random masking fail to consistently mask informative parts of an image, AttMask-High and AttMask-Hint make use of attention to hide salient and all but very salient parts respectively. This gives rise to a more challenging MIM task.

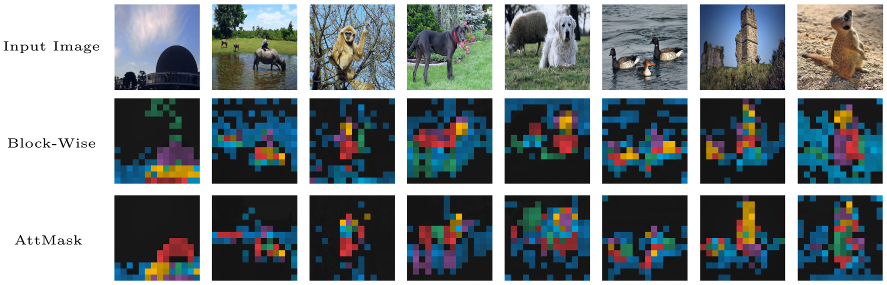

Fig. A1. Multi-head attention maps from the last layer, training iBOT with the default block-wise strategy from BEiT [1] and with our AttMask. From the attention matrix (5) of each head, we extract the attention map of the [CLS] token and display in different color per head the patch tokens that are included in the top 60% of the attention mass

B Experimental Setup

We provide more details on the experimental setup, including multi-crop, training details and evaluation details.

Multi-Crop. Following [4,25], we apply the *multi-crop* strategy [3] to generate a set of m low-resolution *local crops*, which cover only small parts of the image, tokenized as $Z^c_1, \dots, Z^c_m$. Similar to L_{G} (4), the loss is applied globally on the [CLS] tokens, in particular between the student output for a local crop Z^c_j and the teacher output for a global view Z^v, both of which are non-masked:

$$L_{\text{LC}} = -\sum_{v \in V} \sum_{j=1}^{m} f_{\theta'}(Z^v)^{[\text{CLS}]} \log(f_\theta(Z^c_j)^{[\text{CLS}]}). \tag{A1}$$

The overall loss is a weighted sum of L_{MIM} (3), L_{G} (4) and L_{LC} (A1).

Training Details. For our *analysis* and *ablation* (subsection 4.2, subsection 4.4 and subsection A.2), we pre-train models on 20% of ImageNet-1k for 100 epochs. For both iBOT and DINO we use AdamW [14] as optimizer. Unless otherwise stated, we use the ViT-S/16 architecture and a batch size of 240. We warm-up learning rate η for 10 epochs following the linear scaling rule $\eta = 5 \times 10^{-4} \times \texttt{bs}/256$ where `bs` is the batch size and then decay using a cosine schedule. We also use a cosine schedule from 0.04 to 0.4 for weight decay. We set teacher momentum to 0.99 and student temperature to 0.1. We use a linear warm-up for teacher temperature from 0.04 to 0.07 for the first 30 epochs following DINO.

All methods in subsection 4.2, subsection 4.4 and subsection A.2 use the multi-crop scheme with two 224^2 global crops and six 96^2 local crops that approximately scale the training time by a factor of $\gamma = 2 + 6 \times (96/224)^2 = 3.10$. We use color jittering, Gaussian blur and solarization as data augmentations. Local crops scales are sampled from $(0.05, s)$ and global crop scales from $(s, 1)$. We set s to 0.4 for DINO and 0.25 for iBOT. We set the dimensionality of the head output to 65536 for DINO, while for iBOT, we use a shared projection head for [CLS] and patch tokens, of dimensionality 8192. We do not perform weight normalization on the last layer of the MLP heads.

For our *benchmark* (subsection 4.3 and subsection A.1), we pre-train models on 100% of ImageNet-1k for 100 and 300 epochs. For the 100-epoch experiments, the setup is the same as on 20% of ImageNet-1k except for increasing the teacher momentum to 0.996 and the number of local crops to ten. The scaling factor of the training time in this case is $\gamma = 2 + 10 \times (96/224)^2 = 3.84$. For the 300 epochs experiments, we increase the batch size to 800 and set s to 0.32, similar to the iBOT default scale.

Evaluation Details. For the ImageNet-1k evaluation, we use k-NN and linear probing as in DINO [4] and iBOT [25]. We evaluate on ImageNet-1k validation set. For k-NN, we use the [CLS] feature from the last ViT layer and set k to 20. For linear probing, we train a linear classifier using SGD with a batch size of 1024 for 100 epochs. We set learning rate to 0.003 and do not apply weight decay. We apply random resized crops and horizontal flips as data augmentations and keep the central crop. Following DINO [4] and iBOT [25], we use the concatenation of the [CLS] features from the last four layers as input to the linear classifier.

For the evaluation of downstream tasks *with finetuning*, we train models on CIFAR10, CIFAR100 [9] for 500 epochs and on Oxford Flowers [15] for 1000 epochs. We set learning rate to 7.5×10^{-6}, weight decay to 0.05 and use a batch size of 900.

On COCO [11], we evaluate the performance of object detection and instance segmentation downstream tasks. We consider the COCO 2017 set, which contains 118K training images, 5k validation and 20 test-dev. We consider the Cascade Mask R-CNN [2, 7] as task layer and follow the setup from [12]. We use the hyper-parameter configuration from [25]: multi-scale training (resizing image with shorter size between 480 and 800, with the longer side no larger than 1333). We use AdamW [14] with initial learning rate 10^{-4}, the 1× schedule (12 epochs with the learning rate decayed by 10× at epochs 9 and 11) and weight decay 0.05. Unlike [25], where training is on 8 GPUs with 4 images per GPU, we use 2 images per GPU due to hardware limitations. For a fair and direct comparison, we fine-tune iBOT baseline with the same configuration.

We evaluate on ADE20K [24] for the semantic segmentation downstream task. It consists of 25k images in 150 classes, with 20k for training, 2k for validation and 3k for testing. We rely on UperNet [20] as task layer and fine-tune the entire network following the setup from [12]: 160k iterations with 512×512 images. We do not perform multi-scale training and testing. We adopt the same hyper-parameters as in [25]. We use the AdamW [14] optimizer with an initial learning rate of 7×10^{-4} with poly-scheduling, layer decay rate 0.65 and weight decay 0.05. We train on 8 GPUs with 2 images per GPU.

For the evaluation of downstream tasks *without finetuning*, we follow the protocol of DINO on $\mathcal{R}$Oxford, $\mathcal{R}$Paris [18] and DAVIS 2017 [17]. On Caltech-UCSD Birds (CUB200) [19], Cars (CARS196) [8], Stanford Online Products (SOP) [16] and In-Shop Clothing Retrieval (In-Shop) [13], we extract features from test set images and directly apply nearest neighbor search to measure

Recall@k [16]. On Places205 [23], we train a 205-way linear classifier on pre-cached features, using only horizontal flip as augmentation. Training is with SGD for 50 epochs using an initial learning rate of 0.01 that is decreased to 0 with cosine schedule, a batch-size of 1024, and no weight decay.

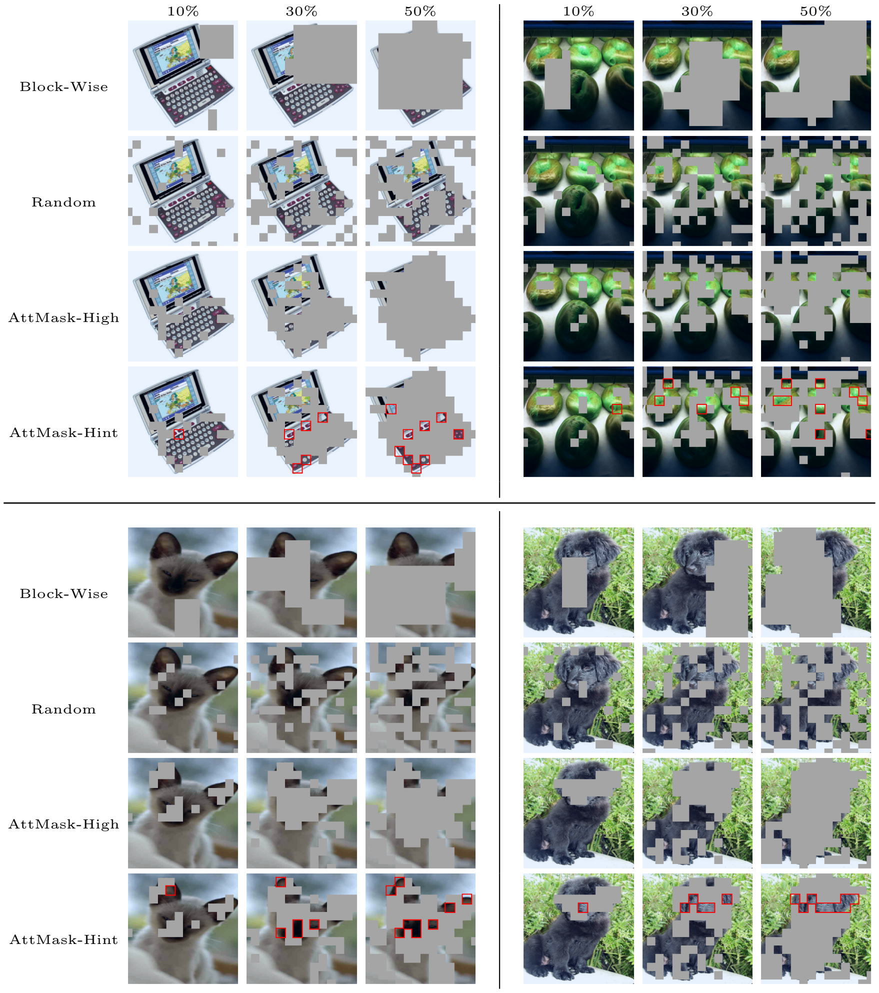

Fig. A2. Illustration of different masking strategies *vs*. mask ratio r (%) (part 1). We compare random Block-Wise masking (BEiT [1]) with Random masking (SimMIM [22]), AttMask-High and AttMask-Hint. Our AttMask-High uses the attention map arising in the encoder to hide patches, while AttMask-Hint reveals very salient patches to leave hints about the identity of the masked object

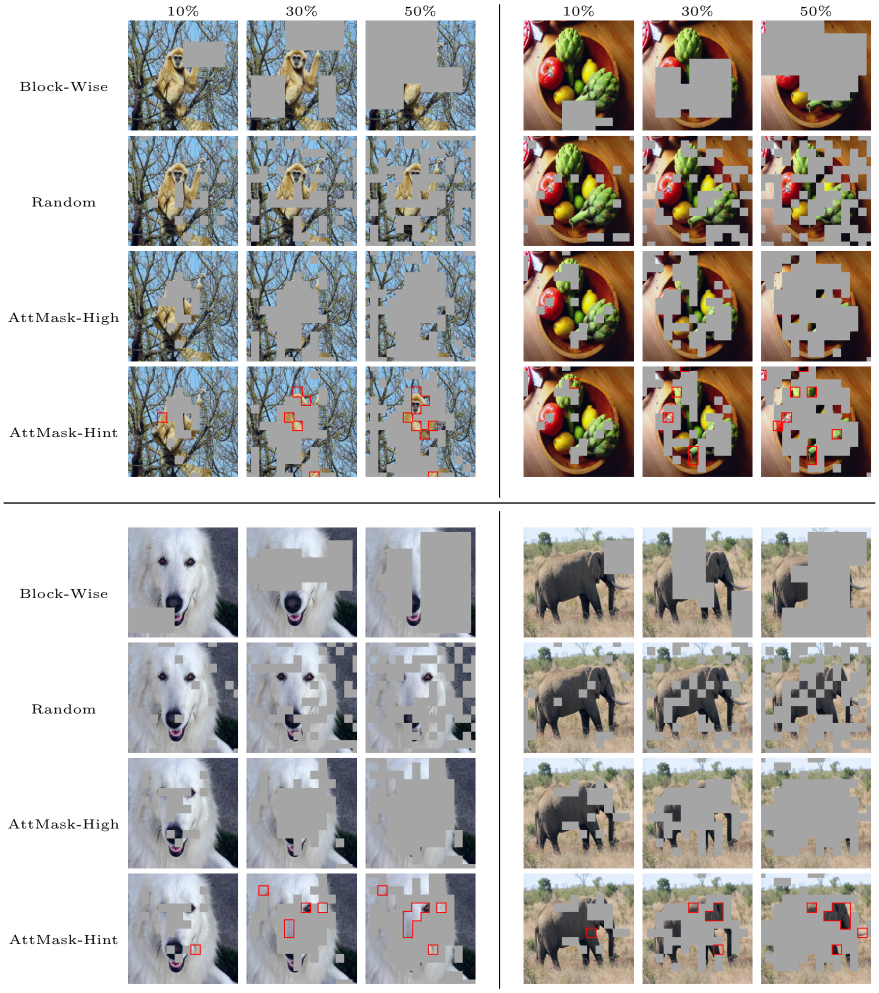

Fig. A3. Illustration of different masking strategies *vs*. mask ratio r (%) (part 2). We compare random Block-Wise masking (BEiT [1]) with Random masking (SimMIM [22]), AttMask-High and AttMask-Hint. Our AttMask-High uses the attention map arising in the encoder to hide patches, while AttMask-Hint reveals very salient patches to leave hints about the identity of the masked object

References

1. Bao, H., Dong, L., Piao, S., Wei, F.: BEit: BERT pre-training of image transformers. In: International Conference on Learning Representations (2022)
2. Cai, Z., Vasconcelos, N.: Cascade r-cnn: high quality object detection and instance segmentation. IEEE transactions on pattern analysis and machine intelligence (2019)
3. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. Advances in Neural Information Processing Systems **33**, 9912–9924 (2020)
4. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9650–9660 (2021)
5. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International Conference on Machine Learning. pp. 1597–1607. PMLR (2020)
6. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al.: Bootstrap your own latent-a new approach to self-supervised learning. Advances in Neural Information Processing Systems **33**, 21271–21284 (2020)
7. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision (2017)
8. Krause, J., Stark, M., Deng, J., Li, F.F.: 3d object representations for fine-grained categorization. ICCVW (2013)
9. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
10. Li, Z., Chen, Z., Yang, F., Li, W., Zhu, Y., Zhao, C., Deng, R., Wu, L., Zhao, R., Tang, M., et al.: Mst: Masked self-supervised transformer for visual representation. Advances in Neural Information Processing Systems **34** (2021)
11. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European Conference on Computer Vision. pp. 740–755. Springer (2014)
12. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2021)
13. Liu, Z., Luo, P., Qiu, S., Wang, X., Tang, X.: Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
14. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: International Conference on Learning Representations (2018)
15. Nilsback, M.E., Zisserman, A.: Automated flower classification over a large number of classes. In: Indian Conference on Computer Vision, Graphics and Image Processing (Dec 2008)
16. Oh Song, H., Xiang, Y., Jegelka, S., Savarese, S.: Deep metric learning via lifted structured feature embedding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
17. Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)

18. Radenović, F., Iscen, A., Tolias, G., Avrithis, Y., Chum, O.: Revisiting oxford and paris: Large-scale image retrieval benchmarking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5706–5715 (2018)
19. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The Caltech-UCSD Birds-200-2011 Dataset. Tech. Rep. CNS-TR-2011-001, California Institute of Technology (2011)
20. Xiao, T., Liu, Y., Zhou, B., Jiang, Y., Sun, J.: Unified perceptual parsing for scene understanding. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)
21. Xie, Z., Lin, Y., Yao, Z., Zhang, Z., Dai, Q., Cao, Y., Hu, H.: Self-supervised learning with swin transformers. arXiv preprint arXiv:2105.04553 (2021)
22. Xie, Z., Zhang, Z., Cao, Y., Lin, Y., Bao, J., Yao, Z., Dai, Q., Hu, H.: Simmim: A simple framework for masked image modeling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9653–9663 (2022)
23. Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., Oliva, A.: Learning deep features for scene recognition using places database. In: Advances in Neural Information Processing Systems (2014)
24. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ade20k dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 633–641 (2017)
25. Zhou, J., Wei, C., Wang, H., Shen, W., Xie, C., Yuille, A., Kong, T.: ibot: Image bert pre-training with online tokenizer. In: International Conference on Learning Representations (2022)