A. Implementation Details

A.1. Pre-training

For pre-training, we mainly follow the setting of MAE [23]. Detailed hyper-parameters for pre-training are listed in Tab. 4. Below we describe the main differences between SiameseIM and MAE.

Augmentation. We use the strong augmentations from MoCo-v3 [13], including random resized cropping, horizontal flipping, color jittering, grayscale conversion, Gaussian blurring and solarization. For masking strategy, we follow BEiT [4] to use blockwise masking with a masking ratio of 60%.

Architecture. We use the standard ViT-B/16 [19] as the backbone for both online and target branches. We stack 2 and 4 Transformer encoder blocks with BatchNorm [31] as the projector and the decoder, respectively. Both the projector and decoder have 768 embedding dimension and 12 heads for each block. The EMA coefficient for target momentum encoder is initialized as 0.995 and is applied with a cosine schedule from 0.995 to 1.0. Before calculating loss, we follow MAE [23] to apply LayerNorm without affine parameters to target, and no normalization to prediction.

Table 4. Hyper-parameters for pre-training.

Hyper-parameters	Value
Layers	12
Hidden size	768
FFN inner hidden size	3072
Attention heads	12
Patch size	16×16
Data augment	RandomResizedCrop RandomHorizontalFlip ColorJitter RandomGrayscale GaussianBlur Solarize
Mask strategy	Blockwise mask
Mask ratio	60%
Input resolution	224×224
Training epochs	1600
Batch size	4096
Adam β	(0.9, 0.95)
Peak learning rate	1.0×10^{-3}
Learning rate schedule	cosine
Warmup epochs	40
Weight decay	0.05
EMA coeff	0.995
EMA schedule	cosine

A.2. Finetuning with 100% Data

We follow the finetuning setting of MAE [23] except that we search for the optimal learning rate. Other hyper-parameters are listed in Tab. 5.

Table 5. Hyper-parameters for ImageNet finetuning.

Hyper-parameters	Value
Layers	12
Hidden size	768
FFN inner hidden size	3072
Attention heads	12
Patch size	16×16
Layer-wise learning rate decay	0.65
Erasing prob.	0.25
Rand augment	9/0.5
Mixup prob.	0.8
Cutmix prob.	1.0
Input resolution	224×224
Finetuning epochs	100
Batch size	1024
Adam β	(0.9, 0.999)
Peak learning rate	1.0×10^{-3}
Learning rate schedule	cosine
Warmup epochs	5
Weight decay	0.05
Label smoothing	0.1
Stochastic depth	0.1

A.3. Linear Probing

We follow the linear probing setting of MAE [23] while always searching for the optimal learning rate. Specifically, an extra BatchNorm layer without affine transformation is added before the final linear classifier. Other hyper-parameters are listed in Tab. 6.

A.4. Finetuning with 1% Data

For few-shot evaluation, we follow the practice in [1]. Specifically, we freeze the backbone and extract representations for each image. Then the cyanure package [38] is used to apply L_2-regularized logistic regression on the representations. Note that for MAE, we report partial finetuning result because it is better than just training a linear classifier [1].

A.5. COCO Detection

We follow [34] to evaluate on COCO [36]. We adjust the learning rate schedule so as to drop the learning rate once the performance saturates. Hyper-parameters are listed in Tab. 7.

Table 6. Hyper-parameters for ImageNet linear probing.

Hyper-parameters	Value
Layers	12
Hidden size	768
FFN inner hidden size	3072
Attention heads	12
Patch size	16×16
Data augment	RandomResizedCrop RandomHorizontalFlip
Input resolution	224×224
Training epochs	90
Batch size	16384
Optimizer	LARS
Peak learning rate	3.2
Learning rate schedule	cosine
Warmup epochs	10
Weight decay	0.0

Table 7. Hyper-parameters for COCO detection.

Hyper-parameters	Value
Layers	12
Hidden size	768
FFN inner hidden size	3072
Attention heads	12
Patch size	16×16
Layer-wise learning rate decay	0.7
Data augment	large scale jittor
Input resolution	1024×1024
Finetuning epochs	100
Batch size	64
Adam β	(0.9, 0.999)
Peak learning rate	1.0×10^{-4}
Learning rate schedule	step
Warmup length	250 iters
Weight decay	0.1
Stochastic depth	0.1
Relative positional embeddings	✓

Table 8. Hyper-parameters for ADE20k semantic segmentatioin.

Hyper-parameters	Value
Layers	12
Hidden size	768
FFN inner hidden size	3072
Attention heads	12
Patch size	16×16
Layer-wise learning rate decay	0.65
Data augment	RandomCrop RandomFlip PhotoMetricDistortion
Input resolution	512×512
Finetuning length	160k iters
Batch size	16
Adam β	(0.9, 0.999)
Peak learning rate	1.0×10^{-4}
Learning rate schedule	linear
Warmup length	1500 iters
Weight decay	0.05
Stochastic depth	0.1
Relative positional embeddings	✓

Table 9. Hyper-parameters for LVIS detection.

Hyper-parameters	Value
Layers	12
Hidden size	768
FFN inner hidden size	3072
Attention heads	12
Patch size	16×16
Layer-wise learning rate decay	0.7
Data augment	large scale jittor
Input resolution	1024×1024
Finetuning epochs	100
Batch size	64
Adam β	(0.9, 0.999)
Peak learning rate	2.0×10^{-4}
Learning rate schedule	step
Warmup length	250 iters
Weight decay	0.1
Stochastic depth	0.1
Relative positional embeddings	✓

A.6. Semantic Segmentation

We follow [4] to use UperNet [54] as the segmentation network. We use the open-source code from mmsegmentation [15] and only change pretrained backbone. Hyper-parameters are listed in Tab. 8.

A.7. LVIS Detection

We follow [34] to evaluate on LVIS [22]. We adjust the learning rate schedule so as to drop the learning rate once the performance saturates. Hyper-parameters are listed in Tab. 9.

A.8. Robustness benchmarks

We finetune the model on original ImageNet using the setting in Tab. 5, and test it on different validation sets [26–28, 47] without further finetuning.

B. Attribution of Assets

ImageNet is subject to the ImageNet terms of access [14]. COCO 2017 is publicly available under the Creative Commons Attribution 4.0 License. As far as we know, they do not contain any personally identifiable information or offensive content.