Good Helper Is around You: Attention-Driven Masked Image Modeling

Zhengqi Liu[1], **Jie Gui**[*1,2], **Hao Luo**[3]

[1] Southeast University, Nanjing, China
[2] Purple Mountain Laboratories, China
[3] Alibaba group, China
{lzq_oscar, guijie}@seu.edu.cn, michuan.lh@alibaba-inc.com

Abstract

It has been witnessed that masked image modeling (MIM) has shown a huge potential in self-supervised learning in the past year. Benefiting from the universal backbone vision transformer, MIM learns self-supervised visual representations through masking a part of patches of the image while attempting to recover the missing pixels. Most previous works mask patches of the image randomly, which underutilizes the semantic information that is beneficial to visual representation learning. On the other hand, due to the large size of the backbone, most previous works have to spend much time on pre-training. In this paper, we propose **Attention-driven Masking and Throwing Strategy** (AMT), which could solve both problems above. We first leverage the self-attention mechanism to obtain the semantic information of the image during the training process automatically without using any supervised methods. Masking strategy can be guided by that information to mask areas selectively, which is helpful for representation learning. Moreover, a redundant patch throwing strategy is proposed, which makes learning more efficient. As a plug-and-play module for masked image modeling, AMT improves the linear probing accuracy of MAE by $2.9\% \sim 5.9\%$ on CIFAR-10/100, STL-10, Tiny ImageNet, and ImageNet-1K, and obtains an improved performance with respect to fine-tuning accuracy of MAE and SimMIM. Moreover, this design also achieves superior performance on downstream detection and segmentation tasks.

Introduction

Due to the superior ability to learn the representation of a large number of unlabeled images, self-supervised learning (SSL) has attracted much attention in the computer vision community. Inspired by masked language modeling (MLM) in the language domain (Brown et al. 2020; Devlin et al. 2019), masked image modeling (MIM) has demonstrated its significant potential in several vision tasks (*e.g.*, classification, object detection, and segmentation) (Lin et al. 2014; Gupta, Dollar, and Girshick 2019). There are state-of-the-art works using MIM that have emerged in the past year, including BEiT (Bao, Dong, and Wei 2022), MAE (He et al. 2022), SimMIM (Xie et al. 2022), and MaskFeat (Wei et al. 2022). These methods recover masked patches of images to provide the self-supervised signal, which has achieved a promising performance. With proper masking strategy, MIM shows the powerful capability of learning general visual representations.

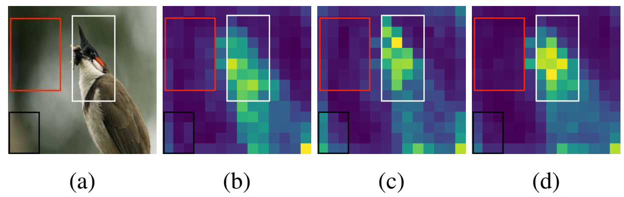

Figure 1: The motivation of AMT. (a) is the original image. (b)-(d) are visualizations for the self-attention of the `[CLS]` token on the heads of the last layer following DINO, which denotes the results of different masking and throwing schemes based on MAE. (b) by random masking strategy, (c) by attention-driven masking strategy, and (d) by attention-driven masking and throwing strategy. (d) has the most focused attention on the object and the least attention on the background.

Masking strategy is so important for MIM that many works attempt to explore different masking strategies to get a better performance. Several ways are tried to mask a portion of the image (He et al. 2022; Wei et al. 2022), including random masking, block-wise masking, and grid-wise masking. SimMIM applies diverse masked patch sizes and aims at finding the most appropriate size for masking. Besides the way of masking, most approaches mentioned above have studied the impact of the masking ratio. Impressively, MIM methods get excellent performance with high masking ratio (*e.g.*, 75% for MAE, 60% for SimMIM).

Though random masking could serve as a strong strategy for MIM, several obvious problems are required to resolve. As shown in Fig. 1b, random masking is prone to make attention dispersed on the whole image and not sufficiently focused on the object. This makes the model waste attention on meaningless background and learn weaker representations consequently. In addition, MIM usually needs numerous computing resources to pretrain, which is unacceptable to consume such resources in certain conditions.

Aiming to address both issues above, we propose **Attention-driven Masking and Throwing Strategy**

*Corresponding author.

Copyright © 2023, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

(AMT) in this paper. Firstly, we simulate Contrastive-Crop (Peng et al. 2022) and CAST (Selvaraju et al. 2021) to get saliency map as a constraint for masking in an absolutely unsupervised manner, which is called semantic information extraction. Then, a strategy of masking the areas with large weights in the saliency map is designed to leverage the semantic information, as shown in Fig. 1c. Although this way could make the model pay more attention to the portions that contain prominent features of the object, two significant problems still exist. First, certain areas of the background also cost much attention, such as the corner of the image; Second, a part of the object gets little attention (body of the bird). To alleviate these problems, we propose a scheme of throwing parts of the image, which dedicates to deleting portions of the image thoroughly. That means the thrown portions are not input into the model and thus reduces the computing cost. Fig. 1d indicates that such throwing scheme not only lets the model consistently pay more attention to the area of the object, but also reduces the attention on the background substantially.

It is worth noting that the proposed AMT is a plug-and-play module for MIM frameworks and could be readily deployed into MIM methods, such as MAE and SimMIM. The semantic information can be obtained with the self-attention mechanism. Thanks to the proposed throwing strategy, our method can speed up the pre-training process significantly. Furthermore, the performance of our approach surpasses that of the original MAE and SimMIM. In specific, our strategy improves the linear probing accuracy of MAE by $2.9\% \sim 5.9\%$ on CIFAR-10, CIFAR-100, STL-10, Tiny ImageNet, ImageNet-1K and fine-tuning accuracy of MAE and SimMIM by $0.2\% \sim 5.8\%$. Outstanding results are also obtained on popular downstream detection and segmentation tasks, including COCO and LVIS.

The main contributions of this paper can be summarized as follows.

- We get the saliency map of the image using the general self-attention mechanism in an unsupervised manner. Such operation could be easily integrated to MIM methods without extra components.
- In AMT, a masking and throwing strategy is proposed based on saliency maps to adjust the attention of the model, accelerating the pre-training process while improving the performance .
- As a plug-and-play module, AMT outperforms original MIM methods on a variety of datasets, showing it has more powerful transferability with less computing cost than random masking strategy.

Related Work

We introduce relevant prior works including self-supervised learning and masking strategy in this section.

Self-supervised Learning

Contrastive learning. For a long time, the main-stream self-supervised learning mode is contrastive learning, whose core objective is to pull positives close and push negatives apart (He et al. 2020; Chen et al. 2020c; Chen, Xie, and He 2021; Chen et al. 2020b; Grill et al. 2020; Caron et al. 2021). One of the most important challenges of contrastive learning is generating reasonable image pairs from datasets. Many previous works explore the ways for that (Zhao et al. 2021; Dwibedi et al. 2021). One of the solutions is to generate saliency maps for images that are used to guide the generation of positives or negatives (Peng et al. 2022; Selvaraju et al. 2021). We simulate these methods to construct a saliency map containing semantic information with the help of the self-attention mechanism.

Masked language modeling (MLM). Transformers have been widely applied to Natural Language Processing (NLP), which have gained great success for pre-training, *e.g.*, BERT (Devlin et al. 2019) and GPT (Brown et al. 2020). These methods predict invisible contents by limited visible parts of the input sequence, which are called masked language modeling (MLM). By pre-training on huge data, such methods have been proved to be scalable on various downstream tasks, showing that MLM has a strong generalization ability.

Masked image modeling (MIM). Almost simultaneously, masked image modeling is proposed (Chen et al. 2020a; Doersch, Gupta, and Efros 2015; Hénaff et al. 2020; Pathak et al. 2016; Trinh, Luong, and Le 2019) and influential works are emerging in the past year. As a pioneer work in this direction, context encoders (Pathak et al. 2016) predict missing pixels in designated areas. Recently, with the popularity of transformers (Vaswani et al. 2017; Liu et al. 2021, 2022; Dosovitskiy et al. 2021; Touvron et al. 2021b,a), MIM comes back into the spotlight. iGPT (Chen et al. 2020a) and ViT (Dosovitskiy et al. 2021) propose particular schemes to utilize the transformer to process images. Subsequently, distillation-based MIM also appears (Zhou et al. 2022). The milestone work BEiT (Bao, Dong, and Wei 2022) uses a trained dVAE network to construct a challenging task that predicts the visual tokens of masked image patches. Another momentous work MAE (He et al. 2022) applies autoencoder into MIM, which learns representations by the encoder and reconstructs original pixels of masked image patches through the decoder. Different from MAE, SimMIM (Xie et al. 2022) and MaskFeat (Wei et al. 2022) employ a linear head instead of a transformer decoder; MaskFeat substitutes original pixels with HOG (Dalal and Triggs 2005) features as the target of reconstruction.

Masking Strategy

Random masking. MIM depends on predicting masked image patches to learn representations, showing masking strategy plays a significant role in MIM. BEiT employs a block-wise random masking strategy, which is prone to mask a block of patches rather than discrete patches. Block-wise masking has been already applied in (Wei et al. 2022). MAE randomly masks a large number of patches and the masked patch size is the same as the input patch size (16×16) of ViT. Moreover, SimMIM studies the impact of different masked patch sizes and chooses a larger size (32×32).

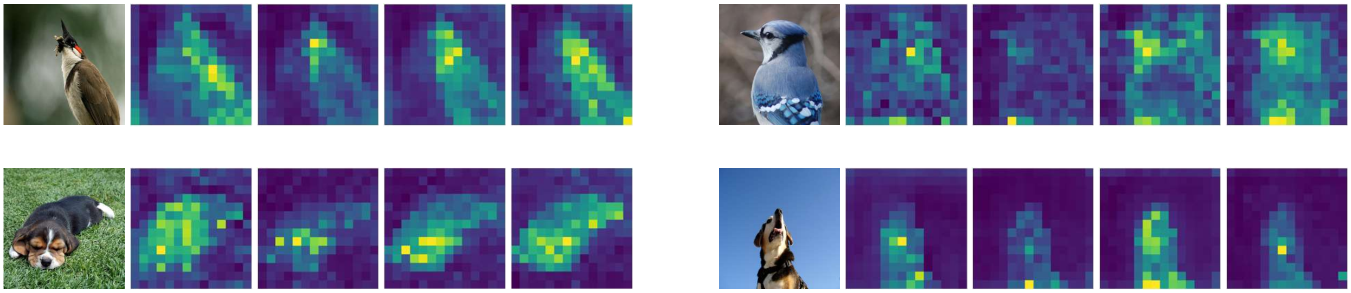

Figure 2: Visualization of attention. For each subfigure, there is the original image, and the attention map from the last layer of the MAE encoder at different training stages (*40th,60th,80th,100th*), respectively.

Selective masking. Other than those random masking strategies, several works explore the schemes of selective masking recently. MST (Li et al. 2021) proposes to mask low-attended patches, showing good performance without additional cost. Afterward, AttMask (Kakogeorgiou et al. 2022) further studies the results of masking diverse attended patches, demonstrating the usefulness of masking highly-attended portions. However, these methods are only applied to a specific distillation-based model. By comparison, our AMT is a plug-and-play module that can be readily applied to popular MIM methods, such as MAE, SimMIM and MaskFeat. SemMAE (Li et al. 2022a) also introduces a semantic-guided masking strategy. However, their approach proposes a particular framework which uses an additional pretrained model to extract features and employs the features in a complex way. This increases the cost of computing resources. In contrast to that, our AMT is a simple and framework-agnostic module, which obtains the semantic information in a fully unsupervised manner without any extra design.

Apart from the masking strategy, we also propose a throwing strategy in AMT, bringing better performance while reducing the computing cost substantially.

Method

In this section, *attention-driven masking and throwing strategy* (AMT) for MIM is introduced. Firstly, we discuss preliminaries about vision transformer and MIM. Then, AMT is described detailly, which includes the acquisition of semantic information, masking scheme, and throwing strategy.

Preliminary

Revisiting Vision Transformer. Here we introduce the common backbone ViT (Dosovitskiy et al. 2021), which regards an image as 16×16 words. Specifically, given an input image $x \in \mathbb{R}^{H \times W \times C}$, firstly it is reshaped to $x_p \in \mathbb{R}^{N \times (P^2 \cdot C)}$ in order to turn 2D images into a 1D sequence of token embeddings, where (H, W) denotes the height and width of the original image, C is the number of channels, (P, P) is the size of each image patch and $N = HW/P^2$ is the number of patches. Afterward, the patches are flattened and projected to dimensionality D through a linear projection $E \in \mathbb{R}^{(P^2 \cdot C) \times D}$ and a `[CLS]` token x_{cls} is appended. Finally, position embedding $E_{pos} \in \mathbb{R}^{(N+1) \times D}$ is added to form input tokenized image :

$$z = [x_{cls}; x_p^1 E; x_p^2 E; ...; x_p^N E] + E_{pos}, \tag{1}$$

where x_p^m denotes the row m of x_p, and $z \in \mathbb{R}^{(N+1) \times D}$. Such tokens could be used as the input for ViT.

Revisiting masked image modeling. Thanks to the patch-level image processing for ViT, we can separately handle each image patch. This makes masking a part of patches possible. As shown in Fig. 3, we denote popular MIM methods (*e.g.* , MAE, SimMIM, MaskFeat) as *original MIM methods*. MIM methods update the weights of the model by predicting the masked parts of the image. l_1 and l_2 losses are often employed

$$L = \| o_G - y_G \|_{p=1 \ or \ 2}, \tag{2}$$

where $o, y \in \mathbb{R}^{CHW}$ are the predicted values and the features to predict (RGB pixels or other features *e.g.* HOG features), respectively. G denotes the loss calculated on pixel level and loss is only calculated on masked patches.

Attention-driven Masking and Throwing

Semantic information extraction. After the image is embedded into sequence of tokens $z \in \mathbb{R}^{(N+1) \times D}$ as Eq. 1, these tokens are put into the transformer block. Each block utilizes a *multi-head self-attention* (MSA) layer to project and divide z into N_h parts. Each part contains the query q_i, key k_i and value v_i for $i = 1, 2, ..., N_h$, where N_h denotes the number of heads, $q_i, k_i, v_i \in \mathbb{R}^{(N+1) \times d}$. Then the MSA can be obtained with softmax:

$$A_i = softmax(q_i k_i / \sqrt{D_h}), \tag{3}$$

where $D_h = D/N_h$, A_i is a $(N+1) \times (N+1)$ attention matrix. Then we average the first row but the first element of A_i over N_h heads:

$$a_w = \frac{1}{N_h} \sum_{i=1}^{N_h} a_i^1, \tag{4}$$

where $a_i^1 \in \mathbb{R}^N$ is the first row of A_i without the first element. Afterward, a_w, which is called masking weights, could be reshaped to $(H/P) \times (W/P)$. Then an interpolation could map it to the original image size. It could be

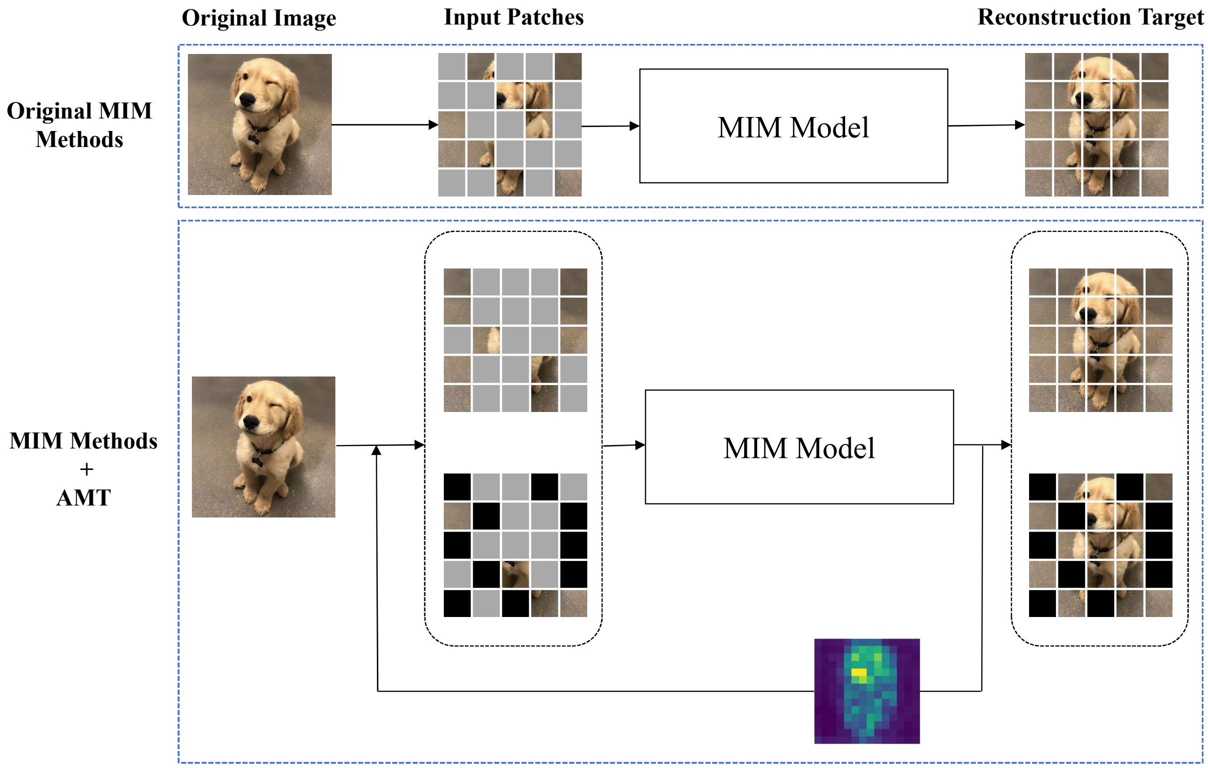

Figure 3: Overview of common MIM methods and AMT. The top of the figure denotes the simplified common MIM methods, the bottom is the simplified overview of our AMT. The gray patches are masked patches. The black patches denote thrown tokens which are not input into the model, meaning that thrown tokens do not cost computational resources.

visualized as shown in Fig. 2. Visibly, even if at the early stage of pre-training, the model could catch the semantic information roughly.

During the pre-training process, a_w is updated at a regular interval (40 epochs). Note that the precise location of the object is unnecessary, since rough location is enough to provide semantic information to guide masking and throwing. Such forward step brings trivial computing cost which could be overlooked (about 1% of the whole pre-training time).

Masking and Throwing. Fig. 3 indicates that original MIM methods apply random masking which gives image patches the same opportunity to be masked. Such operation makes model prone to disperse its attention on the whole image. As shown in Fig. 4, original methods still have quite a few attention on the background, which damages representation learning. On the other hand, pre-training often costs much time due to the large size of the backbone and huge data.

To address these problems, we propose to mask and throw image patches intentionally. Specifically, we use random masking to collect semantic information from the whole image at the early stage of pre-training. After a few epochs, we use the method in this section to obtain attention map. Then we normalize it to $[0, 1]$ and each element in the attention map denotes the weights of the corresponding pixel in the image. When the image is cropped for augmentation before it is input into the model, the corresponding area in the attention map is also cropped. The cropped attention map is tokenized to patches along with the cropped image together. To maintain enough semantic information for MIM, we sample the index of patches N times:

$$M = \mathbb{S}(P_A), \tag{5}$$

where $\mathbb{S}$ is the function that samples according to weights. Each element in $P_A \in \mathbb{R}^N$ denotes the sample weights of the corresponding token in the tokenized image z except for the x_{cls} token. P_A is obtained by summarizing a_w in patch. Finally, $M \in \mathbb{R}^N$ is the result vector of sampling. The element in M denotes the index of patch.

This sampling strategy is prone to rank highly-attended patches at the top and low-attended patches at the bottom. Our attention-driven masking is to mask the top of M and keep a few parts of the object visible, as shown in Fig. 3 *masking only*. This generates a more challenging reconstruction task. Notably, we do not mask highly-attended areas directly, but make the highly-attended patches more prone to be masked. That means highly-attended still has small probability to be visible. Fig. 4 shows that with *attention-driven masking* only, attention is more focused on minor areas that contain salient features. At the same time, it overlooks several significant parts of the project (*e.g.*, the body of the panda, the head of the mushroom, *etc.*) As a result, such attention leads to a lack of ability of the model to learn

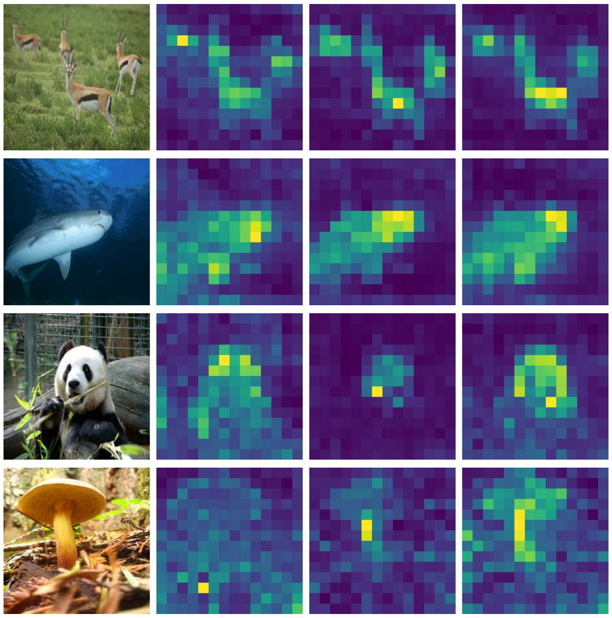

Figure 4: Visualization of the attention map of the last layer in the encoder after 400 epochs pre-training. From left to right, there is the original image, the attention map from the last layer of the MAE encoder using random masking, attention-driven masking, and AMT, respectively.

representations.

To solve this problem, we propose AMT to guide masking and throwing. Since we could roughly estimate the location of the object from the attention map, parts of the background are superfluous for training. Based on this view, we throw a certain amount of tokens in the middle of M. The top tokens of M are masked and the bottom tokens of M are visible:

$$z_I = \mathbb{C}(M, r, t), \tag{6}$$

where $\mathbb{C}$ is *attention-driven masking and throwing strategy* (AMT), t is the throwing ratio of tokens, r is the masking ratio, and z_I denotes the final tokens that includes masked parts and visible parts. From Fig. 4, AMT could strengthen the attention on the object, as well as decrease the attention on the background obviously compared to original methods. Besides, AMT brings a smoother transition between salient and common features (*e.g*, head and body). Thanks to the throwing strategy, the input data is prominently reduced and the pre-training is faster. Algorithm 1 shows the pipeline of AMT. Our AMT is based on the input patched image which is widely used in MIM, so that AMT could be readily integrated into other popular MIM methods. We also study the influence of throwing different areas in Section .

Algorithm 1: Algorithm of AMT for MIM

Input: Image *i*, Masking ratio *r*, Throwing ratio *t*, Height of Image *H*, Width of Image *W*, Patch size *P*

$N = (H \times W)/P^2$ ▷ The number of patches
$mask\_c = N \times r$ ▷ The count of masked tokens
$throw\_c = N \times t$ ▷ The count of thrown tokens
$a_w = Forward(I)$ ▷ Attention map of last layer
$P_A = Normalize(a_w)$ ▷ Sampling weights
$M = \mathbb{S}(P_A)$ ▷ Sampling by Eq. 5
$mask\_idx = M[: mask\_c]$
$throw\_idx = M[mask\_c : mask\_c + throw\_c]$
▷ The index of masked and thrown tokens

Output: $mask\_idx, throw\_idx$

Experiments

Setup

Datasets and Baseline Approaches. We evaluate our method with popular MIM methods (MAE, SimMIM) with linear probing and fine-tuning on ImageNet-1K validation set. We validate the transferability of our method on other downstream tasks. We test the classification accuracy on **CIFAR-10/100** (Krizhevsky 2009), **Tiny ImageNet**, **STL-10** (Coates, Ng, and Lee 2011), and **ImageNet-1K** (Deng et al. 2009) by linear probing and fine-tuning. For object detection and segmentation, we finetune on **COCO** (Lin et al. 2014) and **LVIS** (Gupta, Dollar, and Girshick 2019). Ablations about the ratio and the part of masking and throwing are also provided.

Implementation Details. As a plug-and-play module for MIM methods, what AMT depends on is patch-level masking which is widely utilized in MIM methods to provide the self-supervised signal. Thus, AMT is agnostic to other training components, such as losses, optimizers and learning rate schedules. To compare with the original MIM methods fairly, we severely keep the same training setting as the baseline methods. Such comparison aims at exploring the gain for the performance of AMT.

We pretrain MAE and SimMIM on ImageNet-1K for 400 epochs and 200 epochs respectively. The settings of pre-training follow the original works strictly. We choose the ViT-B/16 as the backbone of the encoder. Typically for MAE, we follow official codes to choose ViT-B/16 with 8 blocks as decoder. For SimMIM, the decoder is a linear head. Moreover, absolute position embedding is used in all experiments. For linear probing, we train a revised linear head with batch normalization. Such linear probing is applied for evaluation on classification with MAE. For fine-tuning, the same settings as original works are applied. `[CLS]` token is used for linear probing and fine-tuning.

For our AMT, we choose the thrown ratio $t = 0.4$ and 0.26 for MAE and SimMIM respectively. To maintain the ratio between masked and visible tokens in original works, the masking ratio is respectively set as $r = 0.45$ and 0.44 for MAE and SimMIM. This eliminates the impact of different ratios between masked and visible tokens. The interval for updating masking weights a_w is 40 epochs (10% of the whole pre-training process of MAE, and 20% of SimMIM). The cost of such extra evaluation step is negligible, and our AMT accelerates training process with the help of throwing strategy. All experiments are conducted on a 4-GPU server.

Method	CIFAR-10		CIFAR-100		Tiny ImageNet		STL-10		ImageNet	
	Linear	Fine-tuning	Linear	Fine-tuning	Linear	Fine-tuning	Linear	Fine-tuning	Linear	Fine-tuning
MAE+Random M	85.2	96.5	65.2	87.4	55.2	76.5	80.9	96.5	56.6	82.6
MAE+AM (ours)	**89.4**	97.4	**69.9**	87.3	**59.9**	76.3	**87.1**	97.4	61.5	82.5
MAE+AMT (ours)	88.1	**97.5**	68.7	**87.8**	59.6	**77.8**	86.8	**97.5**	**61.7**	**82.8**
SimMIM+Random M	-	95.0	-	80.3	-	74.0	-	92.3	-	**81.5**
SimMIM+AM (ours)	-	**97.8**	-	85.9	-	**78.8**	-	**96.5**	-	**81.5**
SimMIM+AMT (ours)	-	97.7	-	**86.1**	-	75.8	-	**96.5**	-	80.7

Table 1: Top-1 accuracy on CIFAR-10/100, Tiny ImageNet, STL-10, and ImageNet. M denotes Masking. AM denotes attention-driven masking. Random Masking is the default masking strategy for MAE and SimMIM.

SimMIM	Ratio (%) Mask	Throw	Fine-tuning Top-1 Acc. (%)	Pre-training costs
+Random M	60	0	74.0	1.0×
+AM	60	0	78.8	∼1.0×
+AMT	44	26	75.8	∼0.76×
+AMT	33	50	75.1	∼0.62×

Table 2: The accuracy on Tiny ImageNet and pre-training costs using SimMIM with different ways of masking and throwing. M denotes masking. AM denotes attention-driven masking.

Classification

We pretrained MAE, SimMIM on ImageNet-1K with diverse masking and throwing strategies for 400 and 200 epochs, respectively. The linear probing and fine-tuning results on various datasets are shown in this section, including CIFAR-10/100, Tiny ImageNet, ImageNet-1K, and STL-10.

Linear probing. As shown in Tab. 1, we evaluate linear probing performance of MAE. Our AMT prominently improves the Top-1 accuracy by $2.9\% \sim 5.9\%$. This shows that our AMT greatly improves the linear separability of learned representations. Note that AMT only uses 60% of the image, which leads to faster pre-training.

Fine-tuning. Fine-tuning accuracy could reflect the strength of the learned non-linear features, which is important for downstream tasks. The results of fine-tuning are shown in Tab. 1. Our AMT improves the fine-tuning accuracy by $0.2\% \sim 5.8\%$, showing the representations learned by AMT has a stronger transferability than original method with random masking. Such capability is useful to the downstream task, which is the main goal of self-supervised pre-training. In addition, Fig. 5 shows both AMT and attention-driven masking achieve better performance than random masking and converges faster. Finally, AMT reaches the comparative performance with attention-driven masking. Besides, we experimentally find the choice of hypermeters for our AMT is more extensive than original methods when fine-tuning, which reduces the time for searching appropriate *lr*.

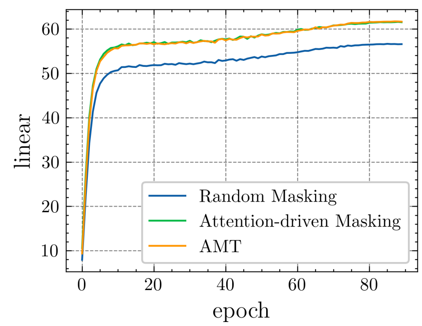

Figure 5: Top-1 linear probing accuracy on ImageNet-1K validation, using MAE with different masking and throwing strategy pretrained on ImageNet-1K.

Downstream Tasks

In this section, results on object detection and instance segmentation tasks are shown to further investigate the transferability of our method. We experiment with models pretrained on ImageNet-1K for 400 epochs. In particular, we perform object detection and instance segmentation on LVIS and COCO. Following ViTDet's detectron2 codebase (Li et al. 2022b; Wu et al. 2019), we employ the same setups with a total batch size of 16 and all experiments here use Mask R-CNN (He et al. 2017) as the detector with a backbone of ViT-B/16. To maintain a fair comparison, all hypermeters are the same in each experiment.

Comparisons on COCO. We fine-tune the models on `train2017` set for 90K iterations and evaluate on `val2017`. As Tab. 3 presented, compared to the original MAE, our method gains consistent improvements on every metric. AMT shows the best performance on metrics except for AP^{mask}. Because AMT is prone to focus attention on less areas than attention-driven masking, AMT achieves a better performance with high threshold. However, AMT has a more extensive attention on the object, which is useful for segmentation with a smaller range of threshold.

Comparisons on LVIS. The comparisons on LVIS dataset are reported to further study the transferability of our method. Different from COCO, LVIS has an imbalanced dis-

Method	*LVIS Instance Segmentation*						*COCO Instance Segmentation*					
	AP^{bbox}	AP^{bbox}_{50}	AP^{bbox}_{75}	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}	AP^{bbox}	AP^{bbox}_{50}	AP^{bbox}_{75}	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}
Random Init	14.6	24.7	15.3	14.3	23.2	15.0	28.1	46.1	29.8	26.2	43.5	27.6
MAE†	25.0	38.5	26.9	24.2	37.0	25.7	40.1	60.3	43.7	36.6	57.6	39.3
MAE+AM	26.2	40.2	28.3	25.4	38.4	27.1	42.2	62.6	46.3	**38.3**	59.7	41.4
MAE+AMT	**26.9**	**41.3**	**28.8**	**26.0**	**39.4**	**27.7**	**42.8**	**63.0**	**47.1**	36.6	**60.1**	**41.6**

Table 3: The results on COCO and LVIS. The models are pretrained on ImageNet-1K. AM denotes attention-driven masking. †: default MAE settings with random masking.

Method	Ratio (%) Mask	Ratio (%) Throw	*Attention-driven Masking*	*Throwing*	*Throwing middle Tokens*	*Acc. (%)*
MAE+Random M	75	0				52.3
MAE+Attention-driven M	75	0	✓			50.1
MAE+Random T	45	40		✓		50.8
MAE+AMT	75	10	✓	✓		51.7
MAE+AMT	45	40	✓	✓		52.6
MAE+AMT	45	40	✓	✓	✓	**53.3**

Table 4: Ablation of different masking and throwing strategies used in MAE. Each model is pretrained on ImageNet-1K for 200 epochs. M denotes Masking. T denotes Throwing. *Acc.* is the Top-1 linear probing accuracy on Tiny ImageNet.

tribution on different classes and the training examples of certain classes are even less than 10. The mask in LVIS is more concise and consistent. These attributes above make detection and segmentation more challenging. We fine-tune models on `train` set for 75K iterations and evaluate on `val` set. As shown in Tab. 3, our AMT improves all metrics on this dataset by at least 1.4%. Compared to attention-driven masking, AMT still has a a significant performance improvement, showing the superiority of AMT.

Ablations

In this section, we pretrain MAE with 6 different masking and throwing strategies to explore the effective components in AMT. Moreover, we show the influence of AMT on the computing cost of pre-training.

Different ways of masking and throwing. In our method, we propose to throw several portions of tokens to strengthen representation learning using the attention map, which reduces computing cost at the same time. As shown in Tab. 4, we present the results of classification accuracy on Tiny ImageNet with 6 different masking and throwing schemes. AMT with 40% throwing ratio and 45% masking ratio shows the best performance, which improves the baseline MAE by 1.0%. MAE with attention-driven masking only and throwing comparatively small portions (10%) of tokens both result in a decline for accuracy. Intuitively, we think attention-driven masking is slower to take effect than AMT. To validate the effectiveness of attention-driven throwing strategy, we experiment MAE with random throwing which randomly throws 40% tokens of the image and masks 45% tokens. That approach has a bad performance, showing the effect of attention-driven throwing. Moreover, in order to study the impact of throwing different areas, we set experiments with throwing low-attended and medium-attended area, respectively. The result shows throwing medium-attended tokens strategy that is the default of our AMT outperforms throwing low-attended parts by 0.7%.

Computing cost. We experiment SimMIM with diverse masking and throwing strategies on Tiny ImageNet. Tab. 2 shows the results. All models are pretrained on ImageNet-1K for 200 epochs. Our attention-driven masking and throwing strategy could obviously improve the performance with less computing cost. When we throw 50% of the image, we accelerates the pre-training $1.6\times$ and the performance is still better than original SimMIM with random masking.

Conclusion

AMT leverages the self-attention in ViT for masking and throwing portions of the image, which utilizes the semantic information obtained by the model itself during the training process. Thus, AMT could make the model focus on the object and overlook the background, resulting in better performance as well as less computing cost. Notably, as a plug-and-play module for masked image modeling, AMT can be easily applied to MIM methods which use ViT as the backbone. We demonstrate that by achieving AMT in typical MIM methods, including MAE and SimMIM. We extensively experiment on a variety of downstream datasets and get superior results, which reveals the excellent transferability of the learned representation. We hope our work could inspire more researches.

Acknowledgments

This work is supported by the NSFC Grant No. 62172090; the CAAI-Huawei MindSpore Open Fund; the Alibaba Group through Alibaba Innovative Research Program. We thank the Big Data Computing Center of Southeast University for providing the facility support on the numerical calculations in this paper.

References

Bao, H.; Dong, L.; and Wei, F. 2022. BEiT: BERT Pre-Training of Image Transformers. In *ICLR*.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. In *NeurIPS*, 1877–1901.

Caron, M.; Touvron, H.; Misra, I.; Jégou, H.; Mairal, J.; Bojanowski, P.; and Joulin, A. 2021. Emerging Properties in Self-Supervised Vision Transformers. In *ICCV*, 9650–9660.

Chen, M.; Radford, A.; Child, R.; Wu, J.; Jun, H.; Luan, D.; and Sutskever, I. 2020a. Generative pretraining from pixels. In *ICML*, 1691–1703.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020b. A simple framework for contrastive learning of visual representations. In *ICML*, 1597–1607.

Chen, X.; Fan, H.; Girshick, R.; and He, K. 2020c. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*.

Chen, X.; Xie, S.; and He, K. 2021. An empirical study of training self-supervised vision transformers. *arXiv preprint arXiv:2104.02057*.

Coates, A.; Ng, A.; and Lee, H. 2011. An Analysis of Single Layer Networks in Unsupervised Feature Learning. In *AISTATS*.

Dalal, N.; and Triggs, B. 2005. Histograms of oriented gradients for human detection. In *CVPR*, 886–893 vol. 1.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*, 248–255.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 4171–4186.

Doersch, C.; Gupta, A.; and Efros, A. A. 2015. Unsupervised visual representation learning by context prediction. In *ICCV*, 1422–1430.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*.

Dwibedi, D.; Aytar, Y.; Tompson, J.; Sermanet, P.; and Zisserman, A. 2021. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In *ICCV*, 9588–9597.

Grill, J.-B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P.; Buchatskaya, E.; Doersch, C.; Avila Pires, B.; Guo, Z.; Gheshlaghi Azar, M.; et al. 2020. Bootstrap Your Own Latent-A New Approach to Self-Supervised Learning. In *NeurIPS*, 21271–21284.

Gupta, A.; Dollar, P.; and Girshick, R. 2019. LVIS: A Dataset for Large Vocabulary Instance Segmentation. In *CVPR*, 5356–5364.

He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; and Girshick, R. 2022. Masked Autoencoders Are Scalable Vision Learners. In *CVPR*, 16000–16009.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 9729–9738.

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask r-cnn. In *ICCV*, 2961–2969.

Hénaff, O. J.; Razavi, A.; Doersch, C.; Eslami, S.; and Oord, A. v. d. 2020. Data-efficient image recognition with contrastive predictive coding. In *ICML*, 4182–4192.

Kakogeorgiou, I.; Gidaris, S.; Psomas, B.; Avrithis, Y.; Bursuc, A.; Karantzalos, K.; and Komodakis, N. 2022. What to Hide from Your Students: Attention-Guided Masked Image Modeling. *arXiv preprint arXiv:2203.12719*.

Krizhevsky, A. 2009. Learning Multiple Layers of Features from Tiny Images. *Master's thesis, University of Tront*.

Li, G.; Zheng, H.; Liu, D.; Su, B.; and Zheng, C. 2022a. SemMAE: Semantic-Guided Masking for Learning Masked Autoencoders. *arXiv preprint arXiv:2206.10207*.

Li, Y.; Mao, H.; Girshick, R.; and He, K. 2022b. Exploring plain vision transformer backbones for object detection. *arXiv preprint arXiv:2203.16527*.

Li, Z.; Chen, Z.; Yang, F.; Li, W.; Zhu, Y.; Zhao, C.; Deng, R.; Wu, L.; Zhao, R.; Tang, M.; et al. 2021. MST: Masked Self-Supervised Transformer for Visual Representation. In *NeurIPS*, 13165–13176.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *ECCV*, 740–755.

Liu, Z.; Hu, H.; Lin, Y.; Yao, Z.; Xie, Z.; Wei, Y.; Ning, J.; Cao, Y.; Zhang, Z.; Dong, L.; Wei, F.; and Guo, B. 2022. Swin Transformer V2: Scaling Up Capacity and Resolution. In *CVPR*, 12009–12019.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. In *ICCV*, 10012–10022.

Pathak, D.; Krahenbuhl, P.; Donahue, J.; Darrell, T.; and Efros, A. A. 2016. Context encoders: Feature learning by inpainting. In *CVPR*, 2536–2544.

Peng, X.; Wang, K.; Zhu, Z.; Wang, M.; and You, Y. 2022. Crafting Better Contrastive Views for Siamese Representation Learning. In *CVPR*, 16031–16040.

Selvaraju, R. R.; Desai, K.; Johnson, J.; and Naik, N. 2021. Casting your model: Learning to localize improves self-supervised representations. In *CVPR*, 11058–11067.

Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; and Jégou, H. 2021a. Training data-efficient image transformers & distillation through attention. In *ICML*, 10347–10357.

Touvron, H.; Cord, M.; Sablayrolles, A.; Synnaeve, G.; and Jégou, H. 2021b. Going Deeper With Image Transformers. In *ICCV*, 32–42.

Trinh, T. H.; Luong, M.-T.; and Le, Q. V. 2019. Selfie: Self-supervised pretraining for image embedding. *arXiv preprint arXiv:1906.02940*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NeurIPS*.

Wei, C.; Fan, H.; Xie, S.; Wu, C.-Y.; Yuille, A.; and Feichtenhofer, C. 2022. Masked Feature Prediction for Self-Supervised Visual Pre-Training. In *CVPR*, 14668–14678.

Wu, Y.; Kirillov, A.; Massa, F.; Lo, W.-Y.; and Girshick, R. 2019. Detectron2. https://github.com/facebookresearch/detectron2. Accessed: 2019-10-11.

Xie, Z.; Zhang, Z.; Cao, Y.; Lin, Y.; Bao, J.; Yao, Z.; Dai, Q.; and Hu, H. 2022. SimMIM: A Simple Framework for Masked Image Modeling. In *CVPR*, 9653–9663.

Zhao, N.; Wu, Z.; Lau, R. W.; and Lin, S. 2021. Distilling Localization for Self-Supervised Representation Learning. In *AAAI*, 10990–10998.

Zhou, J.; Wei, C.; Wang, H.; Shen, W.; Xie, C.; Yuille, A.; and Kong, T. 2022. iBOT: Image BERT Pre-Training with Online Tokenizer. In *ICLR*.