Supplementary Material
SAM Helps SSL: Mask-guided Attention Bias for Self-supervised Learning

Kensuke Taguchi*[1]
kensuke.taguchi.xm@kyocera.jp
Takehiko Kawai*[1]
takehiko.kawai.yb@kyocera.jp
Wataru Imaeda[1]
wataru.imaeda.xm@kyocera.jp
Hironobu Fujiyoshi[2]
fujiyoshi@isc.chubu.ac.jp

[1] KYOCERA Corporation, Japan
[2] Chubu University, Japan
*Equal contribution

A Training Details

A.1 DINO

Loss function. As the baseline CL method, we selected DINO [2], a self-distillation method that feeds multiple views of an image to two encoders. DINO reinterprets the embeddings as the logits of a clustering model, defined by a set of learned "prototypes." DINO's loss function is defined as

$$L^{cl}(\mathbf{z},\mathbf{z}') = H(P(\mathbf{z}),P'(\mathbf{z}')), \tag{1}$$

where $H(a,b) = -a\log b$. Given an input image x, the auxiliary head's output probability distributions over k dimensions are denoted as P and P'. They are obtained by softmax normalization as

$$P(\mathbf{z})^{(i)} = \frac{exp(\mathbf{z}^{(i)}/\tau)}{\sum_{k=1}^{K} exp(\mathbf{z}^{(k)}/\tau)}, \tag{2}$$

$$P'(\mathbf{z}')^{(i)} = \frac{exp((\mathbf{z}'^{(i)}-o)/\tau')}{\sum_{k=1}^{K} exp(\mathbf{z}'^{(k)}-o)/\tau'}, \tag{3}$$

where $\tau, \tau' > 0$ are the temperature parameters that control the sharpness of the output distribution and o is a centering parameter that prevents one dimension from dominating, thereby preventing collapse.

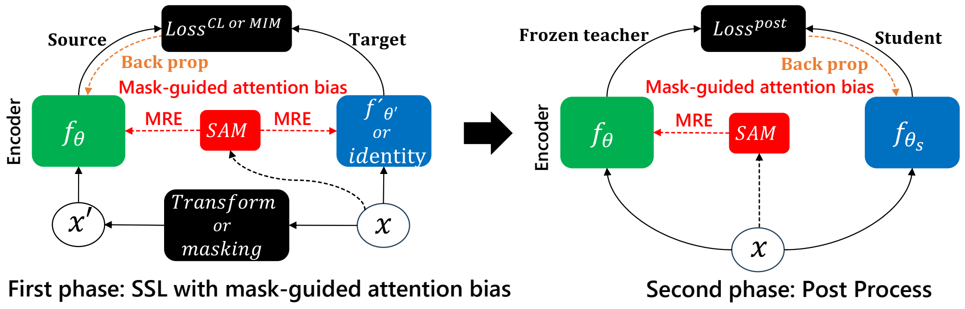

Figure 1: Overview of our method. f_θ be a ViT encoder with mask-guided attention bias and f_{θ_s} be a post-processed ViT student model encoder that does not take mask token vectors.

As post-processing for DINO with mask-guided attention bias in the second phase shown in Fig 1, we used the loss function proposed by RoB [4]. This is defined as

$$L^{kd}_{cl}(\mathbf{z}_s, \mathbf{z}) = H(P(\mathbf{z}_s), P(\mathbf{z})). \tag{4}$$

Note that Eq. (4) does not include a centering parameter as a collapse-preventing regularization term. The implementation details of post-processing in the second phase use the same settings as in the first phase.

Implementation details. Table 1 shows the implementation details of DINO alone and DINO with mask-guided attention bias in the first phase shown in Fig 1. We basically follow Caron *et al.* [2], except that we changed the base learning rate to 2.5e-3, five times the original rate for ImageNet100. We use the linear lr scaling rule [5]: $lr = base\_lr \times \frac{batchsize}{256}$. For training stability, we froze the mask-guided attention bias parameters during the first 100 epochs as a warm-up. Tables 3 and 5 show the evaluation settings for linear probing and full-data fine-tuning, which are same as in Caron *et al.* [2]. For the settings of 1% and 10% labeled data fine-tuned evaluations as few-shot image recognition, we basically follow Semi-ViT in the context of semi-supervised learning [1].

A.2 MAE

Loss function. For the baseline MIM method, we selected MAE [6], which features computational efficiency and state-of-the-art performance in a wide range of downstream tasks. The MAE loss function is defined as

$$L^{mim}(\mathbf{z}^{pred}, \mathbf{x}^{masked}) = MSE(\mathbf{z}^{pred}, \mathbf{x}^{masked}), \tag{5}$$

where $\mathbf{z}^{pred}$ is output from an auxiliary decoder.

Implementation details. Table 2 shows the implementation details for MAE alone and MAE with mask-guided attention bias in the first phase shown in Fig 1. We used the same settings as in He *et al.* [6]. Unlike DINO with mask-guided attention bias, we did not initially freeze the mask-guided attention bias parameters as warm-up. Note that we set the mask size to 2×2 token size in the pretraining with ViT/8 [7]. Tables 4 and 6 show the evaluation settings for linear probing and full-data fine-tuning, which are the same as in He *et al.* [6]. For the settings of 1% and 10% labeled data fine-tuned evaluations, we used the same settings as Semi-ViT [1].

Setting	Value
Optimizer	AdamW [9]
Batch size	512
Base learning rate	2.5e-3
Warmup epochs	10
Learning rate schedule	cosine schedule [8]
Weight decay	0.04 to 0.4
Teacher temp	0.04 to 0.07
Momentum teacher	0.996 to 1
Augmentation	RandomResizedCrop, Color Jittering, Gaussian Blur, Solarization, RandomHorizontalFlip

Table 1: DINO Pretraining settings

Setting	Value
Optimizer	AdamW
Base learning rate	1.5e-4
Optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.95$
Batch size	4096
Learning rate schedule	cosine schedule
Warmup epochs	40
Augmentation	RandomResizedCrop
Masking ratio	0.75
Masking strategy	Random

Table 2: MAE Pretraining settings

Setting	Value
Optimizer	SGD
Base learning rate	1e-3
Optimizer momentum	0.9
Batch size	256
Learning rate schedule	cosine schedule
Training epochs	100
Augmentation	RandomResizedCrop, RandomHorizontalFlip

Table 3: DINO Linear probing settings

Setting	Value
Optimizer	LARS [11]
Base learning rate	0.1
Weight decay	0
Optimizer momentum	0.9
Batch size	16384
Learning rate schedule	cosine decay
Warmup epochs	10
Training epochs	90
Augmentation	RandomResizedCrop

Table 4: MAE Linear probing settings

Setting	Value
Optimizer	AdamW
Base learning rate	1e-4
Weight decay	0.05
Optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.999$
Layer-wise learning rate decay	0.65
Batch size	256
Learning rate schedule	cosine decay
Warmup epochs	5
Training epochs	100
Augmentation	RandAug (9, 0.5) [3]
Label smoothing [10]	0.1
Mixup [13]	0.8
Cutmix [12]	1.0
Drop path	0.1
Random erasing [14]	0.25

Table 5: DINO Full-data fine-tuning settings

Setting	Value
Optimizer	AdamW
Base learning rate	1e-3
Weight decay	0.05
Optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.95$
Layer-wise learning rate decay	0.75
Batch size	1024
Learning rate schedule	cosine decay
Warmup epochs	5
Training epochs	100
Augmentation	RandAug (9, 0.5)
Label smoothing	0.1
Mixup	0.8
Cutmix	1.0
Drop path	0.1

Table 6: MAE Full-data fine-tuning settings

References

[1] Zhaowei Cai, Avinash Ravichandran, Paolo Favaro, Manchen Wang, Davide Modolo, Rahul Bhotika, Zhuowen Tu, and Stefano Soatto. Semi-supervised vision transformers at scale. In *NeurIPS*, 2022.

[2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021.

[3] Ekin Dogus Cubuk, Barret Zoph, Jon Shlens, and Quoc Le. Randaugment: Practical automated data augmentation with a reduced search space. In *NeurIPS*, 2020.

[4] Quentin Duval, Ishan Misra, and Nicolas Ballas. A simple recipe for competitive low-compute self supervised vision models. *arXiv:2301.09451*, 2023.

[5] 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:1706.02677*, 2018.

[6] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.

[7] Ronghang Hu, Shoubhik Debnath, Saining Xie, and Xinlei Chen. Exploring long-sequence masked autoencoders. *arXiv:2210.07224*, 2022.

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

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

[10] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

[11] Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv:1708.03888*, 2017.

[12] 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 *ICCV*, 2019.

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

[14] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *AAAI*, 2020.