PixMIM: Rethinking Pixel Reconstruction in Masked Image Modeling

Anonymous authors
Paper under double-blind review

Abstract

Masked Image Modeling (MIM) has achieved promising progress with the advent of Masked Autoencoders (MAE) and BEiT. However, subsequent works have complicated the framework with new auxiliary tasks or extra pre-trained models, inevitably increasing computational overhead. This paper undertakes a fundamental analysis of MIM from the perspective of pixel reconstruction, which examines the input image patches and reconstruction target, and highlights two critical but previously overlooked bottlenecks. Based on this analysis, we propose a remarkably simple and effective method, PixMIM, that entails two strategies: 1) filtering the high-frequency components from the reconstruction target to de-emphasize the network's focus on texture-rich details and 2) adopting a conservative data transform strategy to alleviate the problem of missing foreground in MIM training. PixMIM can be easily integrated into most existing pixel-based MIM approaches (*i.e.*, using raw images as reconstruction target) with negligible additional computation. Without bells and whistles, our method consistently improves four MIM approaches, MAE, MFF, ConvMAE, and LSMAE, across various downstream tasks. We believe this effective plug-and-play method will serve as a strong baseline for self-supervised learning and provide insights for future improvements of the MIM framework. Code and models will be available.

1 Introduction

Recent years have witnessed substantial progress in self-supervised learning (SSL). Inspired by the success of masked language modeling (MLM) in language processing, masked image modeling (MIM) has been introduced to computer vision, leading to rapid growth in SSL. Pioneering works such as BEiT (Bao et al., 2021) and MAE (He et al., 2022) exploit Vision Transformers (ViT) to learn discriminative visual representations from raw image data without manual annotations, and their transfer learning performance has outperformed the supervised learning counterpart.

Early MIM methods share a simple pipeline – a portion of non-overlapped image patches are randomly masked, and the model learns to extract discriminative representations by reconstructing the pixel or feature values of the masked patches (Bao et al., 2021; He et al., 2022; Xie et al., 2022). To improve the representation quality, some advanced MIM works (Huang et al., 2022; Zhou et al., 2022) incorporate extra auxiliary tasks (*e.g.*, contrastive learning) while some other efforts leverage powerful pre-trained models for distillation (Hou et al., 2022; Peng et al., 2022a). However, these attempts either complicate the overall framework or inevitably introduce non-negligible training costs.

Unlike recent works, this paper investigates the most fundamental but usually overlooked components in the data reconstruction process of MIM, *i.e.*, *the input image patches and the reconstruction target*, and proposes a simple yet effective method that improves a wide range of existing MIM methods while introducing minimal computation overhead. The core of the paper is a meticulous analysis based on the milestone algorithm – MAE (He et al., 2022), which discloses critical but neglected bottlenecks of most pixel-based MIM methods. The analysis yields two important observations:

(1). Reconstruction target: Since the advent of MAE, most MIM methods have adopted raw pixels as the reconstruction target. The training objective requires perfect reconstruction of the masked patches, including

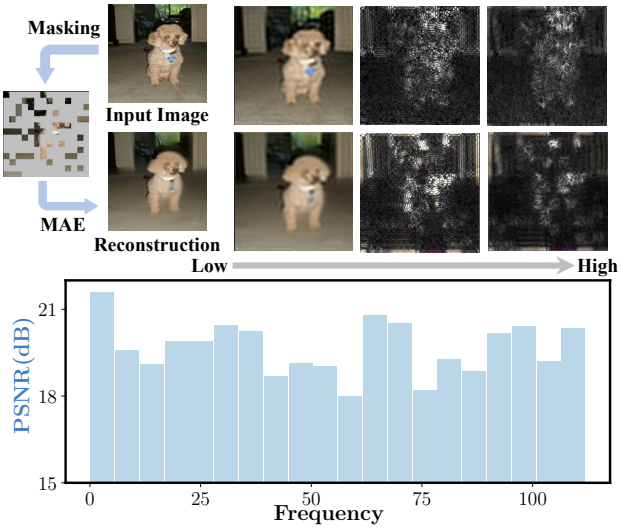

Figure 1: **Frequency analysis with an example image.** (Top) Components belonging to different frequency intervals for the input and MAE-reconstructed image. (Bottom) The peak signal-to-noise ratio (PSNR) of the reconstruction for components of fine-grained frequency intervals. MAE tends to focus on both the low and high-frequency components and reconstructs intricate details. We increased the image brightness for better visualization.

intricate details, *e.g.*, textures. This perfect reconstruction target tends to waste the modeling capacity on short-range dependencies and high-frequency details (see Figure 1), which has been pointed out by BEiT (Bao et al., 2021) and also broadly studied in generative models (Ramesh et al., 2021; Rombach et al., 2022). In addition, studies on the shape and texture bias (Geirhos et al., 2019; 2021) indicate that models relying more on shape biases usually exhibit better transferability and robustness. However, reconstructing the fine-grained details inevitably introduces biases toward textures, thus impairing the representation quality.
(2). Input patches: MAE employs the commonly used Random Resized Crop (RRC) for generating augmented images. However, when coupling the RRC with an aggressive masking strategy (*i.e.*, masking out 75% image patches), the visible patches in MAE's input can only cover 17.1% of the key object on average (see Figure 2 for examples and Section 3.2 for details). Semantic-rich foregrounds are vital for learning good visual features (Touvron et al., 2022). The low foreground coverage during training likely hinders the model's ability to effectively capture the shape and semantic priors, thus limiting the quality of representation.

Guided by the analysis, we propose a method consisting of two simple yet effective modifications to the MIM framework. Firstly, we apply an ideal low-pass filter to the raw images to produce the reconstruction targets, such that the representation learning prioritizes the low-frequency components (e.g., shapes and global patterns). Secondly, we substitute the commonly used RRC with a more conservative image transform operation, *i.e.*, the Simple Resized Crop (SRC) employed by AlexNet (Krizhevsky et al., 2012), which helps to preserve more foreground information in the inputs and encourages the model to learn more discriminative representation. As our method operates directly on raw pixels of the input patches and reconstruction targets to improve the MIM framework, we dub it **PixMIM**. Figure 4 illustrates the overall architecture of our method.

PixMIM can be effortlessly integrated into most existing pixel-based MIM frameworks. We thoroughly evaluate it with three well-established approaches, MAE (He et al., 2022), ConvMAE (Gao et al., 2022), MFF(Liu et al., 2023), and LSMAE (Hu et al., 2022). The experimental results demonstrate that PixMIM consistently enhances the performance of the baselines across various evaluation protocols, including the linear probing and fine-tuning on ImageNet-1K (Deng et al., 2009), the semantic segmentation on ADE20K (Zhou et al., 2018), and the object detection on COCO (Lin et al., 2014), without compromising training efficiency or relying on additional pre-trained models. We additionally conduct experiments to assess the model's robust-

ness against domain shift and exploit an off-the-shelf toolbox (Geirhos et al., 2021) to analyze the shape bias of the model, which further highlights the strength of our method. In summary, our contributions are three-fold:

- We carefully examine the reconstruction target and input patches of pixel-based MIM methods, which reveals two important but previously overlooked bottlenecks.
- Guided by our analysis, we develop a simple and effective plug-and-play method, PixMIM, which filters out the high-frequency components from the reconstruction targets and employs a simpler data transformation to maintain more object information in the inputs.
- Without bells and whistles, PixMIM consistently improves three recent MIM approaches on various downstream tasks with minimal extra computation.

Figure 2: **Visualization of MAE's input patches.** For each example, we show the original image, the image after the Random Resized Crop (RRC), and the visible patches produced by MAE's masking strategy from left to right. The coupling of RRC and an aggressive masking strategy leads to low foreground coverage in the inputs and potentially impairs the representation quality.

2 Related Works

Self-supervised Learning. Since the success of BERT (Devlin et al., 2019) and GPT series (Radford et al., 2018; Brown et al., 2020) in natural language processing, self-supervised learning (SSL) has made revolutionary progress in various areas and gradually replaced the conventional supervised learning paradigm. One main-stream SSL framework in computer vision is contrastive learning, which has shown great effectiveness in image (Chen et al., 2020b; He et al., 2020; Grill et al., 2020; Chen & He, 2021), video (Qian et al., 2021; Feichtenhofer et al., 2021; Hu et al., 2021; Liu et al., 2022b), and multi-modal data (Radford et al., 2021; Jia et al., 2021; Chen et al., 2021; Li et al., 2021). The main philosophy of contrastive learning is to enforce the model to pull augmented views of the same data samples together while pushing views of different samples apart, such that the model learns to extract discriminative representations.

Masked Image Modeling. Compared to contrastive learning, masked image modeling is another paradigm for self-supervised representation learning. It works by masking a portion of an image and enforcing the model to reconstruct these masked regions. BEiT (Bao et al., 2021) masks 60% of an image and reconstructs the features of these masked regions, output by DALL-E (Ramesh et al., 2021). Recently, there are also some attempts (Peng et al., 2022a; Hou et al., 2022) to align these masked features with features from a powerful pre-trained teacher model, *e.g.*, CLIP (Radford et al., 2021). Since these target features contain rich semantic information, the student model can achieve superior results on many downstream tasks. Instead of reconstructing these high-level features, MAE (He et al., 2022) reconstructs these masked pixel values. Besides, MAE only feeds these visible tokens into the encoder, which can speed up the pre-training by 3.1×, compared to BEiT.

3 A Closer Look at Masked Image Modeling

In this section, we first revisit the general formulation of Masked Image Modeling (MIM) and describe the fundamental components (Section 3.1). We then present a careful analysis with the milestone method, MAE (He et al., 2022), to disclose two important but overlooked bottlenecks of most pixel-based MIM approaches (Section 3.2), which guides the design of our method.

3.1 Preliminary: MIM Formulation

The MIM inherits the denoising autoencoder (Vincent et al., 2008) with a conceptually simple pipeline, which takes the corrupted images and aims to recover the masked content. The overall framework typically consists of 1) data augmentation & corruption operation, 2) the auto-encoder model, and 3) the target generator. Table 1 compares representative MIM methods based on the three components. Formally, let $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$ be the original image, and H, W are the height and width of the image, respectively. The corrupted image $\hat{\mathbf{I}}$ is generated with augmentation $\mathcal{A}(\cdot)$ and corruption $\mathcal{M}(\cdot)$, as $\hat{\mathbf{I}} = \mathcal{M}(\mathcal{A}(\mathbf{I}))$. As in supervised learning, the random resized crop (RRC) is the *de facto* operation for $\mathcal{A}(\cdot)$ in MIM (He et al., 2022; Bao et al., 2021; Xie et al., 2022). The corruption $\mathcal{M}(\cdot)$ is instantiated by masking image patches with different ratios (*e.g.* 75% in MAE (He et al., 2022) and 60% in SimMIM (Xie et al., 2022)).

Method	aug&corruption	auto-encoder	target
BEiT (Bao et al., 2021)	RRC+40% mask	ViT+Linear	DALLE
SimMIM (Xie et al., 2022)	RRC+60% mask	ViT+Linear	RGB
MaskFeat (Wei et al., 2022)	RRC+40% mask	ViT+Linear	HOG
ConvMAE (Gao et al., 2022)	RRC+75% mask	ConvViT+MSA	RGB
MAE (He et al., 2022)	RRC+75% mask	ViT+MSA	RGB

Table 1: **Empirical decomposition of MIM approaches.** aug: data augmentation. mask: mask ratio. MSA: multi-head self-attention layer. RRC: random resized crop.

The reconstruction target $\mathbf{Y}$ is also a key component of MIM methods. We denote the target generator function by $\mathcal{T}(\cdot)$, and the target is produced by $\mathbf{Y} = \mathcal{T}(\mathcal{A}(\mathbf{I}))$. The community has explored various target generation strategies $\mathcal{T}(\cdot)$, which could be roughly divided into non-parametric strategies (*e.g.*, identity function for RGB, HOG) and parametric strategies (*e.g.*, a pre-trained model like DALLE (Ramesh et al., 2021) or CLIP (Radford et al., 2021)). Our analysis focuses on the non-parametric family, as it does not rely on external pre-training data and is typically more computationally efficient.

Given the target $\mathbf{Y}$, the autoencoder model $\mathcal{G}(\cdot)$ takes the corrupted images $\hat{\mathbf{I}}$ as the input and generates the prediction $\hat{\mathbf{Y}}$. The model is then optimized by encouraging the prediction to match the pre-defined target $\mathbf{Y}$:

$$\hat{\mathbf{Y}} = \mathcal{G}(\hat{\mathbf{I}}), \quad \mathcal{L} = \mathcal{D}(\mathbf{Y}, \hat{\mathbf{Y}}) \tag{1}$$

The loss $\mathcal{L}$ is computed according to a distance measurement $\mathcal{D}(\cdot)$ (*e.g.*, $L1$ or $L2$ distance) between the prediction and the target. In the following analysis, we investigate MAE (He et al., 2022) by diagnosing its reconstruction target and input image patches, identifying two important but previously overlooked bottlenecks that could have hurt the representation quality.

3.2 Empirical Analysis with MAE

Reconstruction target. MAE and most pixel-based MIM methods enforce the model to reconstruct intricate details of raw images. These complicated details contain textures with repeated patterns and belong to the high-frequency components in the frequency domain, which are usually independent of object shapes or scene structures. However, MAE tends to make significant efforts in encoding and reconstructing high-frequency details, as shown in Figure 1.

According to recent studies on shape and texture biases (Geirhos et al., 2021; 2019), vision models with stronger shape biases behave more like human visual perception, demonstrating better robustness and performing better when transferred to downstream tasks than those with stronger texture biases. Apparently,

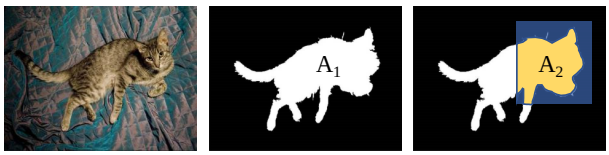

Figure 3: **Computation of object coverage percentage.** In the above example, A_1 is the foreground area. A_2 is the area of the yellow region. The blue rectangular is the cropped image produced by data augmentation. The object coverage percentage is obtained by the ratio between A_2 and A_1.

the current reconstruction target has introduced non-negligible texture biases, which deviate from the insights of previous works and might have hurt the representation quality. In Section 4.1, we provide a straightforward solution to de-emphasize the high-frequency components from the reconstruction target and justify its effectiveness with a quantitative analysis in Figure 5.

Input patches. To better understand the inputs to MIM methods at training time, we quantitatively measure how the input patches of MAE cover the foreground objects of raw images. Specifically, we adopt the binary object masks of ImageNet-1K generated by PSSL (Li et al., 2022b) and propose an object coverage percentage metric to evaluate an image processing operation $\mathcal{F}(\cdot)$, denoted by $\mathcal{J}(\mathcal{F})$. As illustrated by Figure 3, $\mathcal{J}(\mathcal{F})$ is defined as the ratio between areas A_2 and A_1. A_1 and A_2 are the areas of foreground objects in the original image $\mathbf{I}$ and the processed image $\mathcal{F}(\mathbf{I})$, respectively. We then leverage the metric to investigate how MAE's choice of $\mathcal{A}(\cdot)$ and $\mathcal{M}(\cdot)$ have influenced the object coverage. As discussed in Table 1, MAE employs the commonly used RRC for $\mathcal{A}(\cdot)$ and a masking operation with 75% mask ratio for $\mathcal{M}(\cdot)$. We found that $\mathcal{J}(\mathcal{A}) = 68.3\%$, but $\mathcal{J}(\mathcal{M} \circ \mathcal{A})$ sharply reduces to 17.1%, indicating a potential lack of foreground information in the inputs of MAE.

As argued by DeiT III (Touvron et al., 2022), the foreground usually encodes more semantics than the background, and a lack of foreground can result in sub-optimal optimization in supervised learning. In MIM, the coupling of RRC and aggressive masking might have hindered representation learning. In Section 4.2, we rigorously review various augmentation functions $\mathcal{A}(\cdot)$ and propose a simple workaround to preserve more foreground information in the input patches.

4 PixMIM

Based on the analysis, we develop a straightforward yet effective method, **PixMIM**, which addresses the two identified bottlenecks discussed in Section 3. PixMIM includes two strategies: 1) generating low-frequency reconstruction targets (Section 4.1), and 2) replacing the RRC with a more conservative augmentation (Section 4.2). An overview of PixMIM is presented in Figure 4.

4.1 Low-frequency Target Generation

To de-emphasize the model from reconstructing texture-dominated high-frequency details, we propose a novel target generator $\mathcal{G}(\cdot)$, in which we maintain the target in RGB format for efficiency but filter out the high-frequency components. Specifically, we define the low-frequency target generation with the following three steps: 1) domain conversion from spatial to frequency, 2) low-frequency components extraction, and 3) reconstruction target generation from frequency domain (see Figure 4 for an illustration).

Step-1: Domain conversion from spatial to frequency. We use the one-channel image $\mathbf{I}_i \in \mathbb{R}^{H \times W}$ to demonstrate our approach for notation simplicity. With 2D Discrete Fourier Transform (DFT) $\mathcal{F}_{\text{DFT}}(\cdot)$, the frequency representation of the image could be derived by:

$$\mathcal{F}_{\text{DFT}}(\mathbf{I}_i)(u,v) = \sum_{h=0}^{H-1} \sum_{w=0}^{W-1} \mathbf{I}_i(h,w) e^{-i2\pi(\frac{uh}{H} + \frac{vw}{W})} \tag{2}$$

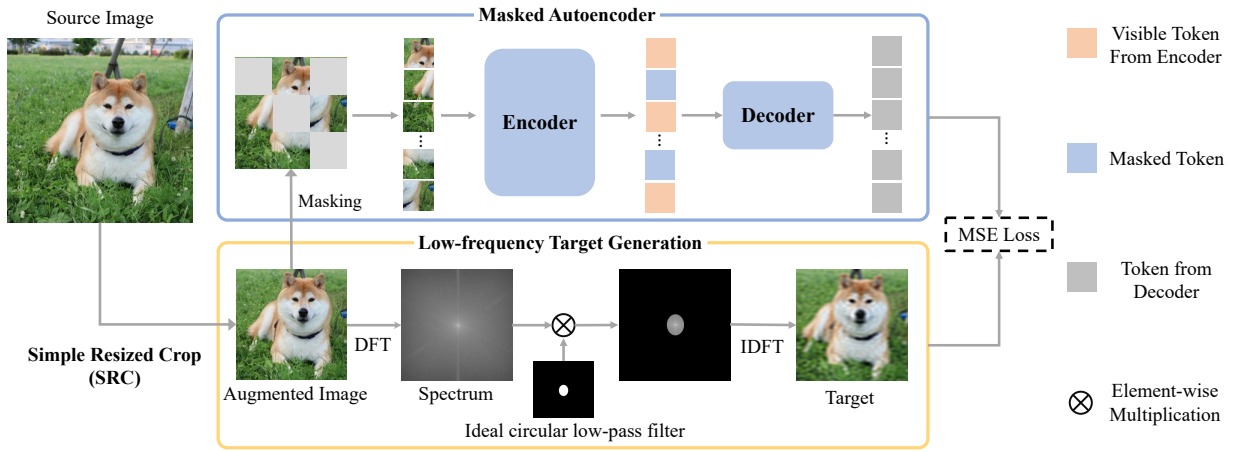

Figure 4: **The architecture of PixMIM.** Guided by the analysis (Section 3), PixMIM consists of straightforward strategies: 1) generates low-frequency reconstruction targets to de-emphasize the texture-dominated details and prioritize the learning of low-frequency patterns (Section 4.1), and 2) replace the commonly used Random Resized Crop (RRC) with the less aggressive Simple Resized Crop (SRC) to alleviate the problem of missing foreground in the input patches (Section 4.2).

Where (u, v) and (h, w) are the frequency spectrum and spatial space coordinates, respectively. $\mathcal{F}_{\text{DFT}}(\mathbf{I}_i)(u, v)$ is the complex frequency value at (u, v). $\mathbf{I}_i(h, w)$ is the pixel value at the (h, w) and i is the imaginary unit. Please refer to the Appendix for full details of the imaginary and real parts of $\mathcal{F}_{\text{DFT}}(\mathbf{I}_i)(u, v)$.

Step-2: Low-frequency components extraction. To only retain the low frequency components of the image $\mathbf{I}_i$, we apply an ideal low-pass filter $\mathcal{F}_{\text{LPF}}$ on the frequency spectrum $\mathcal{F}_{\text{DFT}}(\mathbf{I}_i)$. The ideal low-pass filter is defined as:

$$\mathcal{F}_{\text{LPF}}(u, v) = \begin{cases} 1, & \sqrt{((u - u_c)^2 + (v - v_c)^2)} \leq r, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Where v_c and u_c are the center coordinates of the frequency spectrum. r is the bandwidth of the circular ideal low-pass filter to control how many high-frequency components will be filtered out from the spectrum, and we have $r \in [0, \min(\frac{H}{2}, \frac{W}{2})]$. The extraction process is represented as $\mathcal{F}_{\text{LPF}}(u, v) \otimes \mathcal{F}_{\text{DFT}}(\mathbf{I}_i)(u, v)$, and $\otimes$ is the element-wise multiplication.

Step-3: Reconstruction target generation. We then apply the inverse Discrete Fourier Transform (IDFT) $\mathcal{F}_{\text{IDFT}}$ on the filtered spectrum to generate the RGB image as the final reconstruction target:

$$\mathbf{Y} = \mathcal{F}_{\text{IDFT}}(\mathcal{F}_{\text{LPF}}(u, v) \otimes \mathcal{F}_{\text{DFT}}(\mathbf{I}_i)(u, v)) \tag{4}$$

Both $\mathcal{F}_{\text{DFT}}$ and $\mathcal{F}_{\text{IDFT}}$ can be computed efficiently with Fast Fourier Transform (Brigham & Morrow, 1967). The computation cost of the above three steps is negligible thanks to the highly optimized implementation in PyTorch (Paszke et al., 2019).

To verify if our method successfully de-emphasizes the reconstruction of high-frequency components, Figure 5 presents a frequency analysis across 50,000 images from the validation set of ImageNet-1K, using $r = 40$. Compared to the vanilla MAE, our method produces obviously lower reconstruction PSNR at high-frequency intervals and slightly higher PSNR at low-frequency intervals, justifying the effectiveness of our method.

4.2 More Conservative Image Augmentation

Based on the analysis in Section 3, we would like to retain more foreground information in the input patches to our model. As a high masking ratio is crucial for MIM to learn effective representations (He et al., 2022),

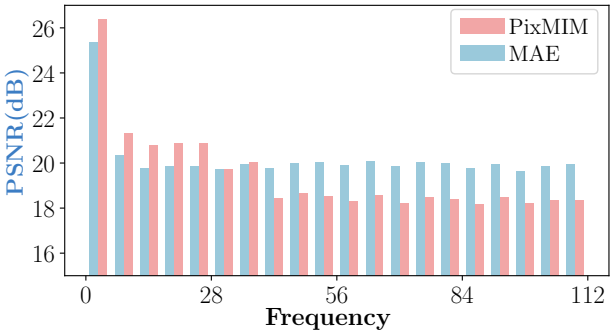

Figure 5: **Frequency analysis of MAE and PixMIM.** The PSNR of the reconstructed image for various frequency intervals (similar to Figure 1), averaged across 50,000 images from ImageNet-1K's validation set. PixMIM shifts the model's focus toward low-frequency components.

the most straightforward strategy is to keep the corruption $\mathcal{M}(\cdot)$ unchanged but make the augmentation function $\mathcal{A}(\cdot)$ more conservative.

We extend our quantitative analysis of object coverage to get Figure 6, which compares RRC with two less aggressive image augmentation operations. Simple Resized Crop (SRC) is the augmentation technique used in AlexNet (Krizhevsky et al., 2012), which resizes the image by matching the smaller edge to the pre-defined training resolution (*e.g.*, 224), then applies a reflect padding of 4 pixels on both sides, and finally randomly crops a square region of the specified training resolution to get the augmented image. Center Crop (CC) always takes the fixed-size crop from the center of the image. The results show that the SRC has much higher $\mathcal{J}(\mathcal{F})$ than RRC and CC. When the masking strategy of MAE is applied, SRC produces a $\mathcal{J}(\mathcal{F})$ of 22.1%, which is very close to the upper bound (*i.e.*, 25%). Therefore, we simply adopt the SRC as the augmentation function $\mathcal{A}(\cdot)$ and take the off-the-shelf implementation from (Touvron et al., 2022).

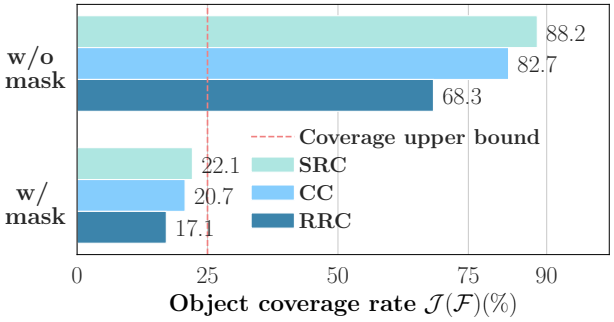

Figure 6: **Object coverage analysis.** SRC retains a higher proportion than RRC and CC, even under the aggressive masking strategy of MAE. Note that the upper bound of the object coverage is 25% when the masking strategy of MAE is applied. (RRC: random resized crop, SRC: simple resized crop, CC: center crop)

Note that when there is no image masking, the SRC raises the $\mathcal{J}(\mathcal{F})$ from the 68.3% of RRC to 88.2%, indicating that it offers less diversity than RRC, which accounts for the performance degeneration in supervised image classification observed by DeiT III (Touvron et al., 2022). But unlike supervised learning, the aggressive image masking in MIM already provides sufficient randomness, and the use of SRC will not hurt the diversity as in supervised learning.

4.3 Plug into existing MIM Methods

Unlike recent approaches such as CAE (Chen et al., 2022), MILAN (Hou et al., 2022), or BEiTv2 (Peng et al., 2022a), our method is lightweight and straightforward. It can easily be plugged into most existing pixel-based MIM frameworks. To demonstrate its effectiveness and versatility, we apply our method to MAE (He et al., 2022), MFF (Liu et al., 2023), ConvMAE (Gao et al., 2022), and LSMAE (Hu et al., 2022) to obtain PixMIM$_{\texttt{MAE}}$, PixMIM$_{\texttt{MFF}}$, PixMIM$_{\texttt{ConvMAE}}$, and PixMIM$_{\texttt{LSMAE}}$ respectively. The experimental results are presented in the next section.

5 Experiments

In Section 5.1, we describe the experimental settings for pre-training and evaluation. Then in Section 5.2, we apply our method to four MIM baselines (*i.e.*, MAE (He et al., 2022), MFF (Liu et al., 2023), ConvMAE (Gao et al., 2022), and LSMAE (Hu et al., 2022)), compare the results with the state of the arts, and discuss the sensitivity of the ImageNet fine-tuning protocol. To complement the ImageNet fine-tuning protocol, Section 5.3 demonstrates additional analyses by checking the robustness of pre-trained models with out-of-distribution (OOD) ImageNet variants and conducting a shape bias analysis. Finally, Section 5.4 provides comprehensive ablation studies for our method.

Evaluation Protocol→			ImageNet		COCO†		ADE20K
Method	Target	Epoch	ft(%)	lin(%)	AP^{box}	AP^{mask}	mIoU
Supervised learning							
DeiT III (Touvron et al., 2022)	-	800	83.8	-	-	-	49.3
Masked Image Modeling w/ pre-trained target generator							
BEiT (Bao et al., 2021)	DALLE	800	83.2	56.7	-	-	45.6
CAE (Chen et al., 2022)	DALLE	800	83.8	68.6	49.8	43.9	49.7
MILAN (Hou et al., 2022)	CLIP-B	400	85.4	78.9	52.6	45.5	52.7
BEiT-v2 (Peng et al., 2022a)	VQ-KD	1600	85.5	80.1	-	-	53.1
MaskDistill (Peng et al., 2022b)	CLIP-B	800	85.5	-	-	-	54.3
Masked Image Modeling w/o pre-trained target generator							
MaskFeat (Wei et al., 2022)	HOG	1600	84.0	62.3	52.3	46.4	48.3
SemMAE (Li et al., 2022a)	RGB	800	83.4	65.0	-	-	46.3
SimMIM (Xie et al., 2022)	RGB	800	83.8	56.7	-	-	-
MAE* (He et al., 2022)	RGB	800	83.3	65.6	51.3	45.7	46.1
PixMIM$_{\texttt{MAE}}$	RGB	800	83.5 (+0.2)	67.2 (+1.6)	51.7 (+0.4)	46.1 (+0.4)	47.3 (+1.2)
MFF (Liu et al., 2023)	RGB	800	83.6	67.0	51.8	46.1	47.9
PixMIM$_{\text{MFF}}$	RGB	800	83.5 (-0.1)	68.2 (+1.2)	52.3 (+0.5)	46.7 (+0.6)	48.6 (+0.7)
ConvMAE* (Gao et al., 2022)	RGB	800	84.6	68.4	52.0	46.3	50.2
PixMIM$_{\texttt{ConvMAE}}$	RGB	800	85.0 (+0.4)	70.5 (+2.1)	53.1 (+1.1)	47.0 (+0.7)	51.3 (+1.1)
LSMAE* (Hu et al., 2022)	RGB	800	83.2	63.7	51.0	45.4	48.5
PixMIM$_{\texttt{LSMAE}}$	RGB	800	83.6 (+0.4)	66.7 (+3.0)	52.1 (+1.1)	46.3 (+0.9)	50.1 (+1.6)

Table 2: **Performance comparison of MIM methods on various downstream tasks.** We report the results with fine-tuning (ft) and linear probing (lin) experiments on ImageNet-1K, objection detection on COCO, and semantic segmentation on ADE20K. The backbone of all experiments is ViT-B (Dosovitskiy et al., 2021). $*$: numbers are reported by running the official code release. †: As there is no uniform number of fine-tuning epochs for MAE, MFF, ConvMAE, and LSMAE for object detection, we fine-tuned PixMIM using the same number of epochs as each respective base method.

5.1 Experiment Settings

We evaluate our methods and validate our design components with extensive experiments over image classification on ImageNet-1K (Deng et al., 2009), object detection on COCO (Lin et al., 2014), and semantic segmentation on ADE20K (Zhou et al., 2018). Unless otherwise specified, we report the performance with ViT-B (Dosovitskiy et al., 2021).

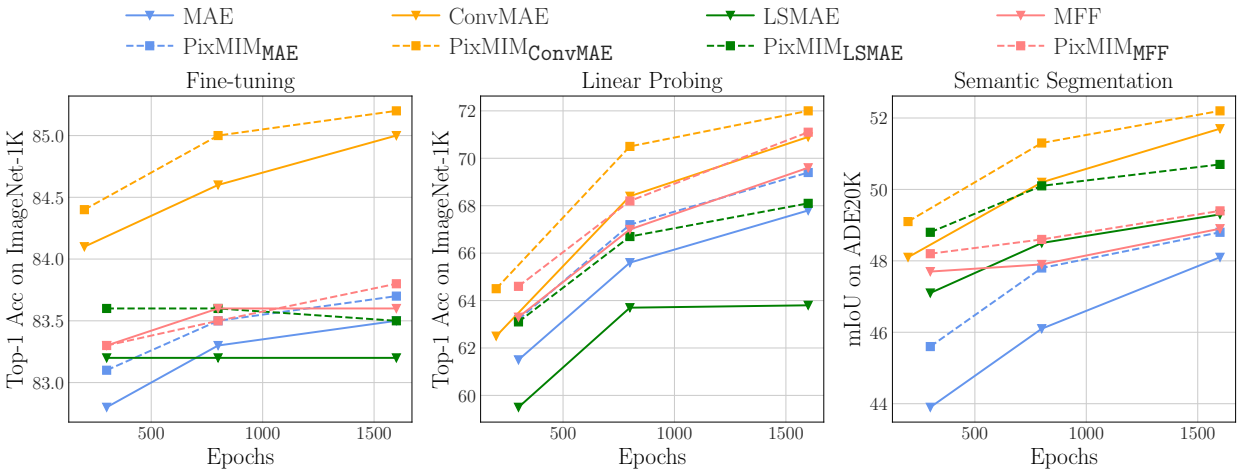

Figure 7: **Performance vs. epoch plots.** With different training epochs, PixMIM consistently brings significant gains to the baseline MIM approaches across various evaluation protocols.

ImageNet-1K (Deng et al., 2009) ImageNet-1K consists of 1.3M images of 1k categories and is split into the training and validation sets. When applying our methods to MAE (He et al., 2022), MFF (Liu et al., 2023), ConvMAE (Gao et al., 2022), and LSMAE (Hu et al., 2022), we strictly follow their original pre-training and evaluation settings on ImageNet-1K to guarantee the fairness of experiments, including the pre-training schedule, network architecture, learning rate setup, and fine-tuning protocols, etc. The only exception is that we increase the batch size of ConvMAE from 1024 to 4096 to accelerate the pre-training, while this change does not affect the performance according to our observations. We provide complete implementation details in the Appendix.

ADE20K (Zhou et al., 2018) For the semantic segmentation experiments on ADE20K, we follow the basic off-the-shelf settings from MAE (He et al., 2022). A UperNet (Xiao et al., 2018) is fine-tuned for 160k iterations with a batch size of 16. In addition, we also turn on the relative position bias and initialize them with zero. We report the Mean Intersection over Union (mIoU) results averaged over two runs for a robust comparison. The full details can be found in the Appendix.

COCO (Lin et al., 2014) For object detection experiments on COCO, we adopt the Mask R-CNN approach (He et al., 2017) that produces bounding boxes and instance masks simultaneously, with the ViT as the backbone. Similar to MAE, we employ the box and mask AP as the metrics. For MAE and LSMAE, we use the official implementation of ViTDet (Li et al., 2022c). For ConvMAE, we use its released official repository. More detailed settings can be found in the Appendix.

Ablation studies All ablation studies are based on the MAE settings. Following the common practice of previous MIM works (Liu et al., 2022a; Chen et al., 2022; Gao et al., 2022), we pre-train all model variants on ImageNet-1K for 300 epochs and comprehensively compare their performance on linear probing, fine-tuning, and semantic segmentation. All other settings are the same as those discussed above.

5.2 Main Results

In Table 2, we show the results of applying our simple method to MAE (He et al., 2022), MFF Liu et al. (2023), ConvMAE (Gao et al., 2022), and LSMAE (Hu et al., 2022), and compare these results with the state-of-the-art MIM approaches. Without extra computational cost, we consistently improve the original MAE, ConvMAE, MFF, and LSMAE across all downstream tasks. The margins on linear probing, object detection, and semantic segmentation are remarkable. Specifically, PixMIM$_{\texttt{LSMAE}}$ significantly improves the original LSMAE on linear probing and semantic segmentation by 3.0% and 1.6%, respectively. To further demonstrate the effectiveness of our method across various pre-training schedules, we plot the *performance*

Method	Target	Bakcbone	ft	lin	seg
LSMAE (Hu et al., 2022)	RGB	ViT-B	83.2	63.7	48.5
MAE (He et al., 2022)	RGB	ViT-B	83.3	65.6	46.1
PixMIM$_{\texttt{MAE}}$	RGB	ViT-B	83.5	67.2	47.8
PixMIM$_{\texttt{MFF}}$	RGB	ViT-B	83.5	68.2	48.6
PixMIM$_{\texttt{LSMAE}}$	RGB	ViT-B	83.6	66.7	50.1
SimMIM (Xie et al., 2022)	RGB	ViT-B	83.8	56.7	-
MaskFeat (Wei et al., 2022)	HOG	ViT-B	84.0	62.3	48.3

Table 3: **Investigating the ImageNet fine-tuning protocol.** Six MIM approaches are *sorted* based on their ImageNet fine-tuning (ft) performance. The fine-tuning result alone hardly distinguishes different approaches with the same backbone and not using an extra pre-trained model for generating training targets, and it does not necessarily correlate with other evaluation protocols. Best viewed in color.

Method	IN-C↓	IN-A	IN-R	IN-S
LSMAE	48.8	34.2	50.3	36.2
PixMIM$_{\texttt{LSMAE}}$	48.0 (-0.8)	36.1 (+1.9)	50.8 (+0.5)	37.1 (+0.9)
MAE	51.7	35.9	48.3	34.5
PixMIM$_{\texttt{MAE}}$	49.9 (-1.8)	37.1 (+1.2)	49.6 (+1.3)	35.9 (+1.4)
ConvMAE	45.5	50.8	54.6	41.1
PixMIM$_{\texttt{ConvMAE}}$	45.3 (-0.2)	52.5 (+1.7)	55.3 (+0.7)	41.8 (+0.7)
MFF	49.0	37.2	51.0	36.8
PixMIM$_{\texttt{MFF}}$	48.5 (-0.5)	40.1 (+2.9)	51.6 (+0.6)	37.8 (+1.0)

Table 4: **Robustness evaluation on ImageNet variants.** To complement the less sensitive ImageNet fine-tuning protocol, we further evaluate the fine-tuned models from the main table on four ImageNet variants. Results are reported in top-1 accuracy, except for IN-C which uses the mean corruption error.

vs. epoch curves in Figure 7. The curves of PixMIM$_{\texttt{MAE}}$, PixMIM$_{\texttt{ConvMAE}}$, PixMIM$_{\texttt{MFF}}$, and PixMIM$_{\texttt{LSMAE}}$ consistently remain above the corresponding base methods by clear gaps. All these results demonstrate the universality and scalability of our methods.

Methods with pre-trained target generator. Although the methods with a powerful pre-trained target generator (Hou et al., 2022; Peng et al., 2022a) achieve the best results in Table 2, they rely on extra pre-training data and bring significant computational overhead to MIM when generating targets dynamically. In contrast, our improvements come with negligible cost and take a step towards closing the gap between pixel-based approaches and those relying on pre-trained target generators.

Remarks on the ImageNet fine-tuning protocol. According to Table 2, the improvements brought by our method on the ImageNet fine-tuning protocol are less obvious than those on the other three protocols. Table 3 investigates the correlation between the evaluation protocols by *sorting* six MIM approaches based on the ImageNet-finetuning performance, and we have the following observations:

- With the same network backbone and not using extra pre-trained models for generating training targets, the ImageNet fine-tuning performances of various methods always show marginal gaps.
- Better result on ImageNet fine-tuning does not necessarily mean better performance on linear probing or semantic segmentation. This is also shown by the curves of LSMAE in Figure 7.

Hence, we argue that ImageNet fine-tuning is *not a sensitive metric*, and we should include more protocols for comprehensively evaluating the representation quality. A potential explanation provided by CAE (Chen et al., 2022) is that the pre-training and fine-tuning data follow the same distribution and can narrow the gap among different methods. We provide additional analyses in the next subsection to complement the ImageNet fine-tuning protocol.

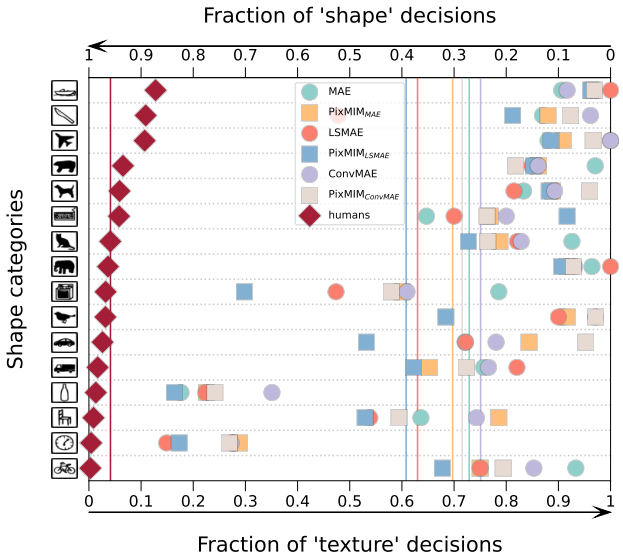

Figure 8: **Shape bias analysis.** PixMIM consistently improves the shape bias of the baselines. Each vertical line is the *weighted average* of all 16 categories.

5.3 Additional Analyses

Two additional experiments are presented to complement the less sensitive ImageNet fine-tuning protocol and further validate the effectiveness of our method.

Robustness checking. we compare the pre-trained models on four out-of-distribution ImageNet variants: ImageNet-Corruption (Hendrycks & Dietterich, 2019), ImageNet-Adversarial (Hendrycks et al., 2021b), ImageNet-Rendition (Hendrycks et al., 2021a), and ImageNet-Sketch (Wang et al., 2019). These datasets introduce various domain shifts to the original ImageNet-1K and are widely used to assess a model's robustness and generalization ability. Table 4 shows that PixMIM$_{\texttt{MAE}}$, PixMIM$_{\texttt{ConvMAE}}$, PixMIM$_{\texttt{LSMAE}}$, and PixMIM$_{\texttt{MFF}}$ consistently outperform their baselines, and the margins of improvement are much more pronounced than those on the validation set of Imagenet-1K. The better robustness against domain shifts strengthens the value of our simple yet effective method.

Shape bias analysis. We take the off-the-shelf shape bias toolbox (Geirhos et al., 2021) to analyze our pre-trained models. Shape bias measures how much the model relies on shapes to extract the semantic representation of the image, quantified as the fraction of correct decisions based on object shape. Figure 8 shows that PixMIM$_{\texttt{MAE}}$, PixMIM$_{\texttt{ConvMAE}}$, and PixMIM$_{\texttt{LSMAE}}$ improve the shape bias of their baselines, confirming that our methods prevent the model from being excessively texture-biased by filtering out the high-frequency components of the target image. The colored lines denote the *weighted average* of shape bias across different categories for different methods.

5.4 Ablation Studies

We further conduct ablation studies for our key design components: the filtering of the high-frequency components of the target image and the use of the Simple Resized Crop.

The bandwidth of the low-pass filter. Table 5a investigates how varying the bandwidth r influences the vanilla MAE. All model variants in the table are trained for 300 epochs following the training recipes of MAE. The optimal bandwidth is 40, and it improves the baseline significantly (*i.e.*, +1.2% on linear probing and +1.7% on semantic segmentation). A narrow bandwidth could discard important information about the

r	ft	lin	seg
baseline	82.8	61.5	43.9
30	82.9	62.5	44.6
35	**82.9**	62.6	44.8
40	82.8	**62.7**	**45.3**
45	82.7	62.2	45.1
50	82.8	61.8	44.8

(a) The bandwidth(r) of the low-pass filter. $r = 40$ yields the optimal result.

aug	ft	lin	seg
RRC	82.8	61.5	43.9
SRC	**82.9**	**62.5**	**44.3**
CC	82.8	62.2	44.1
BG	82.5	59.5	43.4
Resize	82.6	58.2	42.1

(b) The data augmentation. BG: background-focused random cropping, see text for details. Resize: resize the image, no random cropping.

LF	SRC	ft	lin	seg
-	-	82.8	61.5	43.9
✓	-	82.8	62.7	45.3
-	✓	82.9	62.5	44.3
✓	✓	**83.2**	**63.3**	**46.2**

(c) The combination of the low-frequency target (LF) and simple resized crop (SRC).

Table 5: **The ablation studies** with ViT-B/16 pre-trained on ImageNet-1K for 300 epochs. We report fine-tuning (ft), linear probing (lin), and semantic segmentation (seg) results. Our final settings are marked in gray.

Method	RRC	SRC	ft	lin	seg
MAE	✓		82.1	48.4	45.8
MAE		✓	82.4 (+0.3)	48.9 (+0.5)	46.8 (+1.0)

Table 6: Extension of Table 5b, pre-trained on the non-object-centric dataset Place365 (López-Cifuentes et al., 2020). The lacking foreground issue is not unique to MIM training on single-object datasets like ImageNet.

image (*e.g.* edges of objects), leading to a performance drop. In comparison, a too-large bandwidth fails to remove unessential textures effectively.

Replace RRC with SRC. Table 5b compares different data augmentations. The simple resized crop (SRC) brings non-trivial improvement to the original random resized crop (RRC) used by MAE on both linear probing and semantic segmentation. However, recall in DeiT III (Touvron et al., 2022), replacing RRC with SRC degrades the performance as it decreases the cropped image's diversity and impairs the model's generalization ability. The opposite results we obtain in MIM here suggest that the RRC could have led to the severe issue of missing foreground, which is further confirmed by the fact that even the simple center crop can outperform RRC in linear probing and semantic segmentation.

To better support our analysis on the input patches of MAE, we conduct reverse engineering of SRC, which crops mostly the background region instead of the foreground (*i.e.*, the BG entry in Table 5b). Please check the Appendix for implementation details and visualizations. The results demonstrate that the absence of foreground information can significantly impair the representation quality, further confirming our analysis regarding the input patches.

Table 5c further verifies that the gains brought by the two components of PixMIM effectively accumulate over all three evaluation protocols. We also extend Table 5b to a non-object-centric dataset, Place365 (López-Cifuentes et al., 2020). Table 6 shows that SRC still brings non-negligible improvements over RRC when pre-trained on scene-scale images, suggesting that the lacking foreground issue in pixel-based MIM is universal and not specific to object-centric datasets (*e.g.*, ImageNet).

6 Limitation and Future Works

Currently, our experiments are based on ViT-B (Dosovitskiy et al., 2021), which is also the common practice by some other works (Xie et al., 2022; Li et al., 2022a). Some studies, like (He et al., 2022), suggest that after extending experiments to larger models, *e.g.*, ViT-L or ViT-H, the same method can still get equivalent gains, but evaluating the scalability of our methods on larger models is also expected. In addition, the bandwidth r is designed as a hyper-parameter and may vary across different datasets or input resolutions. So a self-

adaptive bandwidth is also expected. Finally, self-supervised pre-training has been criticized for consuming many computational resources. Even though our method brings negligible computation overhead, making the entire pre-training pipeline more efficient should be one of the directions for future research.

7 Conclusion

In this paper, we first provide an empirical analysis of the milestone algorithm, MAE, from the perspective of input patches and reconstruction targets, identifying the potential bottlenecks of existing pixel-based MIM approaches. Based on the analysis, we propose a simple yet effective method, PixMIM, without introducing extra computation overhead or complicating the pre-training pipeline. When applied to three representative pixel-based MIM approaches, PixMIM brings consistent performance boosts across various downstream tasks and improves the model's robustness, demonstrating its effectiveness and universality.

References

Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021. 1, 2, 3, 4, 8, 18, 20, 21

E. O. Brigham and R. E. Morrow. The fast fourier transform. *IEEE Spectrum*, 1967. 6

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *ArXiv*, abs/2005.14165, 2020. 3

Jiacheng Chen, Hexiang Hu, Hao Wu, Yuning Jiang, and Changhu Wang. Learning the best pooling strategy for visual semantic embedding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15789–15798, 2021. 3

Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *Proceedings of the 37th International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2020a. 17, 18

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)*, 2020b. 3

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*, abs/2202.03026, 2022. 8, 9, 10

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*, 2021. 3

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=r1xMH1BtvB. 18, 20, 21

Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020. 18

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 2, 8, 9

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

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021. 8, 12

Christoph Feichtenhofer, Haoqi Fan, Bo Xiong, Ross B. Girshick, and Kaiming He. A large-scale study on unsupervised spatiotemporal representation learning. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

Peng Gao, Teli Ma, Hongsheng Li, Ziyi Lin, Jifeng Dai, and Yu Qiao. MCMAE: Masked convolution meets masked autoencoders. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 2, 4, 8, 9

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *ArXiv*, abs/1811.12231, 2019. 2, 4

Robert Geirhos, Kantharaju Narayanappa, Benjamin Mitzkus, Tizian Thieringer, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Partial success in closing the gap between human and machine vision. In *NeurIPS*, 2021. 2, 3, 4, 11

Priya Goyal, Piotr Dollár, Ross B. Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *ArXiv*, abs/1706.02677, 2017. 17, 18

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. In *NeurIPS*, 2020. 3

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision (ICCV)*, 2017. 9

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3

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, 6, 8, 9, 10, 12, 17

Dan Hendrycks and Thomas G. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *ArXiv*, abs/1903.12261, 2019. 11

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Lixuan Zhu, Samyak Parajuli, Mike Guo, Dawn Xiaodong Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021a. 11

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Xiaodong Song. Natural adversarial examples. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021b. 11

Zejiang Hou, Fei Sun, Yen-Kuang Chen, Yuan Xie, and S. Y. Kung. Milan: Masked image pretraining on language assisted representation. *ArXiv*, abs/2208.06049, 2022. 1, 3, 8, 10

Kai Hu, Jie Shao, Yuan Liu, Bhiksha Raj, Marios Savvides, and Zhiqiang Shen. Contrast and order representations for video self-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 3

Ronghang Hu, Shoubhik Debnath, Saining Xie, and Xinlei Chen. Exploring long-sequence masked autoencoders. *ArXiv*, abs/2210.07224, 2022. 2, 8, 9, 10, 17

Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q. Weinberger. Deep networks with stochastic depth. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (eds.), *Computer Vision – ECCV 2016*. Springer International Publishing, 2016. 18, 20, 21

Zhicheng Huang, Xiaojie Jin, Cheng Lu, Qibin Hou, Mingg-Ming Cheng, Dongmei Fu, Xiaohui Shen, and Jiashi Feng. Contrastive masked autoencoders are stronger vision learners. *ArXiv*, abs/2207.13532, 2022. 1

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning (ICML)*, 2021. 3

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 2012. 2, 7

Gang Li, Heliang Zheng, Daqing Liu, Chaoyue Wang, Bing Su, and Changwen Zheng. SemMAE: Semantic-guided masking for learning masked autoencoders. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022a. 8, 12

Junnan Li, Ramprasaath R. Selvaraju, Akhilesh Deepak Gotmare, Shafiq R. Joty, Caiming Xiong, and Steven C. H. Hoi. Align before fuse: Vision and language representation learning with momentum distillation. In *Neural Information Processing Systems (NeurIPS)*, 2021. 3

Xuhong Li, Haoyi Xiong, Yi Liu, Dingfu Zhou, Zeyu Chen, Yaqing Wang, and Dejing Dou. Distilling ensemble of explanations for weakly-supervised pre-training of image segmentation models. *ArXiv*, abs/2207.03335, 2022b. 5, 19

Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. In *ECCV*, 2022c. 9

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 *ECCV*, 2014. 2, 8, 9

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*, 2022a. 9

Yuan Liu, Jiacheng Chen, and Hao Wu. Moquad: Motion-focused quadruple construction for video contrastive learning. *arXiv preprint arXiv:2212.10870*, 2022b. 3

Yuan Liu, Songyang Zhang, Jiacheng Chen, Zhaohui Yu, Kai Chen, and Dahua Lin. Improving pixel-based mim by reducing wasted modeling capability. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5361–5372, October 2023. 2, 8, 9

Alejandro López-Cifuentes, Marcos Escudero-Viñolo, Jesús Bescós, and Álvaro García-Martín. Semantic-aware scene recognition. *Pattern Recognition*, 102:107256, 2020. 12

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Skq89Scxx. 17, 18

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7. 17, 18, 20, 21

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 6, 19

Zhiliang Peng, Li Dong, Hangbo Bao, Qixiang Ye, and Furu Wei. Beit v2: Masked image modeling with vector-quantized visual tokenizers. *ArXiv*, abs/2208.06366, 2022a. 1, 3, 8, 10

Zhiliang Peng, Li Dong, Hangbo Bao, Qixiang Ye, and Furu Wei. A unified view of masked image modeling. *ArXiv*, abs/2210.10615, 2022b. 8

Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, H. Wang, Serge J. Belongie, and Yin Cui. Spatiotemporal contrastive video representation learning. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

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

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 3, 4

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *ArXiv*, abs/2102.12092, 2021. 2, 3, 4

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 2

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 18

Hugo Touvron, Matthieu Cord, and Hervé Jégou. Deit iii: Revenge of the vit. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (eds.), *ECCV*, 2022. 2, 5, 7, 8, 12

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *International Conference on Machine Learning (ICML)*, 2008. 4

Haohan Wang, Songwei Ge, Eric P. Xing, and Zachary Chase Lipton. Learning robust global representations by penalizing local predictive power. In *NeurIPS*, 2019. 11

Chen Wei, Haoqi Fan, Saining Xie, Chaoxia Wu, Alan Loddon Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 4, 8, 10

Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 18

Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, 2018. 9

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 *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 4, 8, 10, 12

Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv: Computer Vision and Pattern Recognition*, 2017. 19

Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Seong Joon Oh, Youngjoon Yoo, and Junsuk Choe. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 18

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations (ICLR)*, 2018. 18

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision (IJCV)*, 2018. 2, 8, 9

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. 1

A Discrete Fourier Transform

$\mathcal{F}_{\mathrm{DFT}}(\mathbf{I}_i)(u,v)$ in Section 4.1 of the main paper can be decomposed into real and imaginary parts:

$$\mathcal{F}_{\mathrm{DFT}}(\mathbf{I}_i)(u,v) = \mathcal{R}(\mathrm{I}_i)(u,v) + \mathcal{I}(\mathrm{I}_i)(u,v)i \tag{5}$$

Both $\mathcal{R}(\mathrm{I}_i)(u,v)$ and $\mathcal{I}(\mathrm{I}_i)(u,v)$ are real numbers, and i is the imaginary unit. The amplitude and phase of $\mathcal{F}_{\mathrm{DFT}}(\mathbf{I}_i)(u,v)$ can be obtained using the following formulas:

$$\mathcal{A}(\mathrm{I}_i)(u,v) = (\mathcal{R}(\mathrm{I}_i)(u,v)^2 + \mathcal{I}(\mathrm{I}_i)(u,v)^2)^{\frac{1}{2}} \tag{6}$$

$$\mathcal{P}(\mathrm{I}_i)(u,v) = arctan(\frac{\mathcal{R}(\mathrm{I}_i)(u,v)}{\mathcal{I}(\mathrm{I}_i)(u,v)}) \tag{7}$$

B Full Implementation Details

B.1 Pre-training

We show the detailed implementation settings for the pre-training of PixMIM$_{\texttt{MAE}}$, PixMIM$_{\texttt{ConvMAE}}$, PixMIM$_{\texttt{MFF}}$ and PixMIM$_{\texttt{LSMAE}}$ in the following tables. While for the result for LSMAE (Hu et al., 2022) in Table 2 of the main paper, except for that we decrease the batch size from 4096 to 2048 due to the limited computational resources, other settings are kept the same.

config	value
optimizer	AdamW (Loshchilov & Hutter, 2019)
base learning rate	1.5e-4
weight decay	0.05
optimizer momentum	β_1, β_2=0.9, 0.95 (Chen et al., 2020a)
batch size	4096
learning rate schedule	cosine decay (Loshchilov & Hutter, 2017)
warmup epochs (Goyal et al., 2017)	40
augmentation	SimpleResizedCrop

Table 7: Pre-training setting of PixMIM$_{\texttt{MAE}}$, PixMIM$_{\texttt{MFF}}$, and PixMIM$_{\texttt{ConvMAE}}$

B.2 ImageNet Fine-tuning

The implementation details for the fine-tuning of PixMIM$_{\texttt{MAE}}$, PixMIM$_{\texttt{ConvMAE}}$, and PixMIM$_{\texttt{LSMAE}}$ are shown in Table 9, which strictly follow that of MAE (He et al., 2022).

B.3 ImageNet Linear Probing

The implementation details for PixMIM$_{\texttt{MAE}}$ and PixMIM$_{\texttt{LSMAE}}$ are shown in Table 10, which also follow that of MAE (He et al., 2022). For PixMIM$_{\texttt{ConvMAE}}$, we decrease the batch size from 16384 to 4096, following the official setting.

config	value
optimizer	AdamW (Loshchilov & Hutter, 2019)
base learning rate	1.5e-4
weight decay	0.05
optimizer momentum	β_1, β_2=0.9, 0.95 (Chen et al., 2020a)
batch size	2048
learning rate schedule	cosine decay (Loshchilov & Hutter, 2017)
warmup epochs (Goyal et al., 2017)	40
augmentation	SimpleResizedCrop

Table 8: Pre-training setting of PixMIM$_{\texttt{LSMAE}}$

config	value
optimizer	AdamW (Loshchilov & Hutter, 2019)
base learning rate	1e-3
weight decay	0.05
optimizer momentum	β_1, β_2=0.9, 0.999
layer-wise lr decay (Clark et al., 2020; Bao et al., 2021)	0.75
batch size	1024
learning rate schedule	cosine decay
warmup epochs	5
training epochs	100
augmentation	RandAug (9, 0.5) (Cubuk et al., 2020)
label smoothing (Szegedy et al., 2016)	0.1
mixup (Zhang et al., 2018)	0.8
cutmix (Yun et al., 2019)	1.0
drop path (Huang et al., 2016)	0.1

Table 9: End-to-end fine-tuning setting of PixMIM$_{\texttt{MAE}}$, PixMIM$_{\texttt{ConvMAE}}$, PixMIM$_{\texttt{MFF}}$, and PixMIM$_{\texttt{LSMAE}}$

B.4 ADE20K Semantic Segmentation

The implementation details for PixMIM$_{\texttt{MAE}}$ and PixMIM$_{\texttt{LSMAE}}$ are shown in Table 12, and those for PixMIM$_{\texttt{ConvMAE}}$ are in Table 13.

B.5 COCO Object Detection

For PixMIM$_{\texttt{MAE}}$, we follow the official release in Detectron2 (Wu et al., 2019), which fine-tunes the pre-trained model end to end for 100 epochs. While for PixMIM$_{\texttt{LSMAE}}$ and PixMIM$_{\texttt{ConvMAE}}$, we follow the official pre-training epochs, which are 50 and 25 epochs, respectively. Other key hyper-parameters are in Table 14 and Table 15.

C Additional Results

Pre-training for 1600 epochs. Table 17 provides the results of pre-training PixMIM for 1600 epochs and compares it against the corresponding base methods. PixMIM can still bring non-trivial improvements over base methods on various downstream tasks, which verifies the scalability of our method across pre-training epochs.

Pre-training with other model variants In this section, we provide addition results about ViT-S and ViT-L, and pre-train the models fro 1600 epochs. The detailed results are shown in Table 16, and PixMIM still bring consistent improvement over other model variants.

The 2× object detection protocol. We also provide the results with the 2× settings of object detection for PixMIM$_{\text{MAE}}$ in Table 18.

config	value
optimizer	LARS (You et al., 2017)
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 10: Linear probing setting of $\text{PixMIM}_{\texttt{MAE}}$, $\text{PixMIM}_{\texttt{MFF}}$, and $\text{PixMIM}_{\texttt{LSMAE}}$.

config	value
optimizer	LARS (You et al., 2017)
base learning rate	0.1
weight decay	0
optimizer momentum	0.9
batch size	4096
learning rate schedule	cosine decay
warmup epochs	10
training epochs	90
augmentation	RandomResizedCrop

Table 11: Linear probing setting of $\text{PixMIM}_{\texttt{ConvMAE}}$.

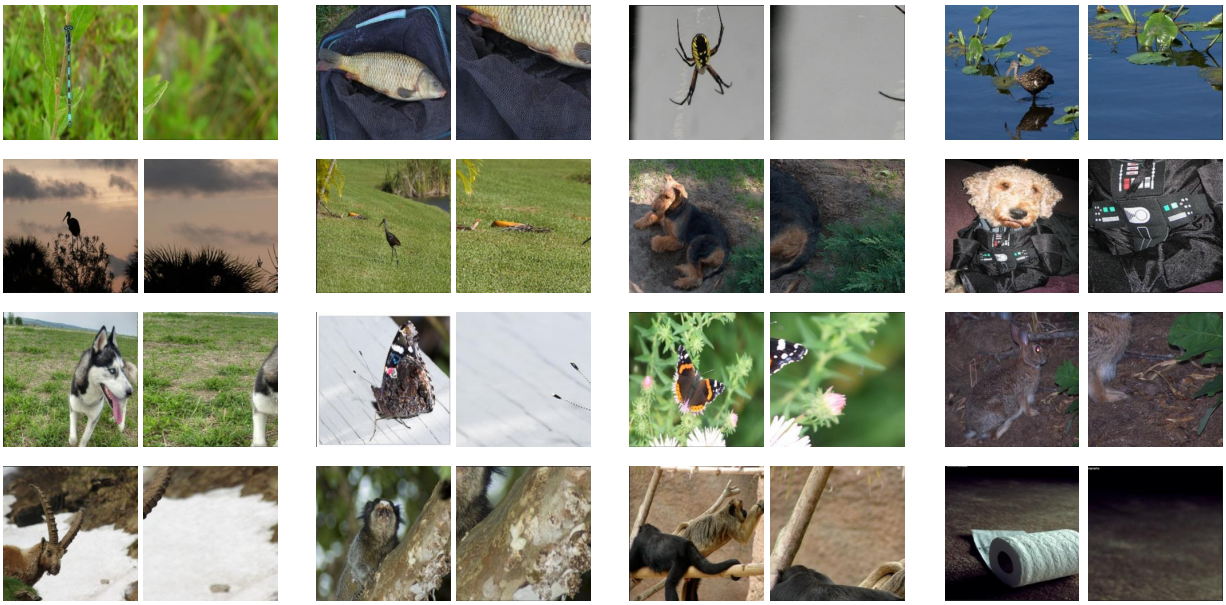

Figure 9: **Pre-train MAE with background cropped images.** Instead of cropping the foreground region in each image, we attempt to input the background into the model. The original image is depicted on the left of each pair, while the cropped background can be seen on the right.

Few-shot fine-tuning. Table 19 presents the results of few-shot fine-tuning, which uses 1% and 10% of ImageNet-1K training data to pre-train the model.

Pre-train MAE with background-focused image crops. We provide the implementation details for the background-focused random cropping in Table 5b of the main paper. To crop the background, we use the ImageNet-1K binary mask from PSSL (Li et al., 2022b) in the RRC (implementedy by PyTorch (Paszke et al., 2019)) to check whether the cropped image contains less than 20% of the object in the original image.

config	value
optimizer	AdamW (Loshchilov & Hutter, 2019)
base learning rate	2e-4
weight decay	0.05
optimizer momentum	β_1, β_2=0.9, 0.999
layer-wise lr decay (Clark et al., 2020; Bao et al., 2021)	0.75
batch size	16
learning rate schedule	cosine decay
warmup iters	1500
training iters	160000
drop path (Huang et al., 2016)	0.1

Table 12: End-to-end semantic segmentation setting of PixMIM$_{\texttt{MAE}}$, PixMIM$_{\texttt{MFF}}$, and PixMIM$_{\texttt{LSMAE}}$

config	value
optimizer	AdamW (Loshchilov & Hutter, 2019)
base learning rate	3e-4
weight decay	0.05
optimizer momentum	β_1, β_2=0.9, 0.999
layer-wise lr decay (Clark et al., 2020; Bao et al., 2021)	0.75
batch size	16
learning rate schedule	cosine decay
warmup iters	1500
training iters	160000
drop path (Huang et al., 2016)	0.1

Table 13: End-to-end semantic segmentation setting of PixMIM$_{\texttt{ConvMAE}}$

If the condition is satisfied, we return the cropped image. Otherwise, we crop another region of the image. We repeat the above procedure 50 times until the condition is satisfied; otherwise, we use the original RRC. Some visualizations of this augmentation strategy can be viewed in Figure 9.

config	value
optimizer	AdamW (Loshchilov & Hutter, 2019)
base learning rate	8e-5
weight decay	0.1
optimizer momentum	β_1, β_2=0.9, 0.999
layer-wise lr decay (Clark et al., 2020; Bao et al., 2021)	0.75
batch size	64
learning rate schedule	cosine decay
training epochs	100/50
drop path (Huang et al., 2016)	0.1

Table 14: End-to-end object detection setting of PixMIM$_{\texttt{MAE}}$, PixMIM$_{\texttt{MFF}}$, and PixMIM$_{\texttt{LSMAE}}$

config	value
optimizer	AdamW (Loshchilov & Hutter, 2019)
base learning rate	1e-4
weight decay	0.1
optimizer momentum	β_1, β_2=0.9, 0.999
layer-wise lr decay (Clark et al., 2020; Bao et al., 2021)	0.85
batch size	16
learning rate schedule	cosine decay
training epochs	25
drop path (Huang et al., 2016)	0.1

Table 15: End-to-end object detection setting of PixMIM$_{\texttt{ConvMAE}}$

Method	Epoch	Model	LIN	SEG	FT
MAE	1600	ViT-S	51.1	42.3	79.5
PixMIM$_{\texttt{MAE}}$	1600	ViT-S	53.2(+2.1)	44.1(+1.9)	80.4(0.9)
MAE	1600	ViT-L	76.0	53.6	85.9
PixMIM$_{\texttt{MAE}}$	1600	ViT-L	77.1(+1.1)	54.4(+0.8)	86.1(+0.2)

Table 16: Performance of applying PixMIM to other model variants

Method	Epoch	LIN	SEG	DET	FT
MAE	1600	67.8	48.1	51.4	83.5
PixMIM$_{\texttt{MAE}}$	1600	69.3(+1.5)	48.7(+0.6)	52.1(+0.7)	83.6
ConvMAE	1600	70.9	51.7	53.6	85.0
PixMIM$_{\texttt{ConvMAE}}$	1600	72.0(+1.1)	52.2(+0.5)	53.8(+0.2)	85.2
MFF	1600	69.6	48.9	52.3	83.6
PixMIM$_{\texttt{MFF}}$	1600	71.1(+1.5)	49.4(+0.5)	52.7(+0.4)	83.8
LSMAE	1600	63.8	49.3	51.6	83.2
PixMIM$_{\texttt{LSMAE}}$	1600	68.1(+4.3)	50.7(+1.4)	53.0(+1.4)	83.5

Table 17: Performance Comparison by pre-training models for 1600 epochs

Method	Epoch	AP^{box}	AP^{mask}
MAE	800	47.3	42.3
PixMIM$_{\text{MAE}}$	800	47.8 (+0.5)	42.8 (+0.5)
MAE	1600	48.6	43.5
PixMIM$_{\text{MAE}}$	1600	49.3 (+0.7)	44.0 (+0.5)

Table 18: Results of 2× settings for object detection

Method	Epoch	1%	10%
MAE	800	45.4	71.2
PixMIM$_{\mathbf{MAE}}$	800	47.9 (+2.5)	72.2 (+1.0)

Table 19: Few-shot fine-tuning with 1% and 10% of ImageNet-1K training data.