

# Improving Pixel-based MIM by Reducing Wasted Modeling Capability

Yuan Liu<sup>1</sup>, Songyang Zhang<sup>1,‡</sup>, Jiacheng Chen<sup>2</sup>, Zhaohui Yu<sup>3</sup>, Kai Chen<sup>1,‡</sup>  
Dahua Lin<sup>1,3</sup>

<sup>1</sup>Shanghai AI Laboratory    <sup>2</sup>Simon Fraser University    <sup>3</sup>The Chinese University of Hong Kong  
<sup>‡</sup> Corresponding author

## Abstract

There has been significant progress in Masked Image Modeling (MIM). Existing MIM methods can be broadly categorized into two groups based on the reconstruction target: pixel-based and tokenizer-based approaches. The former offers a simpler pipeline and lower computational cost, but it is known to be biased toward high-frequency details. In this paper, we provide a set of empirical studies to confirm this limitation of pixel-based MIM and propose a new method that explicitly utilizes low-level features from shallow layers to aid pixel reconstruction. By incorporating this design into our base method, MAE, we reduce the wasted modeling capability of pixel-based MIM, improving its convergence and achieving non-trivial improvements across various downstream tasks. To the best of our knowledge, we are the first to systematically investigate multi-level feature fusion for isotropic architectures like the standard Vision Transformer (ViT). Notably, when applied to a smaller model (e.g., ViT-S), our method yields significant performance gains, such as 1.2% on fine-tuning, 2.8% on linear probing, and 2.6% on semantic segmentation. Code and models are available in MMPretrain<sup>1</sup>.

## 1. Introduction

Self-supervised learning (SSL) has made remarkable progress in language and computer vision. Masked Image Modeling (MIM) is an effective framework in image SSL, which boasts a simple training pipeline, a few handcrafted data augmentations, and high performance across downstream tasks. In the pioneering work of BEiT [1], 40% of the input image is masked, and the model is trained to capture the semantics of the masked patches by reconstructing the DALL-E [37] output features. To simplify pre-training and reduce computational overhead, MAE [14] only feeds the visible tokens into the encoder and encourages the decoder to reconstruct the raw pixels of masked patches. More

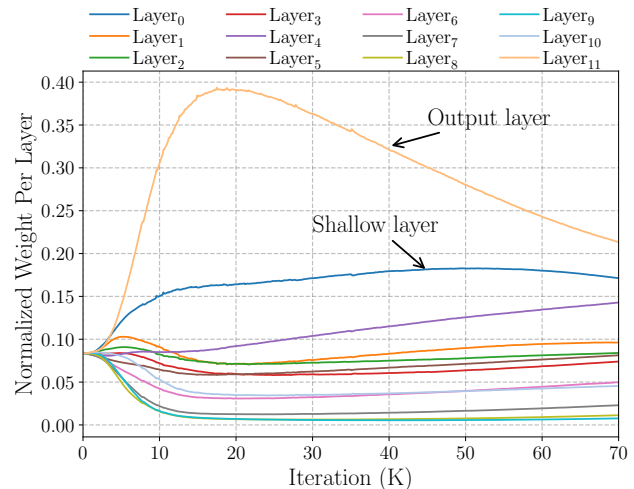


Figure 1: **Fusing shallow encoder layers for MAE.** During the training process, MAE increasingly relies on shallow layers for the pixel reconstruction task, demonstrating a bias toward low-level features.

recently, follow-up works have focused on adding auxiliary tasks or using large-scale pre-trained models to produce reconstruction targets. For instance, CMAE [23] explicitly adds a contrastive task and optimizes it in conjunction with the MIM task, while MILAN [20] and BEiT-v2 [34] employ multimodal pre-trained models such as CLIP [36] to generate the reconstruction features. Among the various MIM methods available, pixel-based approaches such as MAE [14] are particularly interesting because of their simple pre-training pipeline and minimal computational overhead. However, these methods are typically biased towards capturing high-frequency details due to their emphasis on reconstructing raw pixels [1, 30]. As a result, they waste a significant amount of modeling capability that could be better utilized to capture low-frequency semantics. Our objective is to reduce this waste of modeling capacity, aiming for an improved quality of learned representation for downstream visual tasks. Toward this goal, we design two pilot

<sup>1</sup><https://github.com/open-mmlab/mmpretrain>

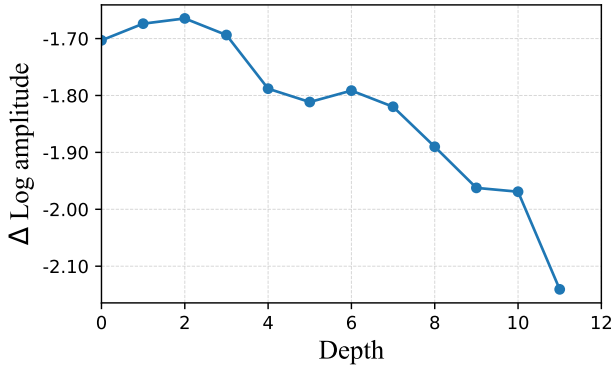


Figure 2: **Frequency analysis of MAE.** Higher log amplitude denotes more high-frequency information. Shallow layers contain more high-frequency information (or low-level information) than deep layers.

experiments based on the representative work of MAE [14] to uncover its neglected design aspects.

(1) **Fuse Shallow Layers:** Rather than solely using the output layer for pixel reconstruction, we implement a weight-average strategy to fuse the output layer with all previous layers. The weights assigned to each layer are normalized and dynamically updated during the pre-training process, with their absolute values indicating the significance of each layer for the reconstruction task. We track the changes in these weights and illustrate them in Figure 1. As depicted in the figure, the model increasingly relies on the features of shallow layers as training progresses.

(2) **Frequency Analysis:** To further understand the property of the representation learned with MAE, we analyze the frequency response of each layer’s feature. We adopt the tools proposed by [33] to transform the encoder features into the frequency domain and visualize the relative log amplitude of the transformed representation in Figure 2. Typically, a higher amplitude indicates that the feature produced by one layer contains *more high-frequency information*. We empirically find that the shallow layers contain significantly more high-frequency components than deep layers, which are mostly related to low-level details (*e.g.*, textures).

Based on our analysis of the pilot experiments, we can conclude that the pixel reconstruction task exhibits a bias towards low-level details. This is evident from the low linear probing accuracy, which highlights the constraints of the pixel reconstruction task. Namely, it requires features that are semantically distinct enough to achieve linear separability. To address this limitation, we propose to explicitly incorporate low-level features obtained from shallow layers into the output layer for the pixel reconstruction task. By doing so, we alleviate the burden of the model having to focus excessively on these low-level details, allowing it to

spend its modeling abilities to capture these high-level semantics.

We denote the proposed method as **Multi-level Feature Fusion (MFF)**. Specifically, we extend the usage of the fusion strategy in the first pilot experiment and systematically investigate the design choices of multi-feature fusion, such as feature selection and fusion strategies. Despite the simplicity of the proposed method, it is a drop-in solution for unleashing the full modeling potential of pixel-based MIM approaches and has the following advantages:

- (1) Employing multi-level feature fusion can enhance the training efficiency of MAE by approximately  $\sim 5x$ , thus helping to reduce the carbon footprint. For example, by pre-training MAE with this strategy for only 300 epochs, we achieve semantic segmentation results that are on par with those obtained after 1600 epochs in the original paper.
- (2) We also consistently and significantly improve performance across all downstream tasks, including semi-supervised fine-tuning and linear probing. Notably, with a small model such as ViT-S, we outperform MAE by 2.8% on linear probing, 2.6% on semantic segmentation, and 1.2% on fine-tuning.
- (3) After evaluating our model on four out-of-distribution datasets, we observe that the approach with multi-level feature fusion exhibits greater robustness than the base method.

Furthermore, we conduct a thorough analysis to unveil how multi-feature fusion works for representation learning. Given the exploratory experiments from the perspective of latent features and optimization, we find that the fusion strategy attenuates high-frequency information in the latent features and flattens the loss landscapes. To summarize, our contributions are three-fold:

- Firstly, we develop a multi-level feature fusion strategy for isotropic backbones such as ViT, achieving superior results compared to various pixel-based MIM approaches.
- Secondly, we have conducted a thorough analysis of how this multi-level feature fusion strategy enhances the model from the perspectives of latent features and optimization. Our examination is meticulous and provides valuable insights.
- Lastly, we have performed extensive and rigorous ablation studies on the design details, which also strengthens the validity of our findings.

## 2. Related Works

**Self-supervised Learning** reviously, many works[9, 41] relied on abundant labeled datasets to achieve promising results. However, annotating this data requires a large number of human labors. Therefore, how to effectively capture useful semantics embedded in the abundance of data available on the Internet is currently a hot topic. In recent times, self-supervised learning has witnessed tremendous growth

in computer vision, following remarkable achievements in natural language processing. These methods cater to diverse inputs, including images[5, 1, 14, 54], videos[29, 21], and multi-modality inputs[36, 3]. They capture rich semantic information by creating effective proxy tasks, such as contrastive learning and masked image modeling, in large amounts of unlabeled data. In comparison to supervised learning[41, 42, 43], these self-supervised learning approaches have gradually outperformed them in numerous downstream tasks and possess immense potential to become the principal pre-training paradigm.

**Feature Pyramid** Utilizing multi-level features has been extensively studied in previous years, and one of the most famous applications is the Feature Pyramid Network (FPN)[26]. This technique has been widely used in many dense tasks such as object detection and semantic segmentation to improve the model’s perception of objects of different scales. Incorporating FPN into existing designs in many works[15, 47] has led to significant improvements. However, the multi-level feature fusion module only accepts features of different scales as input, limiting its adaptation to isotropic architectures such as ViT[9], in which features from different layers are of the same scale. In masked image modeling, most approaches choose ViT as their encoder due to the masked patch prediction task. Therefore, there are few works exploring multi-level feature fusion in this domain. Even though some works [11, 40] aim to explore multi-level fusion in masked image modeling, their applications are still limited to the traditional hierarchical architecture and do not address the issue of being biased toward low-level details for these pixel-based methods.

### 3. Methods

We now introduce the proposed multi-level feature strategy for pixel-based MIM methods. In Section 3.1, we first give a short revisit to the universal framework of pixel-based MIM approaches. Then Section 3.2 describes how to insert multi-level feature fusion into pixel-based MIM approaches. Finally, two key components, the projection layer and fusion strategy, will be discussed in Section 3.3.

#### 3.1. Introduction to Pixel-based MIM

We now introduce the unified formulation of recent pixel-based Masked Image Modeling(MIM) methods. This kind of MIM, using raw images as target, is a denoising auto-encoder [44], and it follows a simple pipeline. Its primary objective is to predict raw pixel values of the original or post-processed images, such as the high-frequency filtered image in PixMIM [30]. When dealing with masked images, we can feed only visible tokens into the encoder or both visible and mask tokens. If only visible tokens are used for the encoder, both the mask token and the latent feature output

by the encoder must be fed into the decoder.

#### 3.2. MIM with Multi-level Feature Fusion

Multi-level feature fusion(MFF) can be incorporated into most existing pixel-based MIM approaches in a plug-and-play manner. Figure 3 gives an overview of the whole framework. To keep the simplicity, we mainly focus on these steps relevant to MFF, leaving out other steps. Given an image  $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$ , we feed it into the encoder,  $\mathbb{E}$ , to get the latent representations:

$$\mathbf{X} = \mathbb{E}(\mathbf{I}) \tag{1}$$

The latent representations, denoted by  $\mathbf{X} = \{x_0, x_1, \dots, x_{N-2}, x_{N-1}\}$ , correspond to the output feature from each transformer layer of the ViT, where  $N$  represents the depth of the encoder. For the pilot experiment in Figure 1, we fuse all-level features from each layer of the encoder. However, indiscriminately fusing all of them may introduce redundancy or even makes the model much harder to be optimized. But finding the most effective layers to fuse induces a large search space. To simplify the layer selection procedure, we follow the guidelines below:

(1) : We first conduct an ablation study to compare the results of fusing shallow layers or deep layers (as shown in Figure 2, shallow layers contain low-level features, and deep layers contain high-level features), and more details are presented in Section 5.4. The results show that utilizing the features of the shallow layers performed significantly better than deep ones. Thus intuitive analysis and the quantitative numbers both indicate that the shallow layer should be selected for fusion.

(2) : We then examine how many layers should be taken into consideration. In addition to the selected shallow layer and output layer, we also try to explore introducing different numbers of intermediate layers for fusion. We refer the reader to Section 5.4 for more details of this experiment.

We finally selected  $M = 5$  additional layers besides the last layer (6 layers in total) and the output features from those layers are used for fusion. We define the indices for these selected layers as  $\mathcal{W}$ ,  $|\mathcal{W}| = M$ . After that, we apply a projection layer,  $\mathcal{P}_i$ , to each of the additional  $M$  layers before fusion.

$$\tilde{\mathbf{X}} = \{\mathcal{P}_i(x_i)\}_{i \in \mathcal{W}} + \{x_{N-1}\} \tag{2}$$

Adding a projection layer to align the feature space between different levels’ features is a common practice in self-supervised learning.

Finally, we introduce the fusion layer,  $\mathcal{F}$ , to fuse multi-level features  $\tilde{\mathbf{X}}$ :

$$\mathbf{O} = \mathcal{F}(\tilde{\mathbf{X}}) \tag{3}$$

$\mathbf{O}$  will be fed into the decoder for pixel reconstruction.

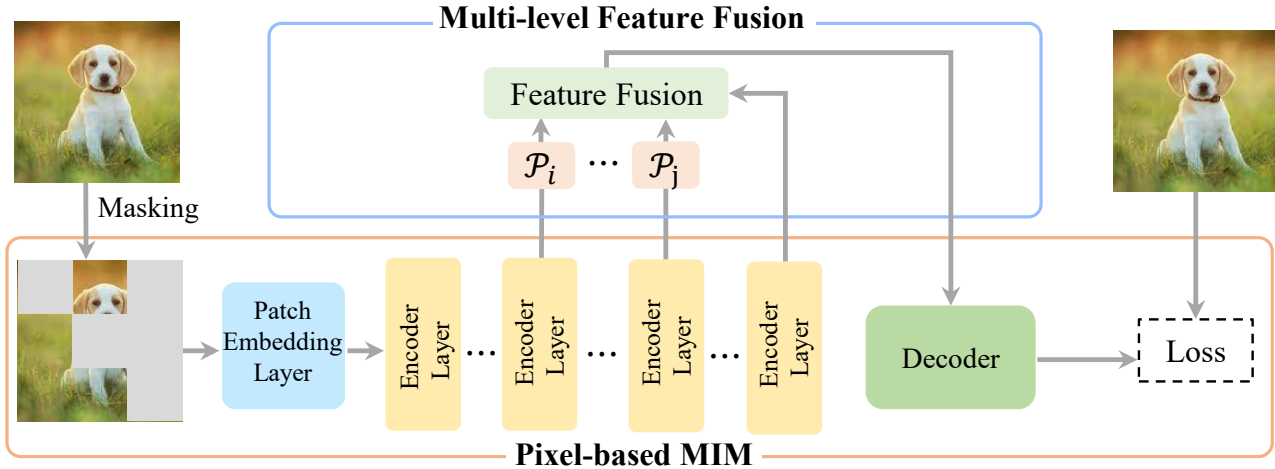


Figure 3: **Multi-level feature fusion for pixel-based MIM.** The multi-level feature fusion (MFF) module can be inserted into existing pixel-based MIM approaches in a plug-and-play manner.  $i, j \in \mathcal{W}$ .

### 3.3. Instantiation of Projection and Fusion Layers

We further investigate the instantiation of the projection layer and the fusion layer.

**Projection Layer:** In terms of the projection layer, indicated as  $\mathcal{P}$ , we focus on two popular options, namely linear projection and non-linear projection. Specifically, we instantiate the non-linear projection with the Linear-GELU-Linear structure. Our experiment has revealed that a simple linear layer is sufficient and effective within our framework.

**Fusion Layer:** The fusion layer aims to gather low-level information from the features of shallower layers feature. We evaluate two commonly employed fusion methods: weighted average pooling and self-attention-based fusion.

$$O = \sum_{i \in \mathcal{W}} w_i \mathcal{P}_i(x_i) + w_{N-1} x_{N-1} \quad (4)$$

The weighted average pooling fusion is illustrated in Equation 4. In this equation,  $w_i$  refers to the weight assigned to each of the  $M$  selected layers, while  $w_{N-1}$  is assigned to the output layer. During the training process, all these weights are dynamically updated and summed up to 1. As for the self-attention method, we use an off-the-shelf transformer layer.

$$\hat{O} = \text{MultiHeadAttention}([\mathcal{P}_i(x_i)_{i \in \mathcal{W}}, x_{N-1}]) \quad (5)$$

After the multi-head attention layer, we extract the transformed tokens corresponding to  $x_{N-1}$  from  $\hat{O}$  to use in pixel reconstruction. Experimental results demonstrate that the weighted average pooling strategy is comparable to self-attention for this purpose while also being simpler and more computationally efficient.

Our method is intuitive and simple, and can be inserted into most of pixel-based MIM approaches without introducing non-negligible computational overhead. We evaluate it on MAE[14] and PixMIM[30], and more detailed results are shown in Section 5.2.

## 4. Analysis

In order to uncover the dark secret behind multi-level feature fusion, we first conduct a frequency analysis on the model (with and without MMF) in Section 4.1. In Section 4.2, we find multi-level feature fusion can be helpful to the optimization of the model, by flattening the loss landscape. Finally, in Section 4.3, we apply the first pilot experiment introduced in Section 1 to EVA [10] and supervised ViT [14], to investigate whether they too require low-level features and to confirm that the bias towards low-level details is the unique and inherent drawback of pixel-based MIM.

### 4.1. Frequency Analysis

We employ multi-level feature fusion to enhance MAE [14], resulting in  $\text{MFF}_{\text{MAE}}$ . The aim of this fusion is to prevent the model from excessively focusing on low-level details. To investigate the change in frequency response before and after the fusion, we transform the feature from the last encoder layer into the frequency domain and calculate the amplitude of various frequency bands. As depicted in Figure 4, multi-level feature fusion diminishes high-frequency responses and amplifies those that belong to the low-frequency range, which supports the efficacy of the fusion technique.

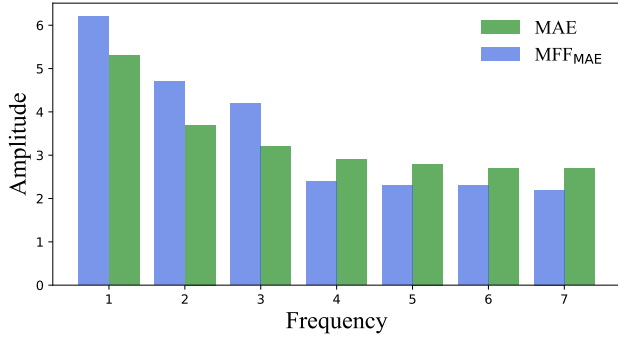


Figure 4: **Frequency analysis of output layer features from the encoder.** Multi-level feature fusion reduces high-frequency components and enhances low-frequency components. The lowest-frequency band is labeled as **1**, while the high-frequency band is labeled as **7**.

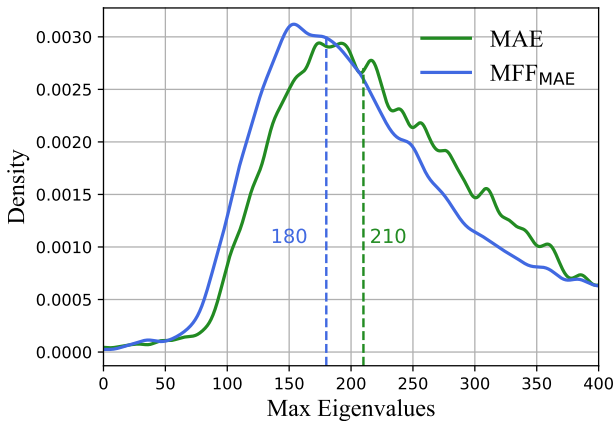


Figure 5: **Hessian max eigenvalues spectrum.** Multi-level feature fusion reduces the magnitude of Hessian max eigenvalues to flatten the loss landscape.

## 4.2. Optimization Analysis

Following [33], we analyze the Hessian max eigenvalue spectrum. Multi-level feature fusion has the additional benefit of reducing the magnitude of Hessian max eigenvalues. As shown in Figure 5, the expected hessian max eigenvalue of MFF<sub>MAE</sub> is smaller than that of MAE[14]. Hessian max eigenvalues represent the local curvature of the reconstruction loss function, and this result suggests that multi-level feature fusion flattens the loss landscape. Large eigenvalues can impede neural network training [12]. Therefore, *multi-level feature fusion can help the model learn better representations by suppressing large Hessian eigenvalues.*

## 4.3. Feature Bias of Different Pre-training Methods

In order to investigate whether being biased towards low-level features is the sole and inherent drawback of

pixel-based MIM, we introduce multi-level feature fusion to EVA[10] and supervised ViT[14]. EVA is one of the representative works that focuses on regressing high-level features produced by CLIP[36], while supervised ViT is one of the works that require the model to map the input image to a semantic label. Both EVA and supervised ViT target high-level features that contain rich semantic information describing the input image. As shown in Figure 6, unlike MAE, the weight of the last layer’s feature for both EVA and supervised ViT is significantly higher than that of the shallow layers. This observation suggests that the bias towards low-level features exhibited by these pixel-based MIM approaches is primarily caused by the raw-pixel reconstruction task.

## 5. Experiment

In Section 5.1, we present the experimental settings for pre-training and evaluation. Next, in Section 5.2, we apply MFF to two MIM baselines, namely MAE [14] and PixMIM [30], and show the improvements brought by such design. In addition, we also evaluate the effectiveness of MFF using a smaller model (*e.g.*, ViT-S) and fine-tune the pre-trained model under a low-shot setting. To evaluate the robustness of our proposed method, Section 5.3 includes additional analyses that assess the robustness of pre-trained models against out-of-distribution (OOD) ImageNet variants. Finally, Section 5.4 presents comprehensive ablation studies of our method.

### 5.1. Experiment Settings

To ensure the efficacy of our methods and design components, we conducted a series of extensive experiments on image classification using ImageNet-1K[8], object detection on COCO [27] and semantic segmentation on ADE20K[52]. Unless otherwise stated, our default settings are based on ViT-B.

**ImageNet-1K [8]** The ImageNet-1K dataset comprises 1.3 million images belonging to 1,000 categories and is divided into training and validation sets. To ensure the fairness of our experiments while applying our methods to MAE [14] and PixMIM[30], we strictly follow their original pre-training and evaluation settings on ImageNet-1K. This includes following the pre-training schedule, network architecture, learning rate setup, and fine-tuning protocols. Furthermore, in addition to the conventional fine-tuning protocol, we fine-tune the model using a low-shot setting, where only a fraction of the training set (*e.g.* 1% and 10%) is used. This approach is consistent with previous works[5] and we ensure that the low-shot fine-tuning setting also strictly follows that of the conventional fine-tuning.

**ADE20K [52]** To conduct the semantic segmentation experiments on ADE20K, we utilize the off-the-shelf set-

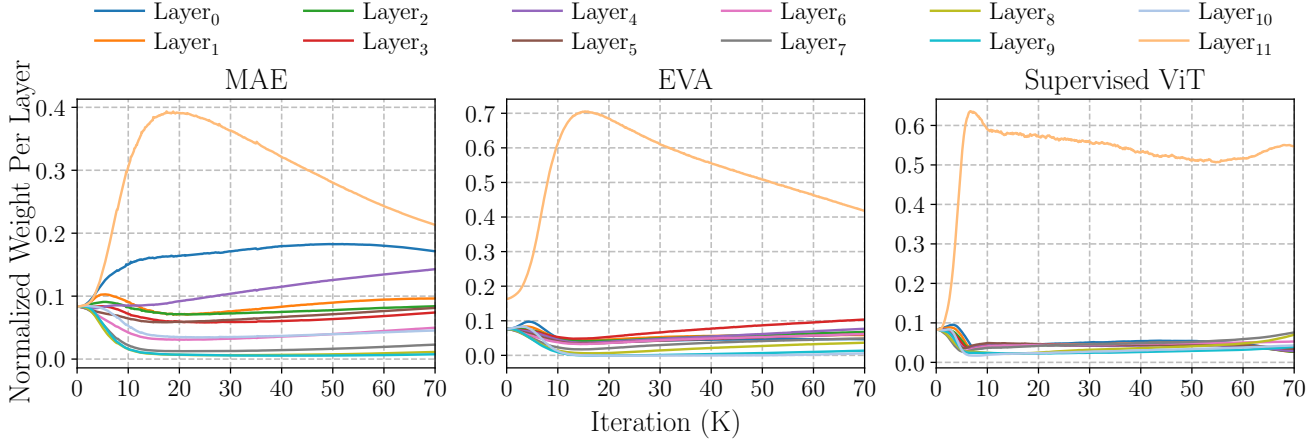


Figure 6: MAE [14] attempts to extract low-level features from shallow layers, whereas EVA [10] and DEiT [41] do not. MAE refers to MIM approaches that use the raw pixel as the target, while EVA refers to MIM approaches that use additional tokenizers like CLIP[36]. Supervised ViT denotes supervised learning approaches. Compared to EVA and supervised ViT, MAE exhibits a greater affinity for low-level features and displays a more aggressive appetite for them.

tings from MAE[14]. With this approach, we fine-tune a UperNet[47] for 160k iterations with a batch size of 16 and initialize the relative position bias to zero.

**COCO [27]** For our object detection experiments on COCO, we adopt the Mask R-CNN approach[15] that enables simultaneous production of bounding boxes and instance masks, with the ViT serving as the backbone. As in MAE, we evaluate the box and mask AP as the metrics for this task. However, we note that there is no universal agreement for the setting of object detection fine-tuning epochs. We have chosen the commonly used  $2\times$  setting, which fine-tunes the model for 25 epochs. Other settings strictly follow those in ViTDet [25].

**Ablation studies** We conduct all of our ablation studies based on the customary MAE settings [28, 6]. We pre-train all model variants on ImageNet-1K for 300 epochs and conduct a comprehensive performance comparison on linear probing, fine-tuning, and semantic segmentation. All other settings are consistent with those discussed previously.

## 5.2. Main Results

The application of multi-level feature fusion to MAE [14] and PixMIM [30] resulted in significant improvements in various downstream tasks, as shown in Table 1. After pre-training the model for 300 epochs, we achieve a 0.5%, 1.8%, and 3.6% improvement over MAE in fine-tuning, linear probing, and semantic segmentation, respectively. Additionally, our model exhibits scalability across pre-training epochs and consistently outperforms these base methods by a substantial margin. Compared to these methods, using an extra heavy tokenizer, *e.g.* CLIP, we also gradually close the performance gap with them. Although fine-tuning ac-

curacy is often considered a reliable measure of the quality of non-linear features in a model, we find that it is not a sensitive metric, as compared to other metrics presented in Table 1. This may be attributed to pre-training and fine-tuning following the same data distribution, and the size of the training set and model capacity being sufficient to offset the performance gap between different methods. To address this limitation, we adopt the following two workarounds: **Low-shot fine-tuning.** This protocol has also been adopted

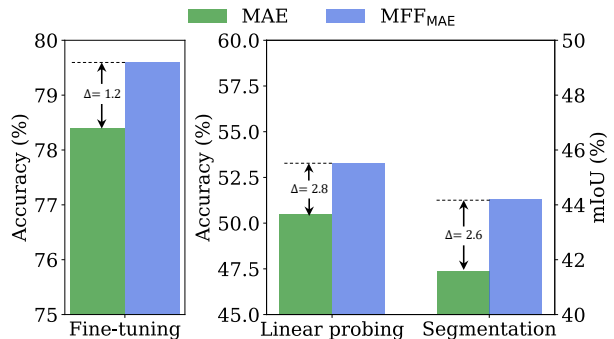


Figure 7: Performance on ViT-S. Applying MFF to ViT-S brings significant improvements on all downstream tasks. We reuse the same setting as for ViT-B, without specifically tuning.

by many previous works, *e.g.* [5]. Rather than utilizing the entire training set, we fine-tune the pre-trained model end-to-end using only 1% and 10% of the training set. As indicated by Table 1, the performance gap between MFF and the base methods is much more prominent when using low-shot fine-tuning, which further verifies the effectiveness of

Evaluation Protocol→			ImageNet		Low-shot		ADE20K
Method	Target	Epoch	ft(%)	lin(%)	1%	10%	mIOU
<b>Supervised learning</b>							
DeiT III[43]	-	800	83.8	-	-	-	49.3
<b>Masked Image Modeling w/ pre-trained target generator</b>							
BEiT[1]	DALLE	800	83.2	56.7	-	-	45.6
CAE[6]	DALLE	800	83.8	68.6	-	-	49.7
MILAN*[20]	CLIP-B	400	85.4	78.9	67.5	79.7	52.7
BEiT-v2[34]	VQ-KD	1600	85.5	80.1	-	-	53.1
MaskDistill[35]	CLIP-B	800	85.5	-	-	-	54.3
<b>Masked Image Modeling w/o pre-trained target generator</b>							
MaskFeat*[46]	HOG	1600	84.0	62.3	52.9	73.5	48.3
SemMAE[24]	RGB	800	83.4	65.0	-	-	46.3
SimMIM[48]	RGB	800	83.8	56.7	-	-	-
MAE*[14]	RGB	300	82.8	61.5	41.4	70.5	43.9
MFF <sub>MAE</sub>	RGB	300	83.3 (+0.5)	63.3 (+1.8)	43.7 (+2.3)	71.4 (+0.9)	47.7 (+3.6)
MAE*[14]	RGB	800	83.3	65.6	45.4	71.2	46.1
MFF <sub>MAE</sub>	RGB	800	83.6 (+0.3)	67.0 (+1.4)	48.0 (+2.6)	72.0 (+0.8)	47.9 (+1.8)
PixMIM[30]	RGB	800	83.5	67.2	47.9	72.2	47.3
MFF <sub>PixMIM</sub>	RGB	800	83.6 (+0.1)	68.2 (+1.0)	49.0 (+1.1)	73.0 (+0.8)	48.6 (+1.3)

Table 1: **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[9]. \*: numbers are reported by running the official code release. Low-shot: end-to-end fine-tuning with 1% and 10% of the training set.

MFF.

**Pre-train with ViT-S.** To mitigate the influence brought by model capacity, we pre-train MAE using ViT-S and compare their performance using fine-tuning, linear probing, and semantic segmentation. Since our objective is to evaluate the improvement brought by MFF to these base methods, we do not specifically tune hyper-parameters for the experiments with ViT-S to achieve state-of-the-art performance, but rather use the same settings as for ViT-B. Due to its smaller capacity compared to ViT-B, ViT-S requires a pre-training method that can effectively capture semantic features to perform well on downstream tasks. As demonstrated in Figure 7, the method with MFF significantly outperforms their base method, further validating the effectiveness of MFF.

We also evaluate our pre-trained models with the object detection protocol and report the  $AP^{\text{box}}$  and  $AP^{\text{mask}}$ . As shown in Table 2, MFF can still bring non-trivial improvements for object detection.

### 5.3. Robustness Evaluation

Robustness evaluation is a common practice in many previous works [53, 20, 14] to assess a model’s ability to handle different types of noise. In this study, we compare our pre-trained models with their correspond-

Method	$AP^{\text{box}}$	$AP^{\text{mask}}$
MAE	47.3	42.4
MFF <sub>MAE</sub>	48.0 (+0.7)	43.0 (+0.6)
PixMIM	47.8	42.8
MFF <sub>PixMIM</sub>	48.1 (+0.3)	43.1 (+0.3)

Table 2: **Results of COCO object detection.**

ing baselines on four out-of-distribution (OOD) ImageNet variants: ImageNet-Corruption [17], ImageNet-Adversarial [18], ImageNet-Rendition [16], and ImageNet-Sketch [45]. These datasets introduce various domain shifts to the original ImageNet-1K and are widely used to evaluate a model’s robustness and generalization ability. As illustrated in Table 4, MFF significantly improves the robustness of MAE and PixMIM on all datasets by a clear margin. The enhanced robustness against domain shifts strengthens the value of our simple yet effective method.

### 5.4. Ablation Studies

**Is shallow layer important?** In order to determine the significance of low-level features from shallow layers, we explore the fusion of the output layer with either a deep or shallow layer. So we try to fuse the output layer with an extra shallow or deep layer, selected from the previous 11

layers	ft	lin	seg	num layers	ft	lin	seg	linear	nonlinear	pool	attn	ft	lin	seg
output	82.8	61.5	43.9	1	82.8	61.5	43.9			✓		83.0	62.7	47.2
+shallow	<b>82.9</b>	<b>61.9</b>	<b>46.4</b>	2	82.9	61.9	46.4	✓		✓		<b>83.0</b>	<b>63.3</b>	<b>47.7</b>
+deep	82.8	61.7	44.5	4	83.0	63.0	46.6		✓	✓		83.0	63.0	46.2
				6	<b>83.0</b>	<b>63.3</b>	<b>47.7</b>	✓			✓	83.1	63.0	47.3
				12	83.3	62.4	46.8							

(a) Which layer for fusion      (b) Number of layers used for fusion      (c) Projection and fusion layer

Table 3: We conducted ablation studies with ViT-B/16 that was pre-trained on ImageNet-1K for 300 epochs. Our report includes results for fine-tuning (ft), linear probing (lin), and semantic segmentation (seg). The final settings are highlighted in gray. (a): **output** denotes the output layer of the encoder. (b): the second row corresponds to the second row of (a). (c): **linear** and **nonlinear** denotes linear and nonlinear projection layers, while **pool** and **attn** represents the fusion strategies of weight-average pooling and self-attention, respectively. The first row of column (c) does not use any projection layer before fusion.

layers of ViT-B[9]. And the specific index of the selected layer will be detailed in the appendix. As illustrated in Table 3a, fusing the output layer with a deep layer only results in marginal improvements. However, incorporating low-level features directly from the shallow layer into the output layer leads to a significant performance boost, as it enables the model to focus on semantic information. Therefore, we have decided to use a shallow layer (*i.e.* the first layer) for multi-level feature fusion.

**How many layers are used for fusion ?** Aside from the output layer and the shallow layer picked in the previous selection, it is reasonable to consider using intermediate layers for fusion, as they may contain additional low-level features or high-level meanings that could assist in the reconstruction task. However, selecting these intermediate layers is a daunting task due to the large search space involved. To simplify the process, we pick an additional 1, 2, and 5 layers evenly spaced between the shallow layer and output layer selected in Table 3a. The specific indices for these selected layers are placed in the appendix. As shown in Table 3b, introducing more layers brings consistent improvements because they may contain unique features, such as textures or colors, that help the model complete the reconstruction task. Nevertheless, when we fuse all these layers, we witness a performance drop in all downstream tasks. This drop may result from the difficulty of optimization, because of the redundancy in these layers.

Method	IN-C↓	IN-A	IN-R	IN-S
MAE	51.7	35.9	48.3	34.5
MFF <sub>MAE</sub>	49.0 (-2.7)	37.2 (+1.3)	51.0 (+2.7)	36.8 (+2.3)
PixMIM	49.9	37.1	49.6	35.9
MFF <sub>PixMIM</sub>	48.5 (-1.4)	40.1 (+3.0)	51.6 (+2.0)	37.8 (+1.9)

Table 4: **Robustness evaluation on ImageNet variants.** To evaluate the robustness of MFF, we further evaluate the models (after fine-tuning) from Table 1 on four ImageNet variants. Results are reported in top-1 accuracy, except for IN-C[17] that uses the mean corruption error.

**Do the projection layer and fusion strategy matter?** In Equation 2, we investigate the influence of the projection layer on the final results. Our findings indicate that a simple linear projection layer is sufficient to achieve satisfactory results, as compared to using no projection layer or a nonlinear projection layer. Incorporating a single linear projection layer offers benefits in mitigating the domain or distribution gap between different layers, as compared to using no projection layer. However, the addition of a nonlinear projection layer, which includes an extra linear projection and GELU activation before the linear projection, introduces computational overhead and is more challenging to optimize. As a result, the non-linear projection achieves sub-optimal performance. With regard to the fusion strategy, we found that the **weight-average pooling** strategy, which assigns a dynamic weight to each layer and then performs element-wise addition, achieves the best performance. Compared to **attn**, this strategy shares the merits of simplicity and smaller computational overhead.

## 6. Conclusion

In this study, we take the first step to systematically explore multi-level feature fusion for the isotropic architecture, such as ViT, in masked image modeling. Initially, we recognize that pixel-based MIM approaches tend to excessively rely on low-level features from shallow layers to complete the pixel value reconstruction task by a pilot experiment. We then apply a simple and intuitive multi-level feature fusion to two pixel-based MIM approaches, MAE and PixMIM, and observe significant improvements in both, gradually closing the performance gap with these approaches by using an extra heavy tokenizer. Finally, we conduct an extensive analysis of multi-level feature fusion and find that it can suppress high-frequency information and flatten the loss landscape. We believe that this work can provide the community with a fresh perspective on these pixel-based MIM approaches and continue to rejuvenate this kind of simple and efficient self-supervised learning paradigm.



## 7. Acknowledgement

This paper is supported by the National Key R&D Program of China (No. 2022ZD0161600) and Shanghai Postdoctoral Excellence Program (No.2022235).

## References

- [1] Hangbo Bao, Li Dong, and Furu Wei. BEiT: BERT pre-training of image transformers. *ArXiv*, 2021. [1](#), [3](#), [7](#), [11](#)
- [2] 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)*, pages 9650–9660, October 2021. [11](#)
- [3] Jiacheng Chen, Hexiang Hu, Hao Wu, Yuning Jiang, and Changhu Wang. Learning the best pooling strategy for visual semantic embedding. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. [3](#)
- [4] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pre-training from pixels. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1691–1703. PMLR, 13–18 Jul 2020. [11](#)
- [5] 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. [3](#), [5](#), [6](#)
- [6] 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. [6](#), [7](#)
- [7] 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. [11](#)
- [8] 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. [5](#)
- [9] 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. [2](#), [3](#), [7](#), [8](#), [11](#)
- [10] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. *arXiv preprint arXiv:2211.07636*, 2022. [4](#), [5](#), [6](#), [12](#)
- [11] 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. [3](#)
- [12] Behrooz Ghorbani, Shankar Krishnan, and Ying Xiao. An investigation into neural net optimization via hessian eigenvalue density. In *International Conference on Machine Learning*, pages 2232–2241. PMLR, 2019. [5](#)
- [13] 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. [11](#)
- [14] 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](#), [11](#)
- [15] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision (ICCV)*, 2017. [3](#), [6](#)
- [16] 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)*, 2021. [7](#)
- [17] Dan Hendrycks and Thomas G. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *ArXiv*, abs/1903.12261, 2019. [7](#), [8](#)
- [18] 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)*, 2021. [7](#)
- [19] Elad Hoffer, Tal Ben-Nun, Itay Hubara, Niv Giladi, Torsten Hoeffler, and Daniel Soudry. Augment your batch: better training with larger batches. *ArXiv*, abs/1901.09335, 2019. [11](#)
- [20] 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](#), [7](#), [12](#)
- [21] 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](#)
- [22] 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, editors, *Computer Vision – ECCV 2016*, pages 646–661, Cham, 2016. Springer International Publishing. [11](#)
- [23] Zhicheng Huang, Xiaojie Jin, Cheng Lu, Qibin Hou, Mingming Cheng, Dongmei Fu, Xiaohui Shen, and Jiashi Feng. Contrastive masked autoencoders are stronger vision learners. *ArXiv*, abs/2207.13532, 2022. [1](#)
- [24] 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)*, 2022. 7
- [25] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. In *ECCV*, 2022. 6
- [26] 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 conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 3
- [27] 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. 5, 6, 11
- [28] 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. 6
- [29] Yuan Liu, Jiacheng Chen, and Hao Wu. Moquad: Motion-focused quadruple construction for video contrastive learning. *ArXiv*, abs/2212.10870, 2022. 3
- [30] Yuan Liu, Songyang Zhang, Jiacheng Chen, Kai Chen, and Dahua Lin. Pixmim: Rethinking pixel reconstruction in masked image modeling. *arXiv preprint arXiv:2303.02416*, 2023. 1, 3, 4, 5, 6, 7, 11
- [31] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017. 11
- [32] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 11
- [33] Namuk Park and Songkuk Kim. How do vision transformers work? In *International Conference on Learning Representations*, 2022. 2, 5
- [34] 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, 2022. 1, 7
- [35] Zhiliang Peng, Li Dong, Hangbo Bao, Qixiang Ye, and Furu Wei. A unified view of masked image modeling. *ArXiv*, abs/2210.10615, 2022. 7
- [36] 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. 1, 3, 5, 6
- [37] 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. 1
- [38] Sucheng Ren, Fangyun Wei, Samuel Albanie, Zheng Zhang, and Han Hu. Deepmim: Deep supervision for masked image modeling. *arXiv preprint arXiv:2303.08817*, 2023. 12
- [39] 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)*, pages 2818–2826, 2016. 11
- [40] Yunjie Tian, Lingxi Xie, Zhaozhi Wang, Longhui Wei, Xiaopeng Zhang, Jianbin Jiao, Yaowei Wang, Qi Tian, and Qixiang Ye. Integrally pre-trained transformer pyramid networks. *arXiv preprint arXiv:2211.12735*, 2022. 3
- [41] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021. 2, 3, 6
- [42] Hugo Touvron, Matthieu Cord, Alaaeldin El-Nouby, Jakob Verbeek, and Herve Jegou. Three things everyone should know about vision transformers. *arXiv preprint arXiv:2203.09795*, 2022. 3
- [43] 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, editors, *ECCV*, 2022. 3, 7
- [44] 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. 3
- [45] Haohan Wang, Songwei Ge, Eric P. Xing, and Zachary Chase Lipton. Learning robust global representations by penalizing local predictive power. In *NeurIPS*, 2019. 7
- [46] 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. 7
- [47] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, 2018. 3, 6
- [48] 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. 7
- [49] Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv: Computer Vision and Pattern Recognition*, 2017. 11
- [50] Sangdoon 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)*, pages 6022–6031, 2019. 11
- [51] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. 11
- [52] 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. 5
- [53] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *International Conference on Learning Representations (ICLR)*, 2022. 7
- [54] 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

## A. Appendix

### A.1. Pre-training

The settings for pre-training strictly follows those in MAE[14] and PixMIM[30], with details shown below:

config	value
optimizer	AdamW [32]
base learning rate	1.5e-4
weight decay	0.05
optimizer momentum	$\beta_1, \beta_2=0.9, 0.95$ [4]
batch size	4096
learning rate schedule	cosine decay [31]
warmup epochs [13]	40

Table 5: Pre-training setting of  $\text{MFF}_{\text{MAE}}$  and  $\text{MFF}_{\text{PixMIM}}$

### A.2. Fine-tuning and linear probing

We also stick to the settings in MAE[14] for the ViT-B[9] model concerning fine-tuning and linear probing. Since our objective is to measure the enhancement brought by MFF and not attain the state-of-the-art (SOTA) performance, we employ the same settings as ViT-B without any specific adjustments for ViT-S.

config	value
optimizer	LARS [49]
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 6: Linear probing setting of  $\text{MFF}_{\text{MAE}}$  and  $\text{MFF}_{\text{PixMIM}}$ .

### A.3. Object detection and segmentation in COCO

All these settings also strictly follow those in MAE[14] but choose the commonly used  $2\times$  settings, which fine-tunes the model on COCO[27] for 25 epochs.

### A.4. Semantic segmentation in ADE20K

We stick to the settings used in MAE[14] and PixMIM[30], fine-tuning the pre-trained model end-to-end for 16k iterations with a batch size of 16.

### A.5. Selected indices of the ablation study

Inspired by the results of the pilot experiment depicted in Figure 1 of the main paper, we choose  $\text{layer}_0$  as the shallow

config	value
optimizer	AdamW[32]
base learning rate	1e-3
weight decay	0.05
optimizer momentum	$\beta_1, \beta_2=0.9, 0.999$
layer-wise lr decay [7, 1]	0.75
batch size	1024
learning rate schedule	cosine decay
warmup epochs	5
training epochs	100
augmentation	RandAug (9, 0.5) [19]
label smoothing [39]	0.1
mixup [51]	0.8
cutmix [50]	1.0
drop path [22]	0.1

Table 7: End-to-end fine-tuning setting of  $\text{MFF}_{\text{MAE}}$ ,  $\text{MFF}_{\text{PixMIM}}$

layer, and  $\text{layer}_{10}$  as the deep layer for the ablation experiment outlined in Table 3(a). Additionally, for ablation study in Table 3(b), we have selected additional two, four, and ten layers, evenly distributed between  $\text{layer}_0$  and the output layer ( $\text{layer}_{11}$ ). The detailed indices for Table 3(b) is shown in the Table 8.

num layers	indices
1	11
2	0,11
4	0,4,8,11
6	0,2,4,6,8,11
12	0,1,2,3,4,5,6,7,8,9,10,11

Table 8: **Detailed indices for Table 3(b) of the main paper.** We try to make the additionally selected indices are evenly distributed between the first layer and last layer.

In addition, similar to the pilot experiment in Figure 1 of the main paper, we observe the weight for each layer of all experiments in Table 3(b) of the main paper. Just as shown in Figure 8, no matter in which case, the model increasing relies on these shallow layers for the reconstruction tasks, indicating the significance of injecting low-level information into the output layer.

### A.6. Transfer learning

We also study transfer learning where we pre-train on ImageNet-1K and fine-tune on several smaller datasets. We follow the training recipe and protocol in DINO[2].  $\text{MFF}_{\text{MAE}}$  consistently outperforms MAE on CIFAR10, CIFAR100, and Stanford Cars. As shown in the following table,  $\text{MFF}_{\text{MAE}}$  consistently improves MAE on all datasets.

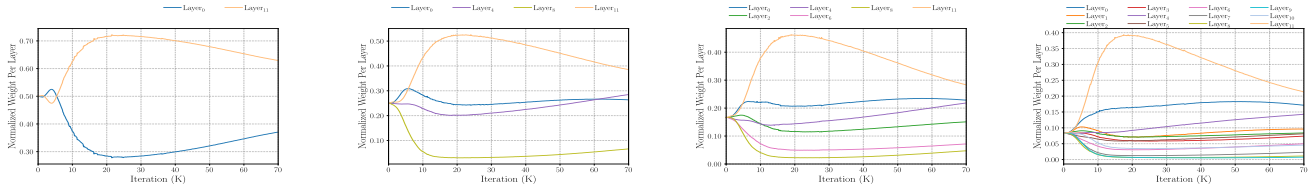


Figure 8: Model increasingly relies on shallow layers.

Method	Epoch	CIFAR10	CIFAR100	Cars
MAE	800	98.4	89.4	94.3
MFF <sub>MAE</sub>	800	98.6 (+0.2)	90.3 (+0.9)	94.7 (+0.4)

Table 9: Transfer learning on smaller datasets.

### A.7. Feature-based MIM does not Suffer from being Biased toward Low-level Feature

To supplement the findings in Figure 6, we apply multi-level feature fusion (MFF) to EVA[10] and MILAN[20], and evaluate their performance with linear probing, fine-tuning and semantic segmentation. Detailed results are shown below:

Method	Epoch	lin	seg	ft
EVA	400	69.0	49.5	83.7
MFF <sub>EVA</sub>	400	68.9	49.4	83.8
MILAN	400	79.9	52.7	85.4
MFF <sub>MILAN</sub>	400	79.7	52.9	85.0

As shown in the table above, MFF brings marginal improvements to feature-based MIMs, consistent with the findings in Figure 6.

### A.8. The Effect of Deep Supervision

To exclude the influence of deep supervision[38], we detach all shallow layers before fusing with the last layer (MFF<sub>MAE</sub><sup>detach</sup>), ensuring that gradients do not propagate through these shortcuts to the shallow layers. As shown in the table below, deep supervision alone does not improve MAE, and MFF’s improvements come from alleviating the problem of being biased toward high-freq components.

Method	model	epoch	lin	seg	ft
MFF <sub>MAE</sub>	ViT-B	800	67.0	47.9	83.6
MFF <sub>MAE</sub> <sup>detach</sup>	ViT-B	800	66.8 (-0.2)	48.0 (+0.1)	83.5(-0.1)