November 18 2021

☆ SimMIM – Simple framework for Masked Image Modeling
– simple design of each component: ① patch size 32×32 – protect ② regression RGB pixels ③ linear layer.

SimMIM: a Simple Framework for Masked Image Modeling

Zhenda Xie[1*] Zheng Zhang[2*] Yue Cao[2*]
Yutong Lin[3] Jianmin Bao[2] Zhuliang Yao[1] Qi Dai[2] Han Hu[2*]
[1]Tsinghua University [2]Microsoft Research Asia [3]Xi'an Jiaotong University
{t-zhxie,zhez,yuecao,t-yutonglin,jianmin.bao,t-zhuyao,qid,hanhu}@microsoft.com

Abstract

This paper presents SimMIM, a simple framework for masked image modeling. We have simplified recently proposed relevant approaches, without the need for special designs, such as block-wise masking and tokenization via discrete VAE or clustering. To investigate what makes a masked image modeling task learn good representations, we systematically study the major components in our framework, and find that the simple designs of each component have revealed very strong representation learning performance: 1) random masking of the input image with a moderately large masked patch size (e.g., 32) makes a powerful pre-text task; 2) predicting RGB values of raw pixels by direct regression performs no worse than the patch classification approaches with complex designs; 3) the prediction head can be as light as a linear layer, with no worse performance than heavier ones. Using ViT-B, our approach achieves 83.8% top-1 fine-tuning accuracy on ImageNet-1K by pre-training also on this dataset, surpassing previous best approach by +0.6%. When applied to a larger model with about 650 million parameters, SwinV2-H, it achieves 87.1% top-1 accuracy on ImageNet-1K using only ImageNet-1K data. We also leverage this approach to address the data-hungry issue faced by large-scale model training, that a 3B model (SwinV2-G) is successfully trained to achieve state-of-the-art accuracy on four representative vision benchmarks using 40× less labelled data than that in previous practice (JFT-3B). The code is available at https://github.com/microsoft/SimMIM.

1. Introduction

"*What I cannot create, I do not understand.*"
— Richard Feynman

"Masked signal modeling" is one such task that learns to create: masking a portion of input signals and trying to predict these masked signals. In NLP, following this philosophy, self-supervised learning approaches built on the masked language modeling tasks have largely repainted the field [2,12,32], i.e., learning very large-scale language models by using huge amounts of unlabeled data has been shown to generalize well to a broad range of NLP applications.

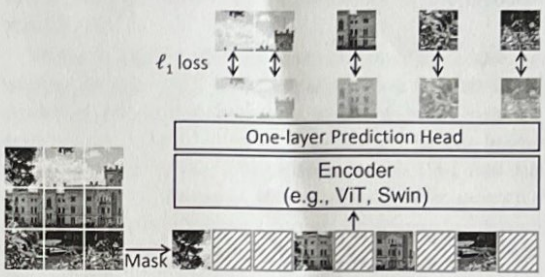

Figure 1. An illustration of our simple framework for masked language modeling, named *SimMIM*. It predicts raw pixel values of the randomly masked patches by a lightweight one-layer head, and performs learning using a simple ℓ_1 loss.

In computer vision, although there are pioneers leveraging this philosophy for self-supervised representation learning [13, 60, 61], in previous years, this line of work was almost buried by the contrastive learning approaches [8, 21, 51]. The different difficulties of applying this task to the language and visual domains can be explained by the differences between two modalities. One of the differences is that images exhibit stronger locality: pixels that are close to each other tend to be highly correlated [27], so the task can be done by duplicating close pixels rather than by semantic reasoning. Another difference is that visual signals are *raw and low-level*, while text tokens are human-generated *high-level* concepts. This raises a question of whether the prediction of low-level signals is useful for high-level visual recognition tasks. A third difference is that the visual signal is *continuous*, and the text token is *discrete*. It is unknown how classification-based masked language modeling approaches can be adapted to handle continuous visual signals well.

Until recently, there have been trials that attempt to bridge modality gaps and resolve the obstacles, by introduc-

*Equal. Zhenda, Yutong, Zhuliang are long-term interns at MSRA.

☆ Computer Vision Masked Signal Modeling – similar to Masked Autoencoder

ing several special designs, for example, by converting continuous signals into color clusters [7], by patch tokenization using an additional network [1], or by a block-wise masking strategy to break short-range connections [1], etc. Through these special designs, the learned representations proved to be well transferable to several visual recognition tasks.

In contrast to requiring special complex designs, in this paper, we present a simple framework which aligns well with the nature of visual signals, as shown in Figure 1, and is able to learn similar or even better representations than previously more complex approaches: *random masking of input image patches, using a linear layer to regress the raw pixel values of the masked area with an ℓ_1 loss.* The key designs and insights behind this simple framework include:

(1)
- Random masking is applied on image patches, which is simple and convenient for vision Transformers. For masked pixels, either larger patch size or higher masking ratio can result in a smaller chance of finding visible pixels that are close. For a large masking patch size of 32, the approach can achieve competitive performance in a wide range of masking ratios (10%-70%). For a small mask patch size of 8, the masking ratio needs to be as high as 80% to perform well. Note that the preferred masking ratios are very different from that in the language domain, where a small masking ratio of 0.15 is adopted as default. We hypothesize that different degrees of information redundancy in two modalities may lead to the different behaviors.

(2)
- A raw pixel regression task is used. The regression task aligns well with the continuous nature of visual signals, which possesses ordering property. This simple task performs no worse than the classification approaches with classes specially defined by tokenization, clustering, or discretization.

(3)
- An extremely lightweight prediction head (e.g., a linear layer) is adopted, which achieves similarly or slightly better transferring performance than that of heavier prediction heads (e.g., an inverse Swin-B). The use of an extremely lightweight prediction head brings a remarkable speedup in pre-training. In addition, we note that a broad range of target resolutions (e.g., 12^2-96^2) perform competitive with the highest 192^2. While heavier heads or higher resolutions generally result in greater generation capability, this greater capability does not necessarily benefit down-stream fine-tuning tasks.

Though simple, the proposed *SimMIM* approach is very effective for representation learning. Using ViT-B, it achieves 83.8% top-1 fine-tuning accuracy on ImageNet-1K, surpassing previous best approach ([1]) by +0.6%. *SimMIM* has also shown to be scalable to larger models: with a SwinV2-H model (658M parameters) [33], it achieves 87.1% top-1 accuracy on ImageNet-1K classification, which is the highest number among methods that use ImageNet-1K data only. This result encourages the use of self-supervised learning to address the increasing data-hungry problem caused by quickly rising model capacity. In fact, with the help of *SimMIM*, we successfully trained a SwinV2-G model with 3 billion parameters [33] using ~40× smaller data than that of Google's JFT-3B dataset, and set new records on several representative benchmarks: 84.0% top-1 accuracy on ImageNet-V2 classification [42], 63.1/54.4 box/mask mAP on COCO object detection [6,31], 59.9 mIoU on ADE20K semantic segmentation [52, 63], and 86.8% top-1 accuracy on Kinetics-400 action recognition [28,35].

While in recent years we have witnessed an increasing overlap between NLP and computer vision in both basic modeling and learning algorithms, as well as in multi-modal applications, which aligns well with how human brains achieve general intelligence capabilities, we hope that our demonstration of "masked signal modeling" in computer vision can drive this trend a bit further and encourage deeper interaction of different AI fields.

2. Related Work

Masked language modeling (MLM) Masked language modeling [12,32] and its auto-regressive variants [2] are the dominant self-supervised learning approaches in the field of natural language processing (NLP). Given visible tokens in a sentence or a sentence pair / triplet, the approaches learn representations by predicting invisible tokens of the input. This line of approaches has repainted the field since about 3 years ago [12], that it enables the learning of very large language models and generalizes well on broad language understanding and generation tasks by leveraging huge data.

Masked image modeling (MIM) Masked image modeling [7, 13, 23, 38, 47] progressed in parallel with the MLM task in NLP but located in a non-mainstream position for a long time. The context encoder approach [38] is a pioneer work in this direction, which masks a rectangle area of the original images, and predicts the missing pixels. CPC [23,47] predicts patches via a verification task in each batch with a contrastive predictive coding loss. Recently, iGPT [7], ViT [15] and BEiT [1] recall this learning approach on the modern vision Transformers, and show strong potential in representation learning by introducing special designs on some components, such as clustering on pixels [7], prediction of mean color [15], and tokenization via an additional dVAE network with a block-wise masking strategy [1]. In contrary to these complex designs, we present an extremely simple framework, SimMIM, which shows similar or even slightly better effectiveness.

Reconstruction based methods are also related to our approach, particularly the auto-encoder approaches [24, 30, 37, 41, 49, 50]. Similar as in our approach, they adopt a reconstruction task to recover the original signals. However, they are based on a different philosophy of visible signal *reconstruction*, other than the *creation or prediction* of invisible signals as in our approach. They thus progress in a very different path, by studying how to effectively regularize the task learning by proper regularization or architecture bottlenecks.

Image inpainting methods Beyond representation learning, masked image modeling is a classical computer vision problem, named image inpainting. This problem has been extensively studied in computer vision for a long time [39, 55, 56], aiming for improving the inpainting quality and without connecting to self-supervised representation learning. While we advocate image inpainting as a strong self-supervised pre-text task, we also find stronger inpainting capability does not necessarily leads to stronger fine-tuning performance on down-stream tasks.

Compressed sensing The approach in this paper is also related to compressed sensing [14], which affirms most of the data we acquire including image signals can be thrown away with almost no perceptual loss. Such claim is also partly supported by recent works of sparse inference [20] that the recognition accuracy has very little drop after throwing a large portion of image features [25, 43, 54]. The observation in this paper goes further for the input signals, that with an extremely small portion of randomly selected input image patches as input, i.e., 10%, the inpainting task can still be learnt to produce good visual representations.

Other self-supervised learning approaches During the last two decades, there have been numerous pretext tasks to learn visual representation in a self-supervised way: gray-scale image colorization [60], jigsaw puzzle solving [36], split-brain auto-encoding [61], rotation prediction [18], learning to cluster [4]. Though very different from masked image modeling, some of them interestingly also follow a philosophy of predicting the invisible parts of signals, e.g., [60, 61] use one or two color channels as input to predict values of other channels. Another large portion of works lie in the contrastive learning approaches [3, 8, 16, 19, 21, 51, 53], which are the previous mainstream. We hope our work can encourage the study of masked language modeling as a pre-text task for self-supervised visual representation learning.

3. Approach

3.1. A Masked Image Modeling Framework

Our approach *SimMIM* learns representation through masked image modeling, which masks a portion of input image signals and predicts the original signals at masked area. The framework consists of 4 major components:

1) *Masking strategy.* Given an input image, this component designs how to select the area to mask, and how to implement masking of selected area. The transformed image after masking will be used as the input.
2) *Encoder architecture.* It extracts a latent feature representation for the masked image, which is then used to predict the original signals at the masked area. The learnt encoder is expected to be transferable to various vision tasks. In this paper, we mainly consider two typical vision Transformer architectures: a vanilla ViT [15] and Swin Transformer [34].
3) *Prediction head.* The prediction head will be applied on the latent feature representation to produce one form of the original signals at the masked area.
4) *Prediction target.* This component defines the form of original signals to predict. It can be either the raw pixel values or a transformation of the raw pixels. This component also defines the loss type, with typical options including the cross-entropy classification loss and the ℓ_1 or ℓ_2 regression losses.

In the following subsections, we will present typical options of each component. These options are then systematically studied. By combining simple designs of each component, we have been able to achieve strong representation learning performance.

3.2. Masking Strategy

For input transformation of masked area, we follow the NLP community [12, 32] and BEiT [1] to use a learnable mask token vector to replace each masked patch. The token vector dimension is set the same as that of the other visible patch representation after patch embedding. For masking area selection, we study the following masking strategies (illustrated in Figure 2):

Patch-aligned random masking. We first present a patch-aligned random masking strategy. Image patches are the basic processing units of vision Transformers, and it is convenient to operate the masking on patch-level that a patch is either fully visible or fully masked. For Swin Transformer, we consider equivalent patch sizes of different resolution stages, 4×4∼32×32, and adopt 32×32 by default which is the patch size of the last stage. For ViT, we adopt 32×32 as the default masked patch size.

Other masking strategies We also try other masking strategies in previous works: 1) [38] introduces a central region masking strategy. We relax it to be randomly movable on the image. 2) [1] introduces a complex block-wise masking strategy. We try this mask strategy on two masked patch sizes of 16 × 16 and 32 × 32.

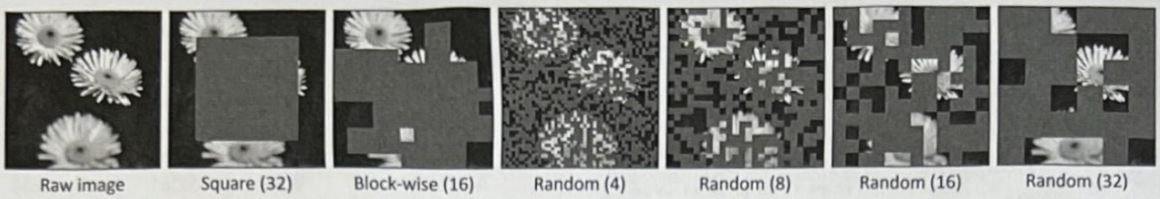

Figure 2. Illustration of masking area generated by different masking strategies using a same mask ratio of 0.6: square masking [38], block-wise masking [1] apply on 16-sized patches, and our simple random masking strategy on different patch sizes (e.g., 4, 8, 16 and 32).

3.3. Prediction Head

The prediction head can be of arbitrary form and capacity, as long as its input conforms with the encoder output and its output accomplishes the prediction target. Some early works follow auto-encoders to employ a heavy prediction head (decoder) [38]. In this paper, we show that the prediction head can be made extremely lightweight, as light as *a linear layer*. We also try heavier heads such as a 2-layer MLP, an inverse Swin-T, and an inverse Swin-B.

3.4. Prediction Targets

Raw pixel value regression The pixel values are continuous in the color space. A straight-forward option is to predict raw pixels of the masked area by regression. In general, vision architectures usually produce feature maps of down-sampled resolution, e.g., 16× in ViT and 32× for most other architectures. To predict all pixel values at a full resolution of input images, we map each feature vector in feature map back to the original resolution, and let this vector take charge of the prediction of corresponding raw pixels.

For example, on the 32× down-sampled feature maps produced by a Swin Transformer encoder, we apply a 1×1 convolution (linear) layer with output dimension of $3072 = 32 \times 32 \times 3$ to stand for the RGB values of 32×32 pixels. We also consider lower resolution targets by downsampling the original images by $\{32\times, 16\times, 8\times, 4\times, 2\times\}$, respectively.

An ℓ_1-loss is employed on the masked pixels:

$$L = \frac{1}{\Omega(\mathbf{x}_M)} \|\mathbf{y}_M - \mathbf{x}_M\|_1, \tag{1}$$

where $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{3HW \times 1}$ are the input RGB values and the predicted values, respectively; M denotes the set of masked pixels; $\Omega(\cdot)$ is the number of elements. We also consider ℓ_2 and smooth-ℓ_1 loss in experiments which perform similarly well, and ℓ_1 loss is adopted by default.

Other prediction targets Previous approaches mostly convert the masked signals to clusters or classes, and then perform a classification task for masked image prediction.

- *Color clustering.* In iGPT [7], the RGB values are grouped into 512 clusters by k-means using a large amount of natural images. Each pixel is then assigned to the closest cluster center. This approach requires an additional clustering step to generate the 9-bit color palette. In our experiments, we use the 512 cluster centers learnt in iGPT.
- *Vision tokenization.* In BEiT [1], a discrete VAE (dVAE) network [40] is employed to transform image patches to dVAE tokens. The token identity is used as the classification target. In this approach, an additional dVAE network needs to be pre-trained.
- *Channel-wise bin color discretization.* The R, G, B channels are separately classified, with each channel discretized into equal bins, e.g., 8 and 256 bins used in the experiments.

3.5. Evaluation protocols

We follow [1] to mainly evaluate the quality of learnt representations by fine-tuning on ImageNet-1K image classification, which is a more usable scenario in practice. We will mainly account for this metric in our ablations. In the system-level comparison, we also follow previous works [1, 3, 7, 8, 19, 21] to report the performance on previous dominant metric of linear probing. Nevertheless, we will not account on this linear probing metric, as our main goal is to learn representations which can well complement the following down-stream tasks.

4. Experiments

4.1. Ablation Study

4.1.1 Settings

We adopt Swin-B [34] as the default backbone in our ablation study, which facilitates us to evaluate the learnt representations also on downstream tasks such as object detection and semantic segmentation (see Appendix). To reduce experimental overhead, we use a default input image size of 192^2 and adapt the window size as 6 to accommodate the changed input image size. The ImageNet-1K image classification dataset is used for both pre-training and fine-tuning.

In self-supervised pre-training, we employ an AdamW optimizer [29] with a *cosine* learning rate scheduler, and train for 100 epochs. The training hyper-parameters are: the batch size as 2048, base learning rate as 8e-4, weight decay as 0.05, $\beta_1 = 0.9$, $\beta_2 = 0.999$, warm-up for 10

Mask Type	Masked patch size	Mask ratio	Top-1 acc (%)
square	32	0.11 (2×2)	82.6
	32	0.25 (3×3)	82.5
	32	0.44 (4×4)	82.5
block-wise	16/32	0.4	82.7/82.7
	16/32	0.6	82.6/82.6
	16/32	0.8	82.4/82.5
random	4/8/16/32	0.4	81.9/82.0/82.4/82.9
	4/8/16/32	0.6	82.0/82.1/82.7/82.8
	4/8/16/32	0.8	82.1/82.4/82.8/82.4
	64	0.1	82.6
	64	0.2	82.6
random	32	0.1	82.7
	32	0.2	82.8
	32	0.3	82.8
	32	0.4	82.9
	32	0.5	**83.0**
	32	0.6	82.8
	32	0.7	82.7
	32	0.8	82.4
	32	0.9	82.4

Table 1. Ablation on different masking strategies (i.e., square, block-wise, and random) with different masked patch sizes (i.e., 4, 8, 16, 32 and 64).

epochs. A light data augmentation strategy is used: random resize cropping with scale range of $[0.67, 1]$ and a aspect ratio range of $[3/4, 4/3]$, followed by a random flipping and a color normalization steps.

The default options for the components of SimMIM are: a random masking strategy with a patch size of 32×32 and a mask ratio of 0.6; a linear prediction head with a target image size of 192^2; an ℓ_1 loss for masked pixel prediction. Our ablation is conducted by varying one option and keeping other settings the same as that of the default.

In fine-tuning, we also employ an AdamW optimizer, 100-epoch training, and a cosine learning rate scheduler with 10-epoch warm-up. The fine-tunig hyper-parameters are: the batch size as 2048, a base learning rate of 5e-3, a weight decay of 0.05, $\beta_1 = 0.9$, $\beta_2 = 0.999$, a stochastic depth [26] ratio of 0.1, and a layer-wise learning rate decay of 0.9. We follow the same data augmentation used in [1], including RandAug [10], Mixup [59], Cutmix [57], label smoothing [45], and random erasing [62].

4.1.2 Masking Strategy

We first study how different masking strategies affect the effectiveness of representation learning. The fine-tuning accuracy of different approaches under multiple masking ratios are summarized in Table 1.

We first notice that the best accuracy of our simple random masking strategy reaches 83.0%, which is +0.3% higher than the best of other more specially designed strategies such as the block-wise masking as in [1].

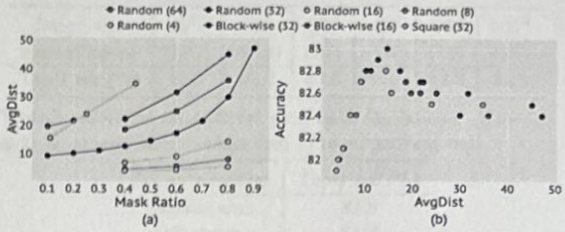

Figure 3. (a) *AvgDist* (averaged distance of masked pixels to the nearest visible pixels) w.r.t. different masking ratios using different masking strategies and different masked patch sizes; (b) fine-tuning performance (top-1 accuracy) w.r.t. *AvgDist*.

Head	#params	Training costs	Top-1 acc (%)
Linear	89.9M	1×	82.8
2-layer MLP	90.9M	1.2×	82.8
inverse Swin-T	115.2M	1.7×	82.4
inverse Swin-B	174.8M	2.3×	82.5

Table 2. Ablation on different prediction heads. A simple linear layer performs the best with lower training costs.

In addition, when a large masked patch size of 32 is adopted, this simple strategy performs stably well on a broad range of masking ratios (10%-70%). We hypothesize that the centering pixel of a large masked patch may be distant enough to visible pixels. Thus it enforces the network to learn relatively long-range connections, even when a low masking ratio is used (e.g., 10%) or all patches around are not masked. Another way to increase the prediction distance is to use larger masking ratio, which also shows to benefit the fine-tuning performance of relatively small patch sizes. By increasing the masking ratio from 0.4 to 0.8 at a patch size of 4, 8 and 16, the accuracy is smoothly improved by +0.2% (from 81.9% to 82.1%), +0.4% (from 82.0% to 82.4%), and +0.4% (from 82.4% to 82.8%), respectively. Nevertheless, the overall accuracy at these smaller patches is not as high as that at a larger patch size of 32. Further increasing the patch size to 64 is observed with degraded accuracy, probably due to the too large prediction distance.

The above observations and analyses can also be well reflected by a newly proposed *AvgDist* metric, which measures the averaged Euclidean distance of masked pixels to the nearest visible ones. The *AvgDist* of different masking strategies w.r.t. varying masking ratios are shown in Figure 3(a). From this figure, we observe that the *AvgDist* of all masking strategies is smoothly increased with growing masking ratios. For random masking strategy, when the masked patch size is low, e.g., 4 or 8, the *AvgDist* is relatively low and grows slowly with increasing masking ratios. On the other hand, when the patch size is large, e.g., 64, very small masking ratio (e.g. 10%) still makes relatively large *AvgDist*. The square and block-wise methods produce

similarly high *AvgDist* values as of patch size 64.

Figure 3(b) plots the relationship between fine-tuning accuracy and the *AvgDist* measure, which follows a ridge shape. The entries of high fine-tuning accuracy roughly distribute in a range of $[10, 20]$ of *AvgDist*, while entries with smaller or higher *AvgDist* perform worse. This indicates that the prediction distance in masked image modeling is encouraged to be moderate, neither too large nor too small. Probably, small distance in masked prediction may let the network learn too much short connections, while large distance may be too difficult to learn. These results also indicate that *AvgDist* may be a good indicator for the effectiveness of masked image modeling.

In our experiments, we adopt a masking ratio of 0.6 on patch size of 32 by default, due to its stable performance. Also note that the masking strategies and ratios in the language domain are very different from what explored in our work, which usually adopts a small masking ratio of 15%. We hypothesize that different degrees of information redundancy by two modalities may lead to the different behaviors.

4.1.3 Prediction Head

Table 2 ablates the effect of different prediction heads, including a linear layer, a 2-layer MLP, an inverse Swin-T and an inverse Swin-B. While generally heavier heads produce slightly lower losses, for example, 0.3722 (inverse Swin-B) versus 0.3743 (a linear layer), the transferring performances on the down-stream ImageNet-1K task are lower. It indicates that stronger inpainting capability does not necessarily result in better down-stream performance. It is probably because that the capacity is largely wasted in the prediction head, which will not be used in down-stream tasks. There is also a practical drawback, that a heavier prediction head brings higher training costs, e.g., the training cost of using an inverse Swin-B is 2.3× of that by a linear layer.

Also note that in previous contrastive learning approaches [8, 19, 21], it is a common practice to use a multi-layer MLP head in the pre-text tasks, instead of a linear layer, which makes the latent feature produced by the encoder moderately distant to the pre-text target, and shows beneficial for the linear probing evaluation metric. In our work, we show that a single linear layer head in our approach, under a fine-tuning metric, has shown competitive or even the optimal transferring performance. It indicates that if our aim is to learn good features for fine-tuning, the important exploration on head designing in contrastive learning approaches may not be necessary for that of masked image modeling.

4.1.4 Prediction Resolution

Table 3 ablates the effect of varying target resolution. It shows that a large range of resolutions (e.g., 12^2-192^2) perform equally well. The transferring performance drops only at a low resolution of 6^2, probably because this option throws too much information away. These results imply the information granularity required by the down-stream image classification task. The effects to other more fine-grained down-stream tasks such as object detection or semantic segmentation will be explored in our future study.

Image size	6^2	12^2	24^2	48^2	96^2	192^2
(ratio of inputs)	(1/32)	(1/16)	(1/8)	(1/4)	(1/2)	(1/1)
Top-1 acc (%)	82.3	82.7	82.8	82.7	82.8	82.8

Table 3. Ablation on different prediction resolutions. A moderately large resolution (no less than $1/16$ all perform well.

Scope to predict	Top-1 acc (%)
masked area	82.8
full image	81.7

Table 4. Ablation on different performing areas of prediction loss. If the loss is computed at masked area, it performs a pure prediction task. If it is computed on the whole image (both masked & unmasked areas), it performs a joint prediction and reconstruction task.

Note that we adopt a default target resolution of 192^2 in our experiments, due to the equally best transferring accuracy and the negligible computation overhead.

4.1.5 Prediction Target

Table 5 compares the effects of different prediction targets. Several observations can be drawn as follows:

- The three losses of ℓ_1, smooth-ℓ_1, and ℓ_2 perform similarly well;
- Carefully defined classes by color clustering [7] or tokenization [1] perform slightly worse than ours;
- A simple color discretization approach by channel-wise equal-sized bins (proposed as an alternative option) performs competitive to ℓ_1 loss, but it requires a careful tuning of the bin number (e.g., 8-bin).

It reveals that it is not necessary to align the target of masked image modeling to be the same classification based as masked language modeling. It is good to align the approach to the own nature of visual signals.

Prediction or reconstruction? While both auto-encoders and masked image modeling approaches learn a network by recovering the original signals, they are built on different philosophies of *visible signal reconstruction* and *prediction of invisible signals*. In our framework, we can instantiate a reconstruction task by also regress the raw pixel values of visible patches in the input.

Table 4 compares the approach which predicts only the masked area as in our default setting and an alternative to recover both masked and unmasked area. The approach predicting the masked area performs significantly better than

Loss	Pred. Resolution	Top-1 acc (%)
Classification		
8-bin	192^2	82.7
8-bin	48^2	82.7
256-bin	192^2	N/A
256-bin	48^2	82.3
iGPT cluster	192^2	N/A
iGPT cluster	48^2	82.4
BEiT	-	82.7
Regression		
ℓ_2	192^2	82.7
smooth-ℓ_1	192^2	82.7
ℓ_1	192^2	82.8
ℓ_1	48^2	82.7
ℓ_1	6^2	82.3

Table 5. Ablation on different prediction targets.

Methods	Input Size	Fine-tuning Top-1 acc (%)	Linear eval Top-1 acc (%)	Pre-training costs
Sup. baseline [46]	224^2	81.8	-	-
DINO [5]	224^2	82.8	78.2	2.0×
MoCo v3 [9]	224^2	83.2	76.7	1.8×
ViT [15]	384^2	79.9	-	∼4.0×
BEiT [1]	224^2	83.2	56.7	1.5×†
Ours	224^2	**83.8**	56.7	1.0×

Table 6. System-level comparison using ViT-B as the encoder. Training costs are counted in relative to our approach. † BEiT requires an additional stage to pre-train dVAE, which is not counted.

that recovering all image pixels as 82.8% vs. 81.7%. This implies that the two tasks are fundamentally different in their internal mechanisms, and the task to *predict* might be a more promising representation learning approach.

4.2. Comparison to Previous Approaches on ViT-B

As previous works [1, 5] performed experiments on the ViT architectures, for fair comparison, we also conduct experiments using the ViT-B architecture.

In pre-training, 800 epochs with a cosine learning rate scheduler and a 20-epoch linear warm-up procedure are employed. All other hyper-parameters strictly follow the same settings as in the ablation study, except that we use a 224^2 input resolution to be the same as in previous approaches. In fine-tuning, we adopt a layer-wise learning rate decay of 0.65 following [1], and keep all other settings strictly the same as in our ablation study. In linear probing, we follow [1] to choose an inter-mediate layer of ViT-B which produces the best linear probing accuracy. 100-epoch training with a 5-epoch linear warm-up step is employed.

Table 6 compares our approach to previous ones on both metrics of fine-tuning and linear probing using ViT-B. Our approach achieves a top-1 accuracy of 83.8% by fine-tuning, which is +0.6% higher than previous best approach [1]. Also note that our approach reserves the highest training efficiency than others thanks to its simplicity, that it is 2.0×, 1.8×, ∼4.0×, and 1.5× more efficient than that of DINO [5], MoCo v3 [9], ViT [15], and BEiT [1] (not counting the time for dVAE pre-training), respectively.

Though our main focus is to learn representations that are better for fine-tuning, we also report the linear probing accuracy of different approaches for reference.

Methods	Pre-train	Fine-tune	Backbone	Top-1 acc (%)	Param
Sup.	192^2	224^2	Swin-B	83.3	88M
Sup.	192^2	224^2	Swin-L	83.5	197M
Sup.	192^2	224^2	SwinV2-H	83.3	658M
Ours	192^2	224^2	Swin-B	84.0	88M
Ours	192^2	224^2	Swin-L	85.4	197M
Ours	192^2	224^2	SwinV2-H	85.7	658M
Ours	192^2	512^2	SwinV2-H	87.1	658M
Ours	192^2	640^2	SwinV2-G	90.2	3.0B

Table 7. Scaling experiments with Swin Transformer as backbone architectures. All our models are pre-trained with input of 192^2. Different to other models, Swin-G is trained on a privately collected ImageNet-22K-ext dataset, with details described in [33].

4.3. Scaling Experiments with Swin Transformer

We adopt Swin Transformer of different model sizes for experiments, including Swin-B, Swin-L, SwinV2-H, and SwinV2-G [33]. To reduce experimental overheads, we adopt a smaller image size of 192^2 in pre-training, and a step learning rate scheduler that the experiments of different training lengths can reuse model training of the first step. The base learning rate of the first learning rate step is set 4e-4 and lasts for 7/8 of the total training epochs. The learning rate is divided by 10 for the remaining epochs. For model sizes of H and G, we use the variants introduced in [33], which have stronger stability than the original version. All models use the ImageNet-1K dataset for training, except that SwinV2-G uses a larger and privately collected ImageNet-22K-ext dataset, as detailed in [33].

When using ImageNet-1K for pre-training, all models are trained by 800 epochs, with most other hyper-parameters following that in ablations. In fine-tuning, a larger image size of 224^2 is employed. For SwinV2-H, we also consider a larger resolution of 512^2. The training length of fine-tuning is set 100-epoch, except for SwinV2-H where 50-epoch is used. The layer-wise learning rate decay is set as 0.8, 0.75, and 0.7 for Swin-B, Swin-L, and SwinV2-H, respectively. Other fine-tuning hyper-parameters follow that in ablation.

Table 7 lists the results of our approach with different model sizes, compared to the supervised counterparts. With *SimMIM* pre-training, all of Swin-B, Swin-L, and SwinV2-H achieve significantly higher accuracy than their supervised counterparts. In addition, the SwinV2-H model with a larger resolution of 512^2 achieves 87.1% top-1 accuracy on

Figure 4. Recovered images using three different mask types (from left to right): random masking, masking most parts of a major object, and masking the full major object.

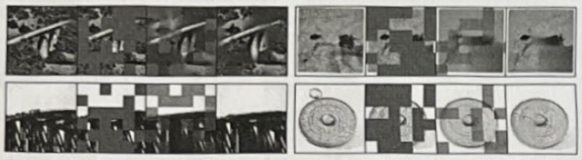

Figure 5. Recovered images by two different losses of predicting only the masked area or reconstructing all image area, respectively. For each batch, images from left to right are raw image, masked image, prediction of *masked patches only*, and reconstruction of *all patches*, respectively.

ImageNet-1K, which is the highest number among methods that use ImageNet-1K data only.

While all previous billion-level vision models rely on Google's JFT-3B dataset for model training [11, 44, 58], the proposed *SimMIM* approach is used to aid the training of a 3B SwinV2-G model [33] by using ~40× smaller data than that of JFT-3B. It achieves strong performance on four representative vision benchmarks: 84.0% top-1 accuracy on ImageNet-V2 classification [42], 63.1/54.4 box/mask mAP on COCO object detection [6, 31], 59.9 mIoU on ADE20K semantic segmentation [52, 63], and 86.8% top-1 acc on Kinetics-400 action recognition [28, 35]. More details are described in [33].

4.4. Visualization

In this section, we attempt to understand the proposed approach as well as some critical designs through visualizations. All example images are from the ImageNet-1K validation set.

What capability is learned? Figure 4 shows the recovered images with several human-designed masks, to understand what capability is learnt through masked image modeling. The human-designed masks (from left to right) consist of a random mask, a mask to remove most parts of a major object, and a mask to remove all of the major object, respectively. We can draw the following observations: 1) by random masking moderate parts of the major object, both

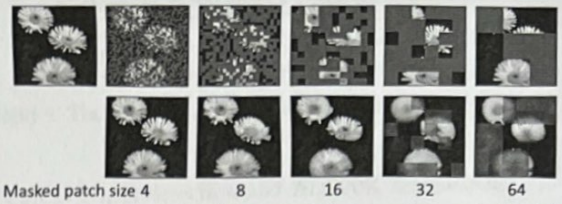

Figure 6. An example of recovered image using masked patch sizes of 4, 8, 16, 32 and 64, and a fixed masked ratio of 0.6.

the shape and texture of masked parts can be well recovered, as shown by the penguin, the mountain, the sailboat, and the persons. On the unmasked area, there is a severe checkerboard artifact due to that the recovery of unmasked area is not learnt during training; 2) by masking most parts of a major object (larger than 90%), the model can still predict an existence of object by the negligible clues; 3) when the objects are fully masked out, the masked area will be inpainted with background textures.

These observations indicate that the approach has learnt strong reasoning ability of objects, and the ability is not due to memorization of image identities or the simple copying of nearby pixels.

Prediction v.s. reconstruction We have shown the comparison of the representations learnt by a masked prediction task (our approach), and a joint masked prediction and visible signal reconstruction task in Table 4, which reveals that the pure masked prediction task performs significantly better. Figure 5 compares the recovery effects by two approaches. It shows that the latter approach makes better looking, however, probably the model capacity is wasted at the recovery of the unmasked area which may not be that useful for fine-tuning.

Effects of masked patch size Figure 6 shows the recovery of an image with different masked patch size under a fixed masking ratio of 0.6. It can be seen that the details can be much better recovered when the masked patch size is smaller, however, the learnt representations transfer worse. Probably, with smaller patch size, the prediction task can be easily accomplished by close-by pixels or textures.

5. Conclusion

This paper presents a simple yet effective self-supervised learning framework, SimMIM, to leverage masked image modeling for representation learning. This framework is made as simple as possible: 1) a random masking strategy with a moderately large masked patch size; 2) predicting raw pixels of RGB values by direct regression task; 3) the prediction head can be as light as a linear layer. We hope our strong results as well as the simple framework can facilitate future study of this line, and encourage in-depth interaction between AI fields.