Hard Patches Mining for Masked Image Modeling

Haochen Wang[1,3] Kaiyou Song[2] Junsong Fan[1,4] Yuxi Wang[1,4] Jin Xie[2] Zhaoxiang Zhang[1,3,4]

[1]Center for Research on Intelligent Perception and Computing,
National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences
[2]Megvii Technology [3]University of Chinese Academy of Sciences
[4]Centre for Artificial Intelligence and Robotics,
Hong Kong Institute of Science & Innovation, Chinese Academy of Science

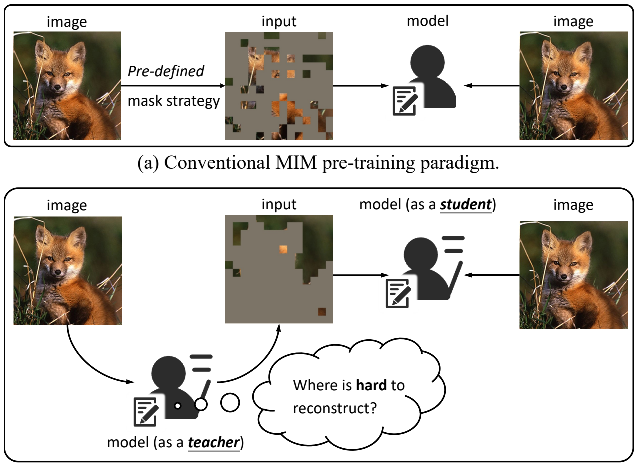
{wanghaochen2022, junsong.fan, zhaoxiang.zhang}@ia.ac.cn
{songkaiyou, xiejin}@megvii.com yuxiwang93@gmail.com

Abstract

Masked image modeling (MIM) has attracted much research attention due to its promising potential for learning scalable visual representations. In typical approaches, models usually focus on predicting specific contents of masked patches, and their performances are highly related to pre-defined mask strategies. Intuitively, this procedure can be considered as training a student (the model) on solving given problems (predict masked patches). However, we argue that the model should not only focus on solving given problems, but also ***stand in the shoes of a teacher*** *to produce a more challenging problem by itself. To this end, we propose Hard Patches Mining (HPM), a brand-new framework for MIM pre-training. We observe that the reconstruction loss can naturally be the metric of the difficulty of the pre-training task. Therefore, we introduce an auxiliary loss predictor, predicting patch-wise losses first and deciding where to mask next. It adopts a relative relationship learning strategy to prevent overfitting to exact reconstruction loss values. Experiments under various settings demonstrate the effectiveness of HPM in constructing masked images. Furthermore, we empirically find that solely introducing the loss prediction objective leads to powerful representations, verifying the efficacy of the ability to be aware of where is hard to reconstruct.*[1]

1. Introduction

Self-supervised learning [6, 8, 9, 18, 20], with the goal of learning scalable feature representations from large-scale datasets without any annotations, has been a research hotspot in computer vision (CV). Inspired by masked language modeling (MLM) [4, 11, 44, 45] in natural language processing (NLP), where the model is urged to predict masked words within a sentence, masked image modeling (MIM), the counterpart in CV, has attracted numerous interests of researchers [3, 13, 19, 26, 42, 61, 66, 69].



(a) Conventional MIM pre-training paradigm.

(b) Our proposed MIM pre-training paradigm.

Figure 1. Comparison between conventional MIM pre-training paradigm and our proposed HPM. **(a)** Conventional approaches can be interpreted as training a ***student***, where the model is only equipped with the ability to solve a given problem under some pre-defined mask strategies. **(b)** Our proposed HPM pre-training paradigm makes the model to be both a ***teacher*** and a ***student***, with the extra ability to *produce a challenging pretext task*.

Fig. 1a illustrates the paradigm of conventional approaches for MIM pre-training [3, 19, 67]. In these typical solutions, models usually focus on predicting specific contents of masked patches. Intuitively, this procedure can be considered as training a student (*i.e.*, the model) on solving given problems (*i.e.*, predict masked patches). To alleviate the spatial redundancy in CV [19] and produce

[1] Code: https://github.com/Haochen-Wang409/HPM

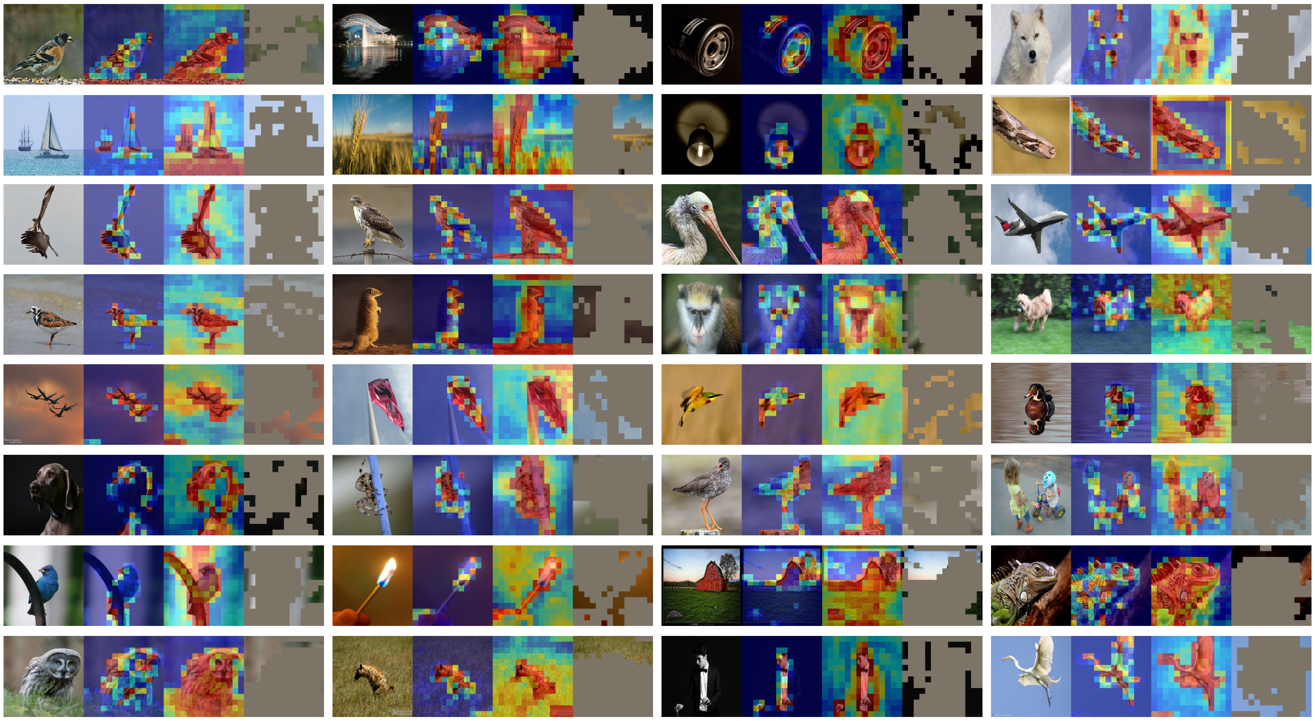


Figure 2. Visual comparison between **reconstruction loss** and **discriminativeness** on **ImageNet** *validation* set. We load the pre-trained ViT-B/16 [14] provided by MAE [19]. For each tuple, we show the **(a)** *input image*, **(b)** *patch-wise reconstruction loss* averaged over 10 different masks, **(c)** *predicted loss*, and **(d)** *masked images* generated by the predicted loss (*i.e.*, patches with top 75% predicted loss are masked). Red means higher loss while blue indicates the opposite. *Discriminative parts tend to be hard to reconstruct.*

a challenging pretext task, mask strategies become critical, which are usually generated under pre-defined manners, *e.g.*, random masking [19], block-wise masking [3], and uniform masking [29]. However, we argue that a difficult pretext task is not all we need, and not only learning to solve the MIM problem is important, but also *learning to produce challenging tasks* is crucial. In other words, as shown in Fig. 1b, by learning to create challenging problems and solving them *simultaneously*, the model can stand in the shoes of both a ***student*** and a ***teacher***, being forced to hold a more comprehensive understanding of the image contents, and thus leading itself by generating a more desirable task.

To this end, we propose *Hard Patches Mining (HPM)*, a new training paradigm for MIM. Specifically, given an input image, instead of generating a binary mask under a manually-designed criterion, we first let the model be a teacher to produce a demanding mask, and then train the model to predict masked patches as a student just like conventional methods. Through this way, the model is urged to learn where it is worth being masked, and how to solve the problem at the same time. Then, the question becomes how to design the auxiliary task, to make the model aware of where the hard patches are.

Intuitively, we observe that the reconstruction loss can be naturally a measure of the difficulty of the MIM task, which can be verified by the first two elements of each tuple in Fig. 2, where the backbone[2] pre-trained by MAE [19] with 1600 epochs is used for visualization. As expected, we find that those discriminative parts of an image (*e.g.*, object) are usually hard to reconstruct, resulting in larger losses. Therefore, by simply urging the model to *predict reconstruction loss* for each patch, and then masking those patches with higher predicted losses, we can obtain a more formidable MIM task. To achieve this, we introduce an auxiliary loss predictor, predicting patch-wise losses first and deciding where to mask next based on its outputs. To prevent it from being overwhelmed by the exact values of reconstruction losses and make it concentrate on the *relative relationship among patches*, we design a novel relative loss based on binary cross-entropy as the objective. We further evaluate the effectiveness of the loss predictor using a ViT-B under 200 epochs pre-training in Fig. 2. As the last two elements for each tuple in Fig. 2 suggest, patches with larger *predicted* losses tend to be discriminative, and thus masking these patches brings a challenging situation, where objects are almost masked. Meanwhile, considering the training evolution, we come up with an easy-to-hard mask generation strategy, providing some reasonable hints at the early stages.

Empirically, we observe significant and consistent improvements over the supervised baseline and vanilla MIM

[2] https://dl.fbaipublicfiles.com/mae/visualize/mae_visualize_vit_base.pth

pre-training under various settings. Concretely, with only 800 epochs pre-training, HPM achieves 84.2% and 85.8% Top-1 accuracy on ImageNet-1K [49] using ViT-B and ViT-L, outperforming MAE [19] pre-trained with 1600 epochs by +0.6% and +0.7%, respectively.

2. Related Work

Self-supervised learning. Aiming at learning from data without any annotations, self-supervised learning (SSL) approaches have raised significant interest in computer vision, and how to design an appropriate pretext task becomes the crux [12, 41, 58, 71]. Among them, contrastive learning [18, 20, 41, 60] based on instance discrimination [64] becomes popular. The core idea lies in urging the model to learn view-invariant features, and thus these methods strongly depend on data augmentations [6, 18]. MIM pursues a conceptually different direction with different behaviors.

Masked image modeling. Since MLM [4, 11, 44, 45] and its autoregressive variants have achieved great success in NLP, MIM, its counterpart in CV, has attracted numerous interests of many researchers [3, 7, 19, 61, 67, 73], with the goal of building a unified self-supervised pre-training framework. Specifically, for MIM, a Vision Transformer (*e.g.*, ViT [14] or its hierarchical variants [38, 39, 57]) is trained to predict pre-defined targets (*e.g.*, discrete tokens [3] generated by a dVAE [47] pre-trained on DALLE [46], raw RGB pixels [19, 29, 36, 67], HoG features [61], frequency [35, 66], and features from a momentum teacher [2, 13, 63, 69, 73]) of masked patches. Also, it has been verified to be an efficient pre-training framework in video understanding [52, 55], cross-modality [1, 17, 22, 26], and 3D cases [34, 40, 42, 70].

Mask strategies in masked image modeling. In NLP, a word is already highly semantic, and thus vanilla random masking brings a challenging pretext task [11, 14], By contrast, the success of masked image modeling heavily relies on the mask strategies due to the spatial information redundancy [19] in computer vision. Concretely, MAE [19] uses a large mask ratio (*i.e.*, 75%), BEiT [3] adopts block-wise masking, and SimMIM [67] finds that larger mask kernels (*e.g.*, 32×32) are more robust against different mask ratios. Furthermore, AttMask [23] masks patches with high attention signals, bringing a more challenging pretext task. ADIOS [50] trains an extra U-Net [48] based masking model by adversarial objectives. SemMAE [28] regards semantic parts as the visual analog of words, and trains an extra StyleGAN [24] based decoder distilled by iBOT [73]. UM-MAE [29] masks one patch in each 2×2 local window, enabling pyramid-based ViTs (*e.g.*, PVT [57], CoaT [68], and Swin [38, 39]) to take the random sequence of partial vision tokens as input. All the masking models of these methods are either pre-defined [2, 3, 19, 23, 61, 67, 73] or separately learned [28, 50]. However, we argue that *learn to mask* the discriminative parts is crucial, which can not only guide the model in a more challenging manner, but also bring salient prior of input images, bootstrapping the performance on a wide range of downstream tasks hence.

3. Method

In this section, we first give an overview of our proposed HPM in Sec. 3.1. Then, the two objectives in HPM, *i.e.*, reconstruction loss and predicting loss are introduced in Sec. 3.2 and Sec. 3.3, respectively. Finally, in Sec. 3.4, the easy-to-hard mask generation manner is described, together with the pseudo-code of the overall training procedure.

3.1. Overview

Introduced in Fig. 1 and Sec. 1, conventional MIM pre-training solutions can be considered as training a student to solve *given* problems, while we argue that *making the model stand in the shoes of a teacher*, producing challenging pretext task is crucial. To achieve this, we introduce an auxiliary decoder to predict the reconstruction loss of each masked patch, and carefully design its objective. Fig. 3 gives an overview of our proposed HPM, introduced next.

HPM consists of a student (f_{θ_s}, d_{ϕ_s}, and d_{ψ_s}) and a teacher (f_{θ_t}, d_{ϕ_t}, and d_{ψ_t}) with the same network architecture. $f_\theta(\cdot)$, $d_\phi(\cdot)$, and $d_\psi(\cdot)$ are encoder, image reconstructor, and reconstruction loss predictor, parameterized by θ, ϕ, and ψ, respectively. The subscript t stands for teacher and s stands for student. To generate consistent predictions (especially for the reconstruction loss predictor), momentum update [20] is applied to the teacher:

$$\boldsymbol{\theta}_t \leftarrow m\boldsymbol{\theta}_t + (1-m)\boldsymbol{\theta}_s, \tag{1}$$

where $\boldsymbol{\theta}_t = (\theta_t, \phi_t, \psi_t)$, $\boldsymbol{\theta}_s = (\theta_s, \phi_s, \psi_s)$, and m denotes the momentum coefficient.

At each training iteration, an input image $\mathbf{I} \in \mathbb{R}^{H\times W\times C}$ is reshaped into a sequence of 2D patches $\mathbf{x} \in \mathbb{R}^{N\times(P^2 C)}$. (H, W) is the resolution of the original image, C is the number of channels, P is the patch size (*e.g.*, 16), and $N = HW/P^2$ hence. Then, $\mathbf{x}$ is fed into the teacher to get patch-wise predicted reconstruction loss $\hat{\mathcal{L}}^t = d_{\psi_t}(f_{\theta_t}(\mathbf{x}))$ described in Sec. 3.2. Based on predicted reconstruction loss $\hat{\mathcal{L}}^t$ and the training status, a binary mask $\mathbf{M} \in \{0,1\}^N$ is generated under an easy-to-hard manner introduced later in Sec. 3.4. The student is trained based on two objectives, *i.e.*, reconstruction loss (Sec. 3.2) and predicting loss (Sec. 3.3)

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{pred}}, \tag{2}$$

where these two objectives work in an alternating way, and reinforce each other to extract better representations, by gradually urging the student to reconstruct hard patches within an image.

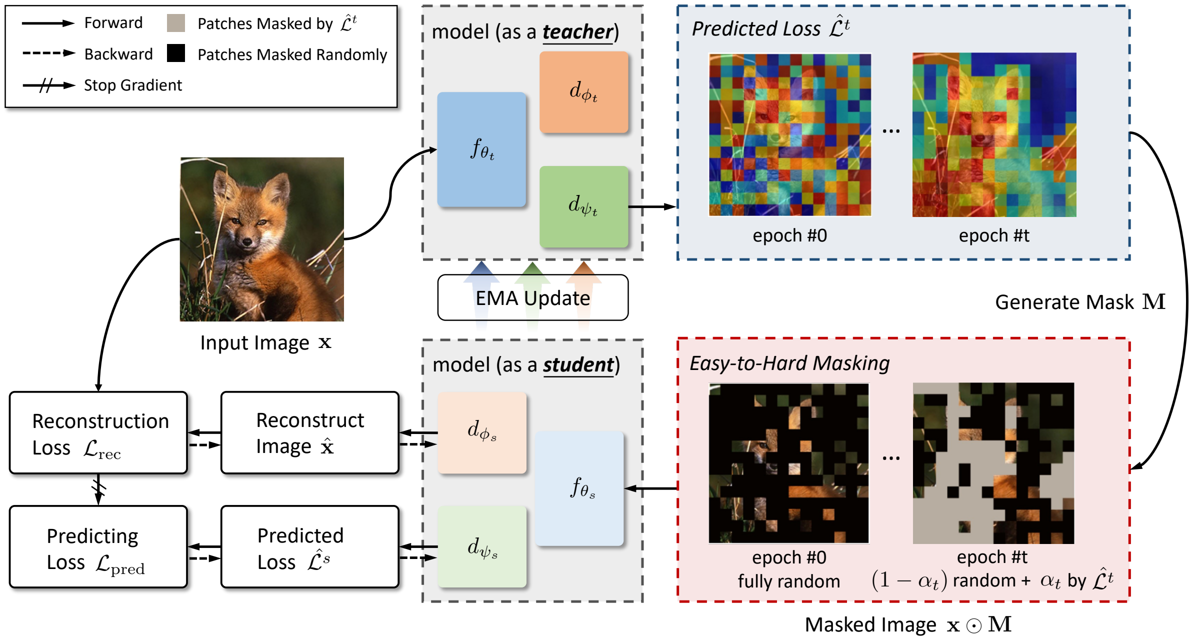

Forward
Backward
Stop Gradient
Patches Masked by $\hat{\mathcal{L}}^t$
Patches Masked Randomly
model (as a ***teacher***)
f_{θ_t}
d_{ϕ_t}
d_{ψ_t}
Predicted Loss $\hat{\mathcal{L}}^t$
epoch #0
...
epoch #t
EMA Update
Generate Mask $\mathbf{M}$
Input Image $\mathbf{x}$
model (as a ***student***)
Easy-to-Hard Masking
Reconstruction Loss $\mathcal{L}_{\text{rec}}$
Reconstruct Image $\hat{\mathbf{x}}$
d_{ϕ_s}
f_{θ_s}
Predicting Loss $\mathcal{L}_{\text{pred}}$
Predicted Loss $\hat{\mathcal{L}}^s$
d_{ψ_s}
epoch #0
fully random
...
epoch #t
$(1-\alpha_t)$ random + α_t by $\hat{\mathcal{L}}^t$
Masked Image $\mathbf{x} \odot \mathbf{M}$

Figure 3. **Illustration of our proposed HPM**, containing a student network and a teacher network, where the teacher is updated by the student in an exponential moving average (EMA) manner. Each network consists of an encoder f_θ, an image reconstructer d_ϕ, and a loss predictor d_ψ, parameterized by θ, ϕ, and ψ, respectively. For each image during pre-training, it is first fed into the teacher to predict the patch-wise reconstruction loss. Then, a binary mask is generated based on the current epoch and the predicted loss. Finally, only visible patches are fed into the student to 1) reconstruct masked patches defined in Eq. (3), and 2) predict relative loss defined in Eq. (5).

3.2. Image Reconstructor

Masked image modeling aims at training an autoencoder (*i.e.*, image reconstructor) to reconstruct the masked portion according to pre-defined targets, *e.g.*, raw RGB pixels [7, 19, 25, 37, 67, 69] and specific features [2, 3, 61, 63, 73].

$$\mathcal{L}_{\text{rec}} = \mathcal{M}\left(d_{\phi_s}(f_{\theta_s}(\mathbf{x} \odot \mathbf{M})), \mathcal{T}(\mathbf{x} \odot (1-\mathbf{M}))\right), \quad (3)$$

where for conventional approaches, the binary mask $\mathbf{M} \in \{0,1\}^N$ is generated by a pre-defined manner. $\odot$ means element-wise dot product, and thus $\mathbf{x} \odot \mathbf{M}$ represents unmasked (*i.e.*, visible) patches and vice versa. $\mathcal{T}(\cdot)$ is the transformation function, generating reconstructed targets. $\mathcal{M}(\cdot,\cdot)$ represents the similarity measurement, *e.g.*, ℓ_2-distance [19], smooth ℓ_1-distance [67], knowledge distillation [13, 73], and cross-entropy [3].

3.3. Hard Patches Mining with a Loss Predictor

It is widely known that in NLP, each word in a sentence is already highly semantic [19]. Training a model to predict only a few missing words tends to be a challenging task in understanding languages [4, 11, 44, 45]. While in CV, on the contrary, an image is with heavy spatial redundancy, and thus plenty of mask strategies are proposed to deal with this issue [3, 19, 23, 28, 50, 67].

Apart from designing a challenging situation by prior knowledge, we argue that *the ability to produce demanding scenarios* is also crucial for MIM pre-training. Intuitively, we consider patches with high reconstruction loss defined in Eq. (3) as *hard patches*, which implicitly indicate the most discriminative parts of an image, which is verified in Fig. 2. Therefore, if the model is equipped with the ability to predict the reconstruction loss for each patch, simply masking those hard patches becomes a more challenging pretext task.

To this end, we employ an extra loss predictor (*i.e.*, d_ψ in Fig. 3) to mine hard patches during training. Next, we will introduce how to design the objective for loss predictor with two variants: 1) absolute loss and 2) relative loss.

Absolute loss. The simplest and the most straightforward way is to define the objective in an MSE manner.

$$\mathcal{L}_{\text{pred}} = \left(d_{\psi_s}(f_{\theta_s}(\mathbf{x} \odot \mathbf{M})) - \mathcal{L}_{\text{rec}}\right)^2 \odot (1-\mathbf{M}), \quad (4)$$

where d_{ψ_s} is the auxiliary decoder of the student parameterized by ψ_s, and $\mathcal{L}_{\text{rec}}$ here is detached from gradient, being a ground-truth for loss prediction. However, recall that our goal is to determine *hard patches* within an image, thus we need to learn the *relative relationship among patches*. Under such a setting, MSE is not the most suitable choice hence, since the scale of $\mathcal{L}_{\text{rec}}$ decreases as training goes on, and thus the loss predictor may be overwhelmed by the scale and the exact value of $\mathcal{L}_{\text{rec}}$. For this purpose, we propose a binary cross-entropy-based relative loss as an alternative.

Relative loss. Given a sequence of reconstruction loss

$\mathcal{L}_{\text{rec}} \in \mathbb{R}^N$, we aim to predict $\texttt{argsort}(\mathcal{L}_{\text{rec}})$ by using a relative loss. That is because, within an image, the patch-wise difficulty of the reconstruction task can be measured by $\texttt{argsort}(\mathcal{L}_{\text{rec}})$. However, as the $\texttt{argsort}(\cdot)$ operation is non-differentiable, it is hard to directly minimize some custom distances between $\texttt{argsort}(d_{\psi_s}(f_{\theta_s}(\mathbf{x}\odot\mathbf{M})))$ and $\texttt{argsort}(\mathcal{L}_{\text{rec}})$. Therefore, we translate this problem into an equivalent one: *dense relation comparison*. Specifically, for each pair of patches (i, j), where $i, j = 1, 2, \cdots, N$ and $i \neq j$, we can implicitly learn $\texttt{argsort}(\mathcal{L}_{\text{rec}})$ by predicting the relative relation of $\mathcal{L}_{\text{rec}}(i)$ and $\mathcal{L}_{\text{rec}}(j)$, *i.e.*, which one is larger. The objective is defined as follows:

$$\begin{aligned}\mathcal{L}_{\text{pred}} = &-\sum_{i=1}^{N}\sum_{\substack{j=1\\ j\neq i}}^{N} \mathbb{1}_{ij}^{+} \log\left(\sigma(\hat{\mathcal{L}}_i^s - \hat{\mathcal{L}}_j^s)\right) \\ &-\sum_{i=1}^{N}\sum_{\substack{j=1\\ j\neq i}}^{N} \mathbb{1}_{ij}^{-} \log\left(1-\sigma(\hat{\mathcal{L}}_i^s - \hat{\mathcal{L}}_j^s)\right),\end{aligned} \tag{5}$$

where $\hat{\mathcal{L}}^s = d_{\psi_s}(f_{\theta_s}(\mathbf{x}\odot\mathbf{M})) \in \mathbb{R}^N$ represents the predicted loss from the student, and $i, j = 1, 2, \ldots, N$ are patch indexes. $\sigma(\cdot)$ indicates $\texttt{sigmoid}$ function, *i.e.*, $\sigma(z) = e^z/(e^z+1)$. $\mathbb{1}_{ij}^{+}$ and $\mathbb{1}_{ij}^{-}$ are two indicators, representing the relative relationship of ground-truth reconstruction losses, *i.e.*, $\mathcal{L}_{\text{rec}}$, between patch i and patch j

$$\mathbb{1}_{ij}^{+} = \begin{cases} 1, & \mathcal{L}_{\text{rec}}(i) > \mathcal{L}_{\text{rec}}(j) \text{ and } \mathbf{M}_i = \mathbf{M}_j = 0, \\ 0, & \text{otherwise}, \end{cases} \tag{6}$$

$$\mathbb{1}_{ij}^{-} = \begin{cases} 1, & \mathcal{L}_{\text{rec}}(i) < \mathcal{L}_{\text{rec}}(j) \text{ and } \mathbf{M}_i = \mathbf{M}_j = 0, \\ 0, & \text{otherwise}, \end{cases} \tag{7}$$

where $\mathbf{M}_i = \mathbf{M}_j = 0$ means that both patch i and j are masked during training.

3.4. Easy-to-Hard Mask Generation

With the reconstruction loss predictor, we are able to define a more challenging pretext task, *i.e.*, mask those hard/discriminative parts of an input image. Concretely, after obtaining the predicted reconstruction loss from the teacher network, *i.e.*, $\hat{\mathcal{L}}^t = d_{\psi_t}(f_{\theta_t}(\mathbf{x}))$, we conduct $\texttt{argsort}(\cdot)$ operation over $\hat{\mathcal{L}}^t$ in a descending order to obtain the relative reconstruction difficulty within the image.

However, in the early training stages, the learned feature representations are not ready for reconstruction but are overwhelmed by the rich texture, which means large reconstruction loss may not be equivalent to discriminative. To this end, we propose an easy-to-hard mask generation manner, providing some reasonable hints that guide the model to reconstruct masked hard patches step by step.

As illustrated in Fig. 3, for each training epoch t, α_t of the mask patches are generated by $\hat{\mathcal{L}}^t$, and the remaining $1 - \alpha_t$ are randomly selected. Specifically, $\alpha_t = \alpha_0 + \frac{t}{T}(\alpha_T - \alpha_0)$, where T is the total training epochs, and $\alpha_0, \alpha_T \in [0, 1]$ are two tunable hyper-parameters. We filter $\alpha_t \cdot \gamma N$ patches with the highest $\hat{\mathcal{L}}^t$ to be masked, and the remaining $(1-\alpha_t)\cdot\gamma N$ patches are randomly masked. The proportion α_t gradually increases from α_0 to α_T in a linear manner without further tuning for simplicity, contributing to an easy-to-hard training procedure.

Algorithm 1 Pseudo-Code of HPM in a PyTorch-like Style.

```
# model_s, model_t: networks for student and teacher
# t, T: current/total epochs
# x: input patchified images
# rec: reconstructed image
# pred: predicted reconstruction loss

# teacher inference
_, pred_t = model_t(x)
# easy-to-hard mask generation
mask = mask_generation(pred_t, t, T, mask_ratio)
# student forward to compute objectives
rec_x, pred_s = model_s(x * mask)
# compute losses
loss_rec = (rec_x - x[~mask]) ** 2
loss_pred = predicting_loss(pred_s, loss_rec, mask)
return loss_rec + loss_pred

# predict relative reconstruction loss
def predicting_loss(pred_s, loss_rec, mask):
    loss_rec = loss_rec[~mask].detach()
    pred_s = pred_s[~mask]
    # generate indicators
    pos = loss_rec.unsqueeze(0) > loss_rec.unsqueeze(1)
    neg = loss_rec.unsqueeze(0) < loss_rec.unsqueeze(1)
    valid = pos + neg
    # compute dense relative relationship
    pred_mat = pred_s.unsqueeze(0) > pred_s.unsqueeze(1)
    # compute predicting loss
    loss_pos = -pos * log(sigmoid(pred_mat))
    loss_neg = -neg * log(1-sigmoid(pred_mat))
    loss = loss_pos + loss_neg
    return loss.sum() / valid.sum()
```

Algorithm 1 summarizes the training procedure, together with the pseudo-code of computing the objective for training the reconstruction loss predictor. Thanks to the simple implementation of the easy-to-hard mask generation, please refer to *Supplementary Material* for the pseudo-code.

4. Experiments

Baseline. We evaluate our proposed HPM under self-supervised pre-training on ImageNet-1K [49]. We take ViT-B/16 [14] as the backbone and MAE [19] pre-trained with 200 epochs on ImageNet-1K [49] as our baseline. Our implementation is based on MAE [19] and UM-MAE [29]. More details can be found in *Supplementary Material*.

ImageNet classification. We evaluate our proposed HPM by 1) end-to-end fine-tuning, 2) linear probing, and 3) k-NN. We report Top-1 accuracy (%) on the validation set. End-to-end fine-tuning (or learning from scratch) and linear probing over image classification are trained for 100 epochs. k-NN is implemented based on DINO [5]. The resolution is kept to 224×224 on both pre-training and evaluation.

COCO object detection and instance segmentation. We take Mask R-CNN [21] with FPN [31] as the object detector, and perform end-to-end fine-tuning on COCO [33] for 1× schedule (12 epochs) for ablations (*i.e.*, Tab. 5) with 1024×1024 resolution. We report AP_{box} for object detection and AP_{mask} for instance segmentation. Our implementation is based on detectron2 [62] and ViTDet [30].

ADE20k semantic segmentation. We take UperNet [65] as the segmentor, and perform end-to-end fine-tuning on ADE20k [72] for 80k iterations for ablations (*i.e.*, Tab. 5) and 160k iterations when comparing with previous methods (*i.e.*, Tab. 7) with 512×512 resolution. We take mIoU [16] as the evaluation metric. Our implementation is based on mmsegmentation [10].

4.1. Ablation Study

We study different reconstruction targets, mask strategies, predicting loss formulations, and downstream tasks in this section. By default, ViT-B/16 [14] is used as the backbone with 200 epochs pre-training and 100 epochs fine-tuning on ImageNet-1K [49]. We highlight our default settings.

Reconstruction targets. We study the effectiveness of different reconstruction targets in Tab. 1, including regressing raw RGB pixels used in MAE [19], and distilling from various teacher models, *i.e.*, the EMA (exponential moving average) teacher used in BootMAE [13], and pre-trained teachers obtained from DINO [5] and CLIP [43]. All these teacher models share the same architecture, *i.e.*, ViT-B/16 [14].

It has been substantiated that directly regressing RGB values of pixels is a simple yet efficient way in MIM pre-training [19]. However, due to the existence of high-frequency noise in some cases, patches with higher frequency tend to have larger reconstruction loss, and thus *hard patches may not be highly semantic* under this setting, which is quite the opposite from our motivation: learn to mine *discriminative* parts of an image instead of high-frequency parts. To this end, we further take features from a teacher model to be the learning target (*e.g.*, DINO [5] and CLIP [43]), to verify the effectiveness of our proposed HPM.

Note that the objective differs when using different reconstruction targets. Specifically, an MSE loss is adopted for RGB regression following MAE [19], while for knowledge distillation cases, we first apply ℓ_2 normalization to the features output from the teacher and the student, and then minimize their MSE distances. This can be also implemented by maximizing their cosine similarities.

As illustrated in Tab. 1, our HPM is able to bootstrap the performances under various learning targets. Taking the pixel regression case as an instance, equipped with the predicting loss and the easy-to-hard mask generation manner, the fine-tuning Top-1 accuracy achieves 82.95%, outperforming MAE [19] by +0.72%. Notably, *only* applying an auxiliary decoder to predict reconstruction loss for each patch brings an improvement of +0.26% fine-tuning accuracy, achieving 82.49%, verifying that *the ability to mine hard patches* brings better extracted feature representations. Then, fully taking advantage of this capability, *i.e.*, generate challenging masks, can further bootstrap the performances, which appears *consistently across different learning targets*.

Table 1. Ablation study on different **reconstruction targets**. We study four different targets, including raw RGB pixels (MAE [19] baseline), and three knowledge distillation targets, *i.e.*, features from the EMA (exponential moving average) model, DINO [5], and CLIP [43]. All cases are pre-trained 200 epochs on ImageNet [49] with ViT-B/16 [14].

target	$\mathcal{L}_{\text{pred}}$	learn to mask	fine-tune	linear	k-NN
Pixel Regression					
RGB (MAE [19]	-	-	82.23	50.80	29.84
	✓	-	82.49 ↑ 0.26	51.26	31.98
	✓	✓	**82.95 ↑ 0.72**	**54.92**	**36.09**
Feature Distillation					
EMA features	-	-	82.99	32.65	20.69
	✓	-	83.13 ↑ 0.14	52.06	35.73
	✓	✓	**83.47 ↑ 0.48**	**55.25**	**35.94**
DINO [5] features	-	-	83.46	61.31	41.53
	✓	-	83.58 ↑ 0.12	63.25	43.02
	✓	✓	**84.13 ↑ 0.67**	**64.17**	**47.25**
CLIP [43] features	-	-	83.20	59.80	42.51
	✓	-	83.31 ↑ 0.11	60.62	43.26
	✓	✓	**83.58 ↑ 0.38**	**62.22**	**45.08**

Table 2. Ablation study on different **mask strategies**. We study the effect of different α_0, α_T, and γ. Large α_T indicates a more difficult pretext task, but the randomness of this strategy decreases.

case	difficulty	randomness	γ	α_0	α_T	fine-tune
random	easy	strong	75	0	0	82.49
learn to mask	↓	↓	75	0	0.5	**82.95 ↑ 0.46**
learn to mask			75	0	1	82.67 ↑ 0.18
learn to mask	hard	weak	75	1	1	81.40 ↓ 1.09
random	easy	strong	50	0	0	82.36
learn to mask	↓	↓	50	0	0.5	**82.56 ↑ 0.20**
learn to mask	hard	weak	50	1	1	82.19 ↓ 0.17
random	easy	strong	90	0	0	82.48
learn to mask	↓	↓	90	0	0.5	**82.66 ↑ 0.18**
learn to mask	hard	weak	90	1	1	80.59 ↓ 1.89

Mask strategies. To verify that harder tasks do bring better performance, we study various mask strategies in Tab. 2, including random masking and our proposed learnable masking. With different α_0 and α_T, we can construct different strategies. For instance, $\alpha_0 = \alpha_T = 0$ indicates that predicted reconstruction losses $\hat{\mathcal{L}}^t$ will not participate in mask generation (*i.e.*, a fully random manner), $\alpha_0 =$

$\alpha_T = 1$, however, means that γN patches with the highest $\hat{\mathcal{L}}^t$ values are kept masked (see Fig. 2).

From Tab. 2, we find that the increase in the difficulty of the pretext task does not consistently lead to better performance. *Retaining a certain degree of randomness* is beneficial for satisfactory results. Specifically, $\alpha_0 = 0$ and $\alpha_T = 0.5$ achieves the best results under different mask ratio γ, which is a more difficult case over $\alpha = \alpha_T = 0$ (*i.e.*, random masking), and with stronger randomness against $\alpha_0 = \alpha_T = 1$. These conclusions are quite intuitive. Directly masking those patches with the highest $\hat{\mathcal{L}}^t$ brings the hardest problem, where discriminative parts of an image are almost masked. That means visible patches are nearly all background (see Fig. 2). *Forcing the model to reconstruct the forehead based on only these backgrounds without any hints makes no sense*, whose performance drops consistently with different values of γ. Therefore, a certain level of randomness is necessary.

We further investigate the effectiveness of producing *hard* pretext task for MIM pre-training in Tab. 3. Note that performing $\texttt{argmin}(\cdot)$ operation over predicted reconstruction loss $\hat{\mathcal{L}}^t$ means we have generated a task even easier than the random baseline. $\alpha_0 < \alpha_T$ indicates an easy-to-hard mask generation introduced in Sec. 3.4, while $\alpha_0 > \alpha_T$ means the opposite, *i.e.*, a hard-to-easy manner, which is also studied in Tab. 3. All results verify the necessity of a hard pretext task and the easy-to-hard manner. Both $\texttt{argmin}(\cdot)$ operation and the hard-to-easy mask generation manner leads to performance degradation over random masking baseline.

Predicting loss formulations. We study different designs of predicting loss in the following table, including absolute loss based on MSE introduced in Eq. (4) and relative loss based on BCE defined in Eq. (5). As expected, BCE is a better choice for mining *relative relationship* between patches, instead of absolute values of reconstruction losses as MSE does, outperforming absolute MSE by +0.18%.

Downstream tasks. We evaluate transfer learning performance using the pre-trained models in Tab. 1, including COCO [33] object detection and instance segmentation, and ADE20k [72] semantic segmentation.

As illustrated in Tab. 5, equipped with our proposed HPM, it outperforms +1.58 AP_{box} and +1.14 AP_{mask} on COCO [33], and +1.60 mIoU on ADE20k [72], over MAE [19] baseline, *i.e.*, taking raw RGB pixel as the learning target. When using CLIP [43] features as the learning target, it outperforms +0.36 AP_{box} and +0.41 AP_{mask} on COCO [33], and +0.76 mIoU on ADE20k [72] over baseline, respectively.

Notably, *only* taking the predicting loss $\mathcal{L}_{\text{pred}}$ as the extra objective manages to boost the performance across downstream tasks, verifying the effectiveness of making the model be the teacher, instead of only a student. These observations are consistent across different learning targets.

Table 3. Ablation study on different **mask strategies**. We study the effectiveness of the $\texttt{argmax}(\cdot)$ performed on predicted reconstruction loss $\hat{\mathcal{L}}^t$ and the "easy-to-hard" manner. Note that $\texttt{argmin}(\cdot)$ means that we mask those easy patches.

case	operation	γ	α_0	α_T	fine-tune
random	-	75	0	0	82.49
learn to mask	$\texttt{argmax}(\cdot)$	75	0	0.5	**82.95** ↑ **0.46**
learn to mask	$\texttt{argmin}(\cdot)$	75	0	0.5	82.36 ↓ 0.13

case	manner	γ	α_0	α_T	fine-tune
random	-	75	0	0	82.49
learn to mask	easy-to-hard	75	0	0.5	**82.95** ↑ **0.46**
learn to mask	hard-to-easy	75	0.5	0	81.71 ↓ 0.78

Table 4. Ablations on **predicting loss formulation**. We study the absolute loss introduced in Eq. (4) and the relative loss described in Eq. (5).

case	fine-tune	linear	k-NN
none (MAE [19])	82.23	51.26	31.98
absolute MSE	82.77 ↑ 0.54	51.85	34.47
relative BCE	**82.95** ↑ **0.72**	**54.92**	**36.09**

Table 5. Ablations on **downstream tasks**. We take RGB and CLIP [43] features as the learning target, representing *pixel regression* and *knowledge distillation* cases. All cases are first pre-trained 200 epochs on ImageNet-1K [49] with ViB-B/16 [14] followed by fine-tuning.

target	$\mathcal{L}_{\text{pred}}$	learn to mask	COCO		ADE20k
			AP_{box}	AP_{mask}	mIoU
RGB	-	-	40.45	37.01	40.49
	✓	-	40.98 ↑ 0.53	37.34 ↑ 0.33	41.45 ↑ 0.96
	✓	✓	**42.03** ↑ **1.58**	**38.15** ↑ **1.14**	**42.09** ↑ **1.60**
CLIP [43]	-	-	46.21	41.55	46.59
	✓	-	46.43 ↑ 0.22	41.80 ↑ 0.25	46.97 ↑ 0.38
	✓	✓	**46.57** ↑ **0.36**	**41.96** ↑ **0.41**	**47.35** ↑ **0.76**

4.2. Comparison with Previous Alternatives

We compare our proposed HPM with the supervised baseline and a wide range of self-supervised alternatives using fine-tuning accuracy in Tab. 6, where selected methods can be summarized into three mainstream: (1) contrastive learning methods [5, 9], (2) MIM with pixel regression methods [19, 67], and (3) MIM with feature distillation methods [3, 13, 73]. Effective pre-training epoch[3] is used for fair comparison following [73]. All methods are evaluated under the same input size *i.e.*, 224×224. We take raw RGB as the learning target following [19, 67].

Notably, with only 200 epochs pre-training, our HPM achieves 83.0% and 84.5% Top-1 accuracy with ViT-B and ViT-L backbone, respectively, surpassing MAE [19] by +0.8% and +1.2%, and the supervised baseline by +2.1%

[3] Effective pre-training epochs accounts the actual trained images/views defined by [73]. Details can be found in *Supplementary Material*.

Table 6. Comparison with state-of-the-art alternatives on **ImageNet-1K**. All methods are evaluated by fine-tuning. The resolution of images is 224×224 for both pre-training and fine-tuning. † means our implementation. ‡ means the result is borrowed from [19].

method		eff. ep.	ViT-B	ViT-L
scratch		-	80.9†	82.6‡
Contrastive Learning				
MoCo v3‡ [9]	[ICCV'21]	600	83.2	84.1
DINO‡ [5]	[ICCV'21]	1600	83.6	-
MIM with Pixel Regression				
MAE [19]	[CVPR'22]	200	82.2†	83.3‡
HPM	[Ours]	200	**83.0**	**84.5**
MAE‡ [19]	[CVPR'22]	1600	83.6	85.1
SimMIM [67]	[CVPR'22]	800	83.8	-
HPM	[Ours]	800	**84.2**	**85.8**
MIM with Feature Distillation				
BEiT‡ [3]	[ICLR'22]	800	83.2	85.2
iBOT [73]	[ICLR'22]	1600	84.0	-
BootMAE [13]	[ECCV'22]	800	84.2	85.9

Table 7. Comparison with state-of-the-art alternatives on **ADE20k semantic segmentation** using UperNet. We take mIoU as the metric. ‡ means the result is borrowed from [19].

method		ViT-B	ViT-L
supervised‡		47.4	49.9
MoCo v3‡ [9]	[ICCV'21]	47.3	49.1
BEiT‡ [3]	[ICLR'22]	47.1	53.3
MAE‡ [19]	[CVPR'22]	48.1	53.6
SemMAE [28]	[NeurIPS'22]	46.3	-
HPM	[Ours]	**48.5**	**54.6**

and +1.9%, respectively. With a longer training schedule, *i.e.*, 800 epochs, HPM achieves 84.2% and 85.8% Top-1 accuracy with ViT-B and ViT-L backbone, outperforming MAE [19] by +0.6% and +0.7%, respectively. Strikingly, HPM reaches comparable results with *feature distillation* alternative BootMAE [13]. From Tab. 1, taking EMA features as the learning target for HPM, which is the same as BootMAE [13], can further improve the performance by $\sim 0.5\%$.

Semantic Segmentation. We experiment on ADE20k [72] using UperNet [65] for 160k iterations in Tab. 7. From the table, we can tell that our HPM significantly improves performance over supervised pre-training by +1.1 mIoU (48.5 *v.s.* 47.4) with ViT-B and +4.7 mIoU (54.6 *v.s.* 49.9) with ViT-L, respectively. More importantly, our HPM outperforms self-supervised alternatives under all settings. For example, with ViT-L, HPM surpasses MAE [19] by +1.0 (54.6 *v.s.* 53.6) mIoU.

Visualization of predicted losses. We provide qualitative results on COCO [33] *validation* set in Fig. 4, where the model has *never* seen this dataset. Patches with higher *predicted* reconstruction loss usually are more discriminative.

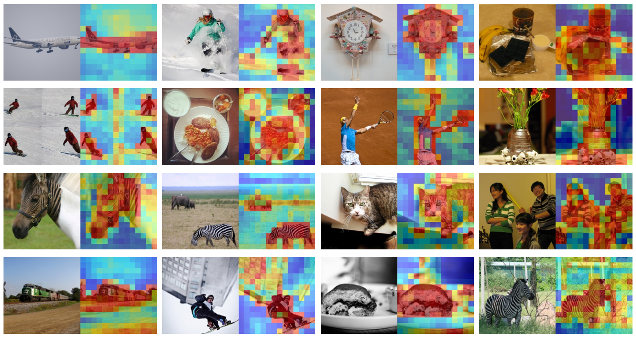


Figure 4. Visualization on **COCO** *validation* set. For each tuple, we show the *image* (left) and *predicted* reconstruction losses (right).

5. Conclusion

In this paper, we find it necessary to *make the model stand in the shoes of a teacher* for MIM pre-training, and verify that the patch-wise reconstruction loss can naturally be the metric of the reconstruction difficulty. To this end, we propose HPM, which introduces an auxiliary reconstruction loss prediction task, and thus guides the training procedure iteratively in a produce-and-solve manner. Experimentally, HPM bootstraps the performance of masked image modeling across various downstream tasks. Ablations across different learning targets show that HPM, as a plug-and-play module, can be effortlessly incorporated into existing frameworks (*e.g.*, pixel regression [19, 67] and feature prediction [13, 61, 73]) and bring *consistent* performance improvements.

Broader impact. Techniques that mine hard examples are widely used in object detection [27, 32, 51]. Loss prediction can be a brand-new alternative. Furthermore, it can be also used as a technique to filter high-quality pseudo-labels in label-efficient learning [15, 59, 60]. Meanwhile, as shown in Fig. 2 and Fig. 4, *the salient area tends to have a higher predicted loss*, and thus HPM may also be used for saliency detection [56] and unsupervised segmentation [53, 54]. We hope these perspectives will inspire future work.

Discussion. As a common problem of MIM, the performances of linear probing and k-NN classification are not as comparable as contrastive learning alternatives [19]. In addition, HPM needs more computation cost due to the extra decoder. It takes $\sim 1.1\times$ time to train our HPM with ViT-L [14] against MAE [19] baseline. How to design a loss prediction task without an extra auxiliary decoder can be further studied.

Acknowledgements

This work was supported in part by the Major Project for New Generation of AI (No. 2018AAA0100400), the National Natural Science Foundation of China (No. 61836014, No. U21B2042, No. 62072457, No. 62006231), and the InnoHK program.

References

[1] Roman Bachmann, David Mizrahi, Andrei Atanov, and Amir Zamir. Multimae: Multi-modal multi-task masked autoencoders. In *European Conference on Computer Vision (ECCV)*, 2022. 3

[2] Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. Data2vec: A general framework for self-supervised learning in speech, vision and language. In *International Conference on Machine Learning (ICML)*, 2022. 3, 4

[3] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. In *International Conference on Learning Representations (ICLR)*, 2022. 1, 2, 3, 4, 7, 8

[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 1, 3, 4

[5] 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)*, 2021. 5, 6, 7, 8

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, 2020. 1, 3

[7] Xiaokang Chen, Mingyu Ding, Xiaodi Wang, Ying Xin, Shentong Mo, Yunhao Wang, Shumin Han, Ping Luo, Gang Zeng, and Jingdong Wang. Context autoencoder for self-supervised representation learning. *arXiv preprint arXiv:2202.03026*, 2022. 3, 4

[8] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1

[9] 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)*, 2021. 1, 7, 8

[10] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation, 2020. 6

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019. 1, 3, 4

[12] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 3

[13] 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)*, 2022. 1, 3, 4, 6, 7, 8

[14] 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. 2, 3, 5, 6, 7, 8

[15] Ye Du, Yujun Shen, Haochen Wang, Jingjing Fei, Wei Li, Liwei Wu, Rui Zhao, Zehua Fu, and Qingjie Liu. Learning from future: A novel self-training framework for semantic segmentation. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 8

[16] Mark Everingham, SM Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision (IJCV)*, 2015. 6

[17] Xinyang Geng, Hao Liu, Lisa Lee, Dale Schuurams, Sergey Levine, and Pieter Abbeel. Multimodal masked autoencoders learn transferable representations. In *International Conference on Machine Learning Workshop (ICMLW)*, 2022. 3

[18] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 1, 3

[19] 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)*, 2022. 1, 2, 3, 4, 5, 6, 7, 8

[20] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 3

[21] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017. 6

[22] Zejiang Hou, Fei Sun, Yen-Kuang Chen, Yuan Xie, and Sun-Yuan Kung. Milan: Masked image pretraining on language assisted representation. *arXiv preprint arXiv:2208.06049*, 2022. 3

[23] Ioannis Kakogeorgiou, Spyros Gidaris, Bill Psomas, Yannis Avrithis, Andrei Bursuc, Konstantinos Karantzalos, and Nikos Komodakis. What to hide from your students: Attention-guided masked image modeling. In *European Conference on Computer Vision (ECCV)*, 2022. 3, 4

[24] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[25] Xiangwen Kong and Xiangyu Zhang. Understanding masked image modeling via learning occlusion invariant feature. *arXiv preprint arXiv:2208.04164*, 2022. 4

[26] Gukyeong Kwon, Zhaowei Cai, Avinash Ravichandran, Erhan Bas, Rahul Bhotika, and Stefano Soatto. Masked vision and language modeling for multi-modal representation learning. *arXiv preprint arXiv:2208.02131*, 2022. 1, 3

[27] Buyu Li, Yu Liu, and Xiaogang Wang. Gradient harmonized single-stage detector. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2019. 8

[28] Gang Li, Heliang Zheng, Daqing Liu, Bing Su, and Changwen Zheng. Semmae: Semantic-guided masking for learning masked autoencoders. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 3, 4, 8

[29] Xiang Li, Wenhai Wang, Lingfeng Yang, and Jian Yang. Uniform masking: Enabling mae pre-training for pyramid-based vision transformers with locality. *arXiv preprint arXiv:2205.10063*, 2022. 2, 3, 5

[30] Yanghao Li, Saining Xie, Xinlei Chen, Piotr Dollar, Kaiming He, and Ross Girshick. Benchmarking detection transfer learning with vision transformers. *arXiv preprint arXiv:2111.11429*, 2021. 6

[31] 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)*, 2017. 6

[32] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017. 8

[33] 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)*, 2014. 6, 7, 8

[34] Haotian Liu, Mu Cai, and Yong Jae Lee. Masked discrimination for self-supervised learning on point clouds. In *European Conference on Computer Vision (ECCV)*, 2022. 3

[35] Hao Liu, Xinghua Jiang, Xin Li, Antai Guo, Deqiang Jiang, and Bo Ren. The devil is in the frequency: Geminated gestalt autoencoder for self-supervised visual pre-training. *arXiv preprint arXiv:2204.08227*, 2022. 3

[36] Jihao Liu, Xin Huang, Yu Liu, and Hongsheng Li. Mixmim: Mixed and masked image modeling for efficient visual representation learning. *arXiv preprint arXiv:2205.13137*, 2022. 3

[37] Xingbin Liu, Jinghao Zhou, Tao Kong, Xianming Lin, and Rongrong Ji. Exploring target representations for masked autoencoders. *arXiv preprint arXiv:2209.03917*, 2022. 4

[38] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[39] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 3

[40] Chen Min, Dawei Zhao, Liang Xiao, Yiming Nie, and Bin Dai. Voxel-mae: Masked autoencoders for pre-training large-scale point clouds. *arXiv preprint arXiv:2206.09900*, 2022. 3

[41] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 3

[42] Yatian Pang, Wenxiao Wang, Francis EH Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. In *European Conference on Computer Vision (ECCV)*, 2022. 1, 3

[43] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021. 6, 7

[44] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018. 1, 3, 4

[45] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. 2019. 1, 3, 4

[46] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning (ICML)*, 2021. 3

[47] Jason Tyler Rolfe. Discrete variational autoencoders. In *International Conference on Learning Representations (ICLR)*, 2017. 3

[48] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 2015. 3

[49] 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)*, 2015. 3, 5, 6, 7

[50] Yuge Shi, N Siddharth, Philip Torr, and Adam R Kosiorek. Adversarial masking for self-supervised learning. In *International Conference on Machine Learning (ICML)*, 2022. 3, 4

[51] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 8

[52] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 3

[53] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. Unsupervised semantic segmentation by contrasting object mask proposals. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 8

[54] Wouter Van Gansbeke, Simon Vandenhende, and Luc Van Gool. Discovering object masks with transformers for unsupervised semantic segmentation. *arXiv preprint arXiv:2206.06363*, 2022. 8

[55] Rui Wang, Dongdong Chen, Zuxuan Wu, Yinpeng Chen, Xiyang Dai, Mengchen Liu, Yu-Gang Jiang, Luowei Zhou, and Lu Yuan. Bevt: Bert pretraining of video transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[56] Wenguan Wang, Qiuxia Lai, Huazhu Fu, Jianbing Shen, Haibin Ling, and Ruigang Yang. Salient object detection in the deep learning era: An in-depth survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2021. 8

[57] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 3

[58] Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2015. 3

[59] Yuchao Wang, Jingjing Fei, Haochen Wang, Wei Li, Liwei Wu, Rui Zhao, and Yujun Shen. Balancing logit variation for long-tail semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 8

[60] Yuchao Wang, Haochen Wang, Yujun Shen, Jingjing Fei, Wei Li, Guoqiang Jin, Liwei Wu, Rui Zhao, and Xinyi Le. Semi-supervised semantic segmentation using unreliable pseudo-labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3, 8

[61] Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 3, 4, 8

[62] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019. 6

[63] Zhirong Wu, Zihang Lai, Xiao Sun, and Stephen Lin. Extreme masking for learning instance and distributed visual representations. *arXiv preprint arXiv:2206.04667*, 2022. 3, 4

[64] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[65] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *European Conference on Computer Vision (ECCV)*, 2018. 6, 8

[66] Jiahao Xie, Wei Li, Xiaohang Zhan, Ziwei Liu, Yew Soon Ong, and Chen Change Loy. Masked frequency modeling for self-supervised visual pre-training. *arXiv preprint arXiv:2206.07706*, 2022. 1, 3

[67] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 3, 4, 7, 8

[68] Weijian Xu, Yifan Xu, Tyler Chang, and Zhuowen Tu. Co-scale conv-attentional image transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 3

[69] Kun Yi, Yixiao Ge, Xiaotong Li, Shusheng Yang, Dian Li, Jianping Wu, Ying Shan, and Xiaohu Qie. Masked image modeling with denoising contrast. *arXiv preprint arXiv:2205.09616*, 2022. 1, 3, 4

[70] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[71] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European Conference on Computer Vision (ECCV)*, 2016. 3

[72] 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)*, 2017. 6, 7, 8

[73] 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. 3, 4, 7, 8