# Evolved Part Masking for Self-Supervised Learning

Zhanzhou Feng[1] Shiliang Zhang[1,2]
[1]National Key Laboratory for Multimedia Information Processing,
School of Computer Science, Peking University
[2]Peng Cheng Laboratory
fengzz@stu.pku.edu.cn, slzhang.jdl@pku.edu.cn

## Abstract

*Existing Masked Image Modeling methods apply fixed mask patterns to guide the self-supervised training. As those patterns resort to different criteria to mask local regions, sticking to a fixed pattern leads to limited vision cues modeling capability. This paper proposes an evolved part-based masking to pursue more general visual cues modeling in self-supervised learning. Our method is based on an adaptive part partition module, which leverages the vision model being trained to construct a part graph, and partitions parts with graph cut. The accuracy of partitioned parts is on par with the capability of the pre-trained model, leading to evolved mask patterns at different training stages. It generates simple patterns at the initial training stage to learn low-level visual cues, which hence evolves to eliminate accurate object parts to reinforce the learning of object semantics and contexts. Our method does not require extra pre-trained models or annotations, and effectively ensures the training efficiency by evolving the training difficulty. Experiment results show that it substantially boosts the performance on various tasks including image classification, object detection, and semantic segmentation. For example, it outperforms the recent MAE by 0.69% on imageNet-1K classification and 1.61% on ADE20K segmentation with the same training epochs.*

## 1. Introduction

Recent years have witnessed a boom in continuously growing representation learning capability and data demands of deep neural networks like CNN [21, 37] and vision transformers [14, 27, 33]. To tackle the increasing demand for labelled data, Masked Language Modeling (MLM) [3, 13] has been adopted to train natural language processing models through self-supervised learning on large-scale data. Inspired by the success of MLM, many works propose Masked Image Modeling (MIM) to pre-train vision models on unlabeled images for a series of downstream tasks [2,18,38]. MLM masks several words in the input sentences and supervises the network to recover masked words according to semantics provided by remaining words. MIM follows a similar idea of MLM to mask a portion of regions in input images, then trains the vision model to recover masked contents from visible regions. As images are not structured representations like sentences, different MIM works have to resort to different criteria to generate mask patterns.

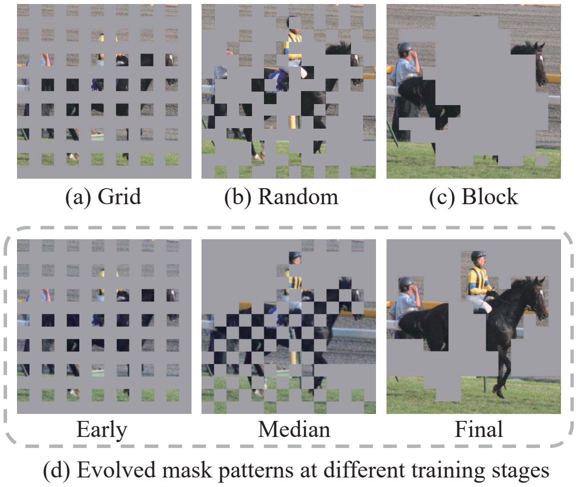

Figure 1. (a), (b), and (c) are three basic mask patterns adopted in existing MIM methods. (d) illustrates the proposed evolved part masking, where the generated mask patterns evolve with the capability of vision model being trained.

Mask patterns in existing works can be divided into three categories according to their masked image cues. Some works like MAE [18] and SimMIM [38] do not differentiate visual cues in images, and randomly mask local regions or patches. Another line of the works, such as MST [24], propose to preserve crucial cues in the image to enhance the learning of local context. The third line of works such as AttnMask [22] and SemMAE [23] propose to completely mask cues like object region in images to pose a more chal-

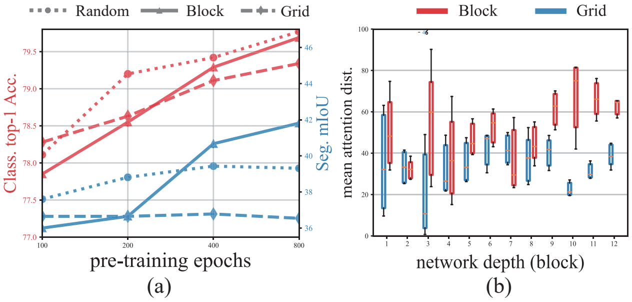

Figure 2. Illustration of effects of different mask patterns to down-stream tasks in (a), and learned parameters in (b). In (a), random pattern and block pattern perform best in image classification and semantic segmentation, respectively. (b) shows the mean attention distance across images at different layers of the pre-trained model. Results indicate different mask pattern are suited to different tasks.

lenging pretext task. A more detailed review to existing works will be presented in Sec. 2.

Mask patterns in those works lead to different visual cues modeling tasks and varied difficulties. To study the impact of mask patterns on self-supervised pre-training, we adopt three basic masking methods in Fig. 1(a)-(c) to different vision tasks. Fig. 2(a) explores their effects to two vision tasks. It can be observed that, random pattern and block pattern perform best in image classification and semantic segmentation, respectively. It is also clear that, more training epochs do not boost the performance of grid pattern and random pattern in segmentation. Fig. 1(b) further visualizes the average attention length of neurons at each layer of the pre-trained model. It indicates that, neurons trained by grid mask mostly focus on nearby regions with shorter attention distances. As longer attention distance benefits the learning of contextual cues, block pattern is more preferred by dense prediction tasks like semantic segmentation.

Fig. 2 indicates that, the criteria for generating mask patterns largely determines visual cues that the network could learn in the pre-training phase. For instance, masking the complete object regions is more beneficial for learning semantics and contexts than grid mask. Masking grid pattern makes the network neuron pay more attention to nearby regions, and favors the initial training stage in classification, by posting an easier learning task. Therefore, different mask patterns are suited to different down-stream tasks. This finding leads to one fundamental challenge to self-supervised learning: the pre-training procedure have no clue which task it will be applied to.

Instead of sticking to a fixed mask pattern, we propose the evolved part masking to pursue more general visual cues modeling capability in self-supervised learning. The evolved part masking is expected to model visual cues at different scales, accelerate the training convergence. To this end, we generate masks by partitioning object parts in training images. An adaptive part partition module is adopted to leverage the vision model being trained to construct a part graph, and partition parts with graph cut. The accuracy of partitioned parts is on par with the capability of vision model, leading to evolved mask patterns at different training stages, as illustrated in Fig. 1(d). In other words, the initial training stage generates simple patterns to learn low-level visual cues, which hence evolves to mask different object parts to reinforce the learning of object semantics and contexts.

The adaptive part partition module generates parts according to the relationship among image patches inferred by the vision model. The relevance among patches learned by the vision transformers are encoded in the attention map. Our method hence constructs a patch association graph based on attention maps, and tackle the unlabeled part partition as a classic graph cut problem. It implements the graph cut with an efficient Expectation-Maximization (EM) algorithm [1, 6, 30]. The generated masks embed extra contextual cues among image patches to supervise the training of vision model. The updated model in-turn boosts the accuracy of part partition. Iteratively conducting mask generation and model training results in a loop that trains vision models on the unlabeled dataset. The mask patterns thus could evolve to present different visual cues learning tasks.

We test the effectiveness of the proposed method on three popular MIM architectures, *i.e.,* MAE [18], BEiT [2] and SimMIM [38]. Our method brings significant performance enhances for those three architectures, especially on the semantic segmentation task, *e.g.*, boosts the mIoU by 2%. When compared with recent self-supervised learning methods, our method achieves comparable performance with fewer pre-training epochs, and superior performance with similar training epochs. To the best of our knowledge, this is an original effort on evolved part masking for self-supervised learning. Our method does not require extra pre-trained models or annotations. It effectively ensures the training efficiency, and enhances the generalization ability of trained model by evolving the mask patterns, thus shows potentials to boost the performance of pre-trained vision models.

## 2. Related work

This section briefly reviews recent works on self-supervised learning and masked image modeling, which are closely related to this work.

**Self-supervised Learning.** The past few years have witnessed the boom of Self-Supervised Learning (SSL) in visual representation learning. Generally, the SSL works design an annotation-free pretext task to learn representations. Contrastive learning has dominated the learning algorithms in those works for the past few years. It works by pulling positive samples together and pushing negative samples apart. Related works includes SimCLR [8], MoCo [19], Swav [4], and BYOL [17], *etc.*.

Another category of SSL research predicts the original image based on the partially observed data. For example, RotNet [16] predicts the 2D rotation applied to the input image. The CFN [28] randomly shuffles the image patches and takes Jigsaw puzzles as the pretext task. Autoencoder is a commonly used generative SSL models, which is trained by minimizing the reconstruction error. It has an encoder that maps the input data to a latent space and a decoder to reconstruct the image from the latent representation. Denoising autoencoders (DAE) [34] corrupts an input signal and learns to reconstruct the original signal. A series of methods can be viewed as generalized DAE with different ways of generating corrupted images, including degrading the resolution [7], masking regions [29], or removing certain color channels [39], *etc.*. MIM also can be regarded as one of DAE variants.

**Masked Image Modeling (MIM).** Inspired by the success of MLM [3, 13] in NLP, MIM has been proposed to tackle the data-hungry issue of vision transformers [14, 33]. Generally, MIM methods make use of a vision transformer as the backbone and learn representations from images corrupted by masking. As one of the core designs of MIM, the mask methods largely determine the knowledge that the network could learn in the pre-training phase.

According to the masking criteria, existing methods can be divided into three categories: (a) Random masking is the most common and straightforward method. MAE [18] is one of the representative works that utilise an asymmetric autoencoder to recover a randomly masked input. SimMIM [38] randomly masks larger square patches and minimizes the $\ell_1$ loss between raw pixel values and predicted results. (b) The second category reserves crucial cues for MIM. For example, MST [24] masks only nonessential patches and preserve key patterns in images. MFM [36] uses low-pass/high-pass filters to perform masking, and most object regions with clear semantics are preserved. (c) The third category proposes to mask clues like object regions completely. BEiT [2] employs a block-wise masking method to mask some image objects as a whole. AttnMask [22] proposes to mask patches belonging the most attended objects. SemMAE [23] leverages the iBOT [41] for semantic segmentation and produces the mask according to the segmentation result. Besides, ADIOS [32] utilizes a learned adversarial masking subnet to pose a more challenging MIM task.

**Difference with previous works.** Instead of following fixed criteria to generate mask patterns, we generate evolved masks for different training stages. Compared to SemMAE [23] and ADIOS [32], our method does not introduce extra networks or training cost. It leverages the model being trained to determine the cues that should be masked. Evolved masks make training difficulty on par with the capability of network being trained, hence ensures a more effective and fast self-supervised learning.

# 3. Method

## 3.1. Overview

Our goal is to train a vision transformer on an unlabeled dataset $\mathcal{D}$. For an input image $x \in \mathbb{R}^{HW \times 3}$, where $H, W$ are the spatial size. We generate a binary mask $M \in \{0,1\}^{HW}$ on $x$, and apply $M$ to self-supervised learning. Specifically, we adopt an encoder-decoder structure to recover $x$ from a masked input. For the $t$-th training epoch, the training objective can be denoted as,

$$\arg\min_{\theta,\theta'} \mathop{\mathbb{E}}_{x\sim\mathcal{D}} \mathcal{H}(G^{(t)}_{\theta'}(F^{(t)}_{\theta}(x \odot M)), x \odot (1-M)), \quad (1)$$

where $\odot$ is the element-wise product, $x \odot M$ denotes the masked input. $F^{(t)}_{\theta}$ and $G^{(t)}_{\theta'}$ are encoder and decoder in $t$-th training epoch, respectively. $\mathcal{H}(\cdot,\cdot)$ is the similarity measurement, *e.g.,* $l2$-distance [18] or cross-entropy [2].

In Eq (1), the mask $M$ largely determines encoded cues in the optimized parameters $\theta$ and $\theta'$. Sticking to a fixed mask pattern optimizes the model towards specific tasks. To enhance the generalization capability to different tasks, we aim to evolve the mask patterns at different training stages, *e.g.*, simple masks to learn low-level visual cues at initial training stage, and more complicated masks to learn object semantics at later training stage.

Since the masks are binary values, making them gradually evolve is not trivial. We introduce masking probability values $P^{(t)} \in \mathbb{R}^{HW}$ for each patch to determine $M^{(t)}$, and the ones with high probability values will be masked out. Let $N = H \times W$ denote number of patches. Given a mask ratio $r$ in $t$-th pre-training epoch, the $m = \lfloor N \times r \rfloor$ patches with high probability values $P^{(t)}_i$ will be masked out. Formally, let

$$\mathcal{I}^{(t)} = \text{argsort}(P^{(t)}_i) \quad (2)$$

be the indices sorted by the scores $P^{(t)}_i$, and the patch will be masked as its indices lie in $\mathcal{I}^{(t)}_{1:m}$, which can be formalized as

$$M^{(t)}_i = \begin{cases} 0, \text{if } i \in \mathcal{I}^{(t)}_{1:m} \\ 1, \text{otherwise.} \end{cases} \quad (3)$$

In the early stages of pre-training, as the network acquired less cues modeling capacity, the generated masks consist of more grid-wise patterns that are independent of network capacity. As training progresses, the generated mask gradually evolves to mask several parts inferred by the network being trained to guide it to learn more challenging cues modeling. We use an increasing $\alpha$ to control

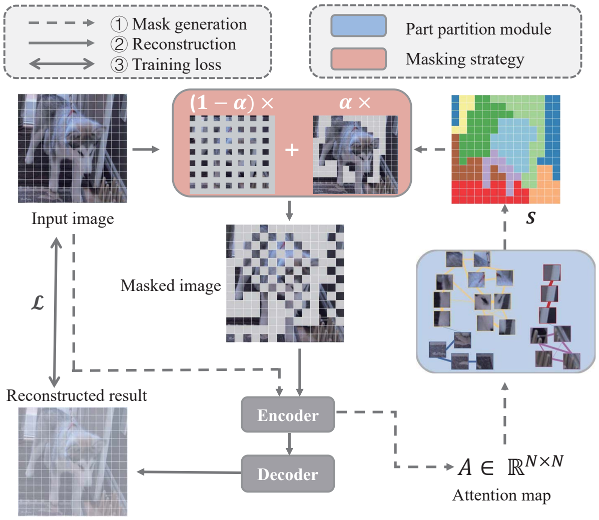

Figure 3. The pipeline of proposed evolved part masking using MAE [18] as an example. Input image is fed into the encoder extracting the attention map $A$ to establish the patches association graph. The part partition module produces the parts annotation $S$ by partitioning the graph. Based on $S$, we generate masks superimposing the effect of grid-wise and part-wise weighted by a dynamic parameter $\alpha$.

the proportion of two mask patterns in the generative masks in $t$-th training epoch, that

$$\alpha^{(t)} = (\frac{t}{total\_epoches})^{\gamma}, \tag{4}$$

where $\gamma$ is a hyperparameter that controls the radian of the $\alpha$ curve following [23]. More discussion about $\gamma$ can be seen in Sec. 4.2. So the evolving process can be achieved by weighted summation of corresponding masking probabilities $P^{grid}$ and $P^{part}$, which can be written as

$$P_i^{(t)} = (1 - \alpha^{(t)}) \times P_i^{grid} + \alpha^{(t)} \times (P_i^{part})^{(t)}. \tag{5}$$

where the generation of $P_i^{grid}$ is static, while the generation of $P_i^{part}$ evolves along the network's modeling capacity. Further, We provide a pseudo-code implementation of the masking strategy in the appendix.

The pipeline of the proposed method can be seen in Fig. 3 and we provide a pseudocode implementation in Appendix.1. Let $\delta \in \mathbb{R}^N$ denote a series of random numbers sampled from a uniform distribution that will be used below to assign values to $P$. We then elaborate on the masking probabilities generation in the following subsections.

### 3.2. Adaptive parts generation

For part-based masking, we produce the $t$-th epoch part annotation $S^{(t)} \in \mathbb{N}^N$ with an adaptive part partition module. Patches belonging to the same part are labelled with the identical natural number in $S^{(t)}$. Then the part-wise masks can be generated by assigning the same masking probability to the patches with the same $S^{(t)}$ value, that

$$(P_i^{part})^{(t)} = \delta_{S_i^{(t)}}. \tag{6}$$

The definition of parts is ambiguous in SSL due to the lack of manually annotated "ground truth" like segmentation tasks. In this work, a semantic part is defined as a group of patches with stronger relationships among its members than between its members and the remainder of the image.

A series of works [2, 5] have demonstrated that transformer attention map can reflect the semantic relationship between tokens and a higher attention value represents a stronger patch relationship. Here we take the attention map from the model being trained for parts partition without introducing additional networks, thus the generated parts are on par with the modeling capacity of the trained model.

Specifically, the attention map produced by the current encoder $F_\theta^{(t)}$ is denoted as $A^{(t)} \in \mathbb{R}^{N \times N}$. Part partition module builds a graph $G = (V, E)$ where nodes $V$ are image patches; $E$ is the edge set; and edge weight $w_{i,j}$ between nodes $i$ and $j$ is the positive attention value among patches, that

$$w_{i,j}^{(t)} = \begin{cases} A_{i,j}^{(t)}, \text{ if } A_{i,j}^{(t)} > 0 \\ 0, \text{ otherwise.} \end{cases} \tag{7}$$

Thus the semantic parts partition is reformulated to a classic graph partition problem. For simplification, we omit the superscript $(t)$ representing the current training epoch below. According to the our definition of part, the objective of graph partition can be written as,

$$\arg\min_S \sum_{i,j} \text{sign}(S_i - S_j) \times w_{i,j}, \tag{8}$$

where $\text{sign}(\cdot)$ is used to distinguish the patches belonging to identical parts or not, that

$$\text{sign}(x) = \begin{cases} 1, \text{if } x \neq 0 \\ -1, \text{if } x = 0. \end{cases} \tag{9}$$

Here we use a simple Expectation-Maximization (EM) algorithm to solve the partition problem following [1, 6, 30]. Since edge weights are calculated pairwise between patches, nodes embedding $\phi_i$ can be taken as $w_i$ to preserve its second-order proximity. Suppose there are $K$ partitions on the graph and the nodes $v_i$ belonging to partition $k$ follows multivariate gaussian distribution

$$v_i \sim \mathcal{N}(\psi_k, \mathcal{C}_k), k \sim \pi_k, \tag{10}$$

where $\psi_k$ is a mean vector; $\mathcal{C}_k$ is a covariance matrix and $\pi_k$ is the probability distribution of partition $k$. The density function for node $v_i$ can be written as

$$p_k(v_i \mid \omega_k) = \frac{1}{\sqrt{(2\pi)^N|\mathcal{C}_k|}} e^{-\frac{1}{2}(\phi_i-\psi)^T\mathcal{C}_k^{-1}(\phi_i-\psi_k)}, \quad (11)$$

where $\omega_k = (\psi_k, \mathcal{C}_k, \pi_k)$ denotes the distribution parameters; $|\mathcal{C}_k|$ is the determinant of $\mathcal{C}_k$. The algorithm solves the problem with iterative E-steps and M-steps. In each E-step, given estimated distribution parameters, we calculate the expectation of the node $v_i$ as

$$E_{i,k} = \frac{\pi_k p_k\left(v_i \mid \omega_k\right)}{\sum_{k=1}^{K} \pi_k p_k\left(v_i \mid \omega_k\right)}. \quad (12)$$

In each M-step, the expectation is used to update the mean vectors and covariance matrix, as

$$\hat{\pi}_k = \frac{\sum_{i=1}^{N} E_{i,k}}{\sum_{j=1}^{K}\sum_{i=1}^{N} E_{i,j}}, \quad (13)$$

$$\hat{\psi}_k = \frac{\sum_{i=1}^{N} E_{i,k}\phi_i}{\sum_{i=1}^{N} E_{i,k}}, \quad (14)$$

$$\hat{\mathcal{C}}_k = \frac{\sum_{i=1}^{N} E_{i,k}\left(\phi_i - \hat{\psi}_k\right)\left(\phi_i - \hat{\psi}_k\right)^T}{\sum_{i=1}^{N} E_{i,k}}. \quad (15)$$

After the iteration, the converged parts partition results are noted as

$$S_i = \arg\max_k E_{i,k}. \quad (16)$$

The generated part annotations are then fed into Eq. (6) to generate a part-wise mask. During the pre-training process, we can adjust the partition number K to control the granularity of the divided parts, and further improve the model's learning of parts associations with different granularities. More discussion can be seen in Sec. 4.2.

### 3.3. Grid generation

For grid-wise masking, we assign the identical score for patches in the same relative location in the mesh, which can be calculated by the patch index as

$$r(i) = (\lfloor\frac{i}{W}\rfloor \bmod 2) \times 2 + (i \bmod 2), \quad (17)$$

where the result $r(i) \in \{0,1,2,3\}$ is the relative location in the mesh. Thus the grid-wise masking generation can be implemented by assign probability that

$$P_i^{grid} = \delta_{r(i)}. \quad (18)$$

### 3.4. Analysis

**Space and Time consumption.** To verify the efficiency of the proposed method, we measure the space and time consumption of the proposed masking strategy during pre-training. The result show that mask generation occupies only $1.7\%$ of the total memory space and $12\%$ of the pre-training time[1]. Compared to the saved training epochs, increased consumption is minor.

**Mask visualization.** In Fig. 4, we visualize the part annotations $S$ and masks $M$ generated at different $\alpha$. It can be seen that the generated $S$ annotates the relative patches with the same labels, and the $S$ produced in different pre-training stages reflects the cues modeling capacity of the model at that time. As the $\alpha$ value increases, the generated mask changes from grid-wise masking to retaining more patches for some parts and less for others, finally removing several parts completely.

## 4. Experiments

### 4.1. Experimental setup

**Datasets.** Self-supervised pre-training is performed on $1.28M$ images from the imageNet-1K [12] training set. Then we do supervised training to evaluate their performance on classification, segmentation and detection tasks. Classification performances are validated on imageNet-1K with end-to-end fine-tuning or linear probing following the common evaluation protocol [18, 23, 38]. For dense prediction tasks, we report the mean intersection-over-Union (mIoU) on ADE20K [40] and bounding box average precision (AP-box) on COCO [26] for semantic segmentation and object detection repectively.

**Implementation details.** Different capacity Vision transformers are utilized as the backbones in our study, *i.e.,* ViT-S and ViT-B [14]. We apply the proposed masking method to three popular MIM models, *i.e.,* MAE [18], BEiT [2] and SimMIM [38] with masking probabilities consistent with those reported in the papers. Models structure and optimization settings follow that in the corresponding works. By default, the mask ratio $r$ is set to $0.75$ following the original model [18], $K$ is linearly reduced from 40 to 10 and $\gamma$ is set to 2, which are ablated in Sec. 4.2. The implementation of block-wise masking follows the masking way in BEiT [2]. For semantic segmentation, we take UperNet [35] as the framework and use the pre-trained encoder parameter to initialize the model backbone. The segmentation models are then fine-tuned on ADE20K for $80K$ with the default setting. For object detection, we adapt the ViT to take the place of the vanilla FPN backbone [25] in Mask R-CNN [20] following [18] and fine-tune the model for 15 epochs.

[1] The time consumption are measured using torch.profiler.

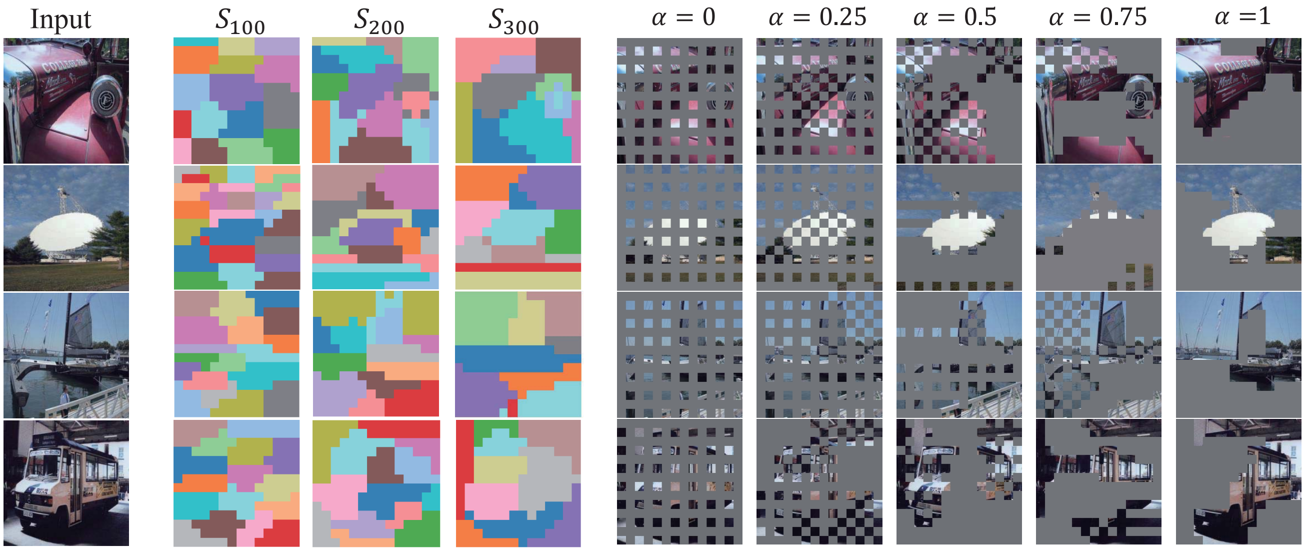

Figure 4. Visualization of the part annotations $S$ and masks $M$ generated in different pre-training phases. The subscript $t$ of $S_t$ denotes current training epoch. The generated mask tends to remove the complete part information as the $\alpha$ increases.

| Epochs | Classification | | | | Segmentation | | | | Detection | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | grid | random | block | ours | grid | random | block | ours | grid | random | block | ours |
| Random ini. | 71.53 | 71.53 | 71.53 | 71.53 | 21.17 | 21.17 | 21.17 | 21.17 | 19.31 | 19.31 | 19.31 | 19.31 |
| 100 | 78.28 | 78.11 | 77.85 | 78.34 | 36.65 | 37.60 | 36.02 | 37.85 | 36.11 | 32.70 | 31.78 | 34.94 |
| 200 | 78.63 | 79.20 | 78.55 | 79.89 | 36.67 | 38.81 | 36.65 | 40.42 | 36.21 | 34.42 | 33.15 | 36.23 |
| 400 | 79.11 | 79.42 | 79.29 | 80.36 | 36.78 | 39.43 | 40.67 | 41.22 | 35.19 | 35.21 | 35.08 | 37.17 |
| 800 | 79.34 | 79.77 | 79.69 | 80.67 | 36.54 | 39.31 | 41.81 | 41.97 | 34.62 | 38.73 | 38.91 | 39.02 |

Table 1. Downstream tasks performance after fine-tuning. Models are pretrained on imageNet-1K [12] with different masking methods. We report imageNet-1K Top-1 accuracy, ADE20K mIoU [40], and COCO AP-box [26] for classification, semantic segmentation and object detection, respectively. The first line is the performance with random initialization in fine-tuning.

| $\alpha$ | $\gamma$ | top-1. acc | mIoU |
|---|---|---|---|
| 0 | - | 78.63 | 36.67 |
| 0.5 | - | 79.02 | 38.13 |
| 1 | - | 75.71 | 35.82 |
| Dynamic | 0.2 | 78.13 | 38.75 |
| | 0.5 | 79.30 | 39.61 |
| | 1 | 79.42 | 39.94 |
| | 2 | 79.89 | 40.42 |
| | 5 | 79.51 | 39.06 |

Table 2. Impact of different $\alpha$ on imageNet-1K classification [12] and ADE20K segmentation [40]. Dynamic $\alpha$ outperforms static one.

## 4.2. Ablation Study

Models in ablation experiments are built upon the ViT-S backbone [14] and asymmetric MAE architecture [18]. More ablation studies refers to Appendix.

**Comparison with static masking.** We first investigate the performance of static masking methods and the proposed evolved method on various downstream tasks in Tab. 1. It can be seen that different static masking methods exhibit distinct advantages. For example, grid-wise masking gives the network better performance with fewer pre-training epochs. And compared with classification tasks, block-wise masking brings more performance improvements on dense prediction tasks under sufficient pre-training. The properties exhibited by random masking lie in between, including strengths and weaknesses of the two. Meanwhile, our method combines the advantages and overcomes the disadvantages by varying masking criteria along the pre-training process, which outperforms these static methods in both performance and efficiency.

**Mask weights.** The weight $\alpha$ controls the proportion of grid-wise and part-wise masks which changes dynamically along the training, and its value is determined by $\gamma$ according to Eq. (4). In Tab. 2, we compare the impact of different $\alpha$ on imageNet-1K classification [12] and ADE20K segmentation [40] pre-trained for 200 epochs. Fixing the $\alpha$ value to 0 or 1 is equivalent to grid-wise or part-wise masking, which brings about poor semantic knowledge learning or slow convergence. A dynamic $\alpha$ solves these problems by combining the advantages of different mask methods. When the $\gamma$ value is set to 2, the network can achieve the

| Fiexd $K$ | top-1. acc | Dynamic $K$ | top-1. acc |
|---|---|---|---|
| 5 | 78.22 | $30 \rightarrow 5$ | 79.43 |
| 10 | 78.94 | $40 \rightarrow 5$ | 79.80 |
| 30 | 79.25 | $30 \rightarrow 10$ | 79.72 |
| 40 | 79.12 | $40 \rightarrow 10$ | 79.89 |

Table 3. Comparison of imageNet-1K [12] classification performance under different $K$. Dynamic K is more robust and perform better.

| Method | Classification | | Segmentation | |
|---|---|---|---|---|
| | top-1. acc | top-5. acc | mIoU | mAcc |
| MAE [18] | 79.20 | 94.61 | 38.81 | 79.61 |
| +ours | 79.89 | 94.78 | 40.42 | 79.95 |
| BEiT [2] | 79.05 | 94.57 | 38.39 | 79.31 |
| +ours | 79.61 | 94.74 | 39.63 | 79.84 |
| SimMIM [38] | 78.51 | 94.18 | 38.13 | 79.15 |
| +ours | 79.08 | 94.55 | 39.57 | 79.80 |

Table 4. ImageNet-1K accuracy and ADE20K performance of three popular MIM models before and after applying the proposed masking with fine-tuning.

best performance on both tasks. This means that properly extending the pre-training epochs of grid-wise masking in the early stage will help learn of parts relationships later.

**Partition number.** During the pre-training process, the partition number $K$ linearly decreases, guiding the network to learn connections between components to between objects. It can be seen from Tab. 3 that a smaller fixed $K$, will harm the model performance, since it is too challenging to make the network predict an entire object in the early training stage. While a gradually decreasing $K$, which brings about a progressive learning process, is robust to the setting of maximum and minimum values.

### 4.3. Benchmark performance

We validate the effect of the proposed method on three popular MIM models, *i.e.,* MAE [18], BEiT [2] and SimMIM [38] and evaluate performance on imageNet-1K classification [12] and ADE20K segmentation [40]. The models are pre-trained using the official code for 200 epochs and fine-tuned on downstream tasks with consistent experimental settings. Results are shown in Tab. 4. Our method brings performance improvement for all three methods, especially on the segmentation task (38.81% *v.s.* 40.42%). For the works originally with random masking, *e.g.*, MAE [18] and SimMIM [38], our method can efficiently boost the performance by explicitly learning better parts relationships. And it can also improve BEiT [2] that originally uses block-wise masks with better training efficiency.

Comparison with recent SSL methods on common imageNet-1K classification setting are shown in Tab. 5. All these methods share the same backbone for fair comparison, *i.e.,* ViT-B [14]. DINO [5], MoCo v3 [11] and AttnMask [22] use an extra momentum encoder as the teacher. MST [24] introduces an MLP head to align the features of the teacher and student. SemMAE [23] uses a pre-trained iBOT to extract token features. Related work ADIOS [32] is not included as it evaluates on other benchmarks. Our method gets comparable performance with fewer pre-training epochs, e.g., 200 *v.s.* 300 in fine-tuning accuracy with state-of-art works. With the same pre-training setting, the proposed method can outperform the SOTA by 0.5%. While our method performs modestly on linear probing, this metrics cannot measure the ability of non-linear representation—which is indeed a strength of deep learning, as domenstrated in [10, 18].

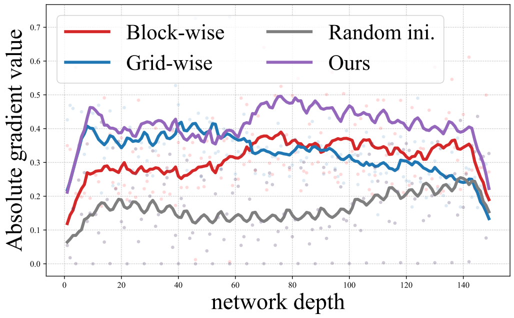

Figure 5. Fine-tuning gradient value against network depth on imageNet-1K classification [12]. The data are fitted using exponential moving averages for better visualisation. The gray line represents the model with random initialization of parameters, without pre-training.

### 4.4. Analysis

This section studies the properties of masking methods and analyses the causes of their performance gap.

The lottery ticket hypothesis [15] demonstrates winning ticket weights tend to change by a larger amount than weights in the rest of the network, which is accompanied by larger gradients. Fig. 5 shows the fine-tuning gradients for different initialized models. It can be seen that grid-wise masks better helps model convergence in shallow layers. And block-wise masking is more helpful for training deep layers, promoting the model learning high-level semantic relationships. The proposed evolved masking method facilitates both deep and shallow network layers.

In Fig. 2 (b), we show the size of the attended area, where these methods exhibit distinct sizes. To further explore the reasons, we visualize the attention maps of some patches in Fig. 6. It can be seen that grid-wise masking makes the pre-trained model focus on adjacent texture-similar features rather than semantically-similar ones. For example, the token on clothes only focuses on the clothes patches and not the person wearing it. Since there are always nearby visi-

| Method | Pre-train data | Date | Extra model | Epochs | Linear probing | Fine-tune |
|---|---|---|---|---|---|---|
| *Supervised traning from scratch* | | | | | | |
| ViT [14] | IN-1K w/ label | ICLR 2021 | - | 300 | - | 77.9 |
| DeiT [33] | IN-1K w/ label | ICML 2021 | - | 300 | - | 81.8 |
| *Contrastive-based SSL Pre-training* | | | | | | |
| DINO [5] | IN-1K | CVPR 2021 | momentum ViT | 1600 | 74.6 | 82.8 |
| MoCo v3 [11] | IN-1K | ICCV 2021 | momentum ViT | 300 | **76.5** | 83.2 |
| MST [24] | IN-1K | NIPS 2021 | MLP Head | 100 | 75.0 | - |
| AttnMask [22] | IN-1K | ECCV 2022 | momentum ViT | 100 | 76.1 | - |
| *MIM SSL Pre-training* | | | | | | |
| BEiT [2] | IN-1K+DALL-E | ICCV 2021 | dVAE [31] | 300 | 56.7 | 82.9 |
| CAE [9] | IN-1K | arxiv 2022 | - | 300 | 64.1 | 83.6 |
| MAE [18] | IN-1K | CVPR 2022 | - | 300 | 64.4 | 83.6 |
| SimMIM [38] | IN-1K | CVPR 2022 | - | 800 | 56.7 | 83.8 |
| MFM [36] | IN-1K | arxiv 2022 | - | 300 | - | 83.1 |
| SemMAE [23] | IN-1K | NIPS 2022 | iBOT [41] | 800 | 68.7 | 83.3 |
| Ours | IN-1K | - | - | 200 | 59.6 | 83.6 |
| Ours | IN-1K | - | - | 300 | 64.7 | **84.1** |

Table 5. Comparison of popular self-supervise learning methods on imageNet-1K [12] using ViT-B [14] as the encoder. Evaluation protocols include top-1 linear probing accuracy and top-1 fine-tuning accuracy. All entries are on an image size of 224 × 224.

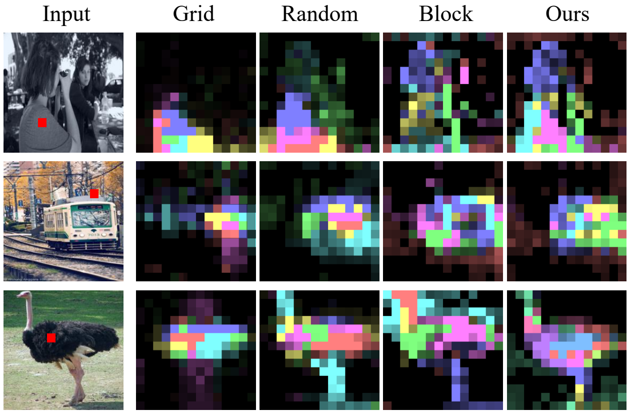

Figure 6. Visualization of attention map for sampled patches under different masking methods. A red dot in the input image annotates the sampled patch. Attention heads are encoded in various colours, and the brightness indicates the attention value.

ble patches to provide cues, the model tends to reconstruct the masked content based on these visible patches, which makes it attends mainly to nearby resemble patches. Yet, the other masking approaches, to varying degrees, allow the network to learn more about high-level relationships. For the masks containing objects with less visible patches, the model must learn to model the relationship between global cues to help predict these objects. Compared to random masks, block-wise ones make the model establish connections between objects on a larger scale. Meanwhile, model with our method also learned the relationship between objects, and the recognition of local features is more accurate.

To summarize, the masks with the visible and the masked patches containing similar content make the pre-trained model learn more low-level texture and better facilitate convergence in shallow layers. Masking the entire parts in the image makes the model pay more attention to global information and learn the connection between objects, which benefits more on the convergence of deep layers and dense prediction tasks. The proposed method enables the mask to evolve with model training, combining the advantages of static methods and overcoming disadvantages.

## 5. Conclusion

This paper investigates the masks on MIM and proposes an evolved part-wise masking method. We find that the choice of masking method directly affects the knowledge learned by the pre-trained model, *e.g.*, grid-wise masking guides the network to learn more local pattern knowledge and removing entire parts pushes it to learn more semantic relationships. The proposed masks combine the advantages of different masking ways by making the masks evolve with the pre-training process and model parameters. We build the patch association graph and reformulate the image partition into a classic graph cut problem, which adaptively groups the patches without introducing additional networks and extra training. With extensive experiments, the model pre-trained under the proposed method demonstrates good versatility and scalability for downstream visual tasks. We hope our study and method will provide timely insights for the superior representation learning ability of SSL.

**Acknowledgement** This work is supported in part by The National Key Research and Development Program of China under Grant No. 2018YFE0118400, in part by Natural Science Foundation of China under Grant No. U20B2052, 61936011.

## References

[1] Arindam Banerjee, Inderjit S Dhillon, Joydeep Ghosh, Suvrit Sra, and Greg Ridgeway. Clustering on the unit hypersphere using von mises-fisher distributions. *Journal of Machine Learning Research*, 6(9), 2005. 2, 4

[2] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021. 1, 2, 3, 4, 5, 7, 8

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 1, 3

[4] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924, 2020. 3

[5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021. 4, 7, 8

[6] Sandro Cavallari, Vincent W Zheng, Hongyun Cai, Kevin Chen-Chuan Chang, and Erik Cambria. Learning community embedding with community detection and node embedding on graphs. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 377–386, 2017. 2, 4

[7] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International conference on machine learning*, pages 1691–1703. PMLR, 2020. 3

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 3

[9] Xiaokang Chen, Mingyu Ding, Xiaodi Wang, Ying Xin, Shentong Mo, Yunhao Wang, Shumin Han, Ping Luo, Gang Zeng, and Jingdong Wang. Context autoencoder for self-supervised representation learning. *arXiv preprint arXiv:2202.03026*, 2022. 8

[10] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021. 7

[11] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9640–9649, 2021. 7, 8

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5, 6, 7, 8

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1, 3

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 3, 5, 6, 7, 8

[15] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018. 7

[16] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018. 3

[17] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020. 3

[18] 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*, pages 16000–16009, 2022. 1, 2, 3, 4, 5, 6, 7, 8

[19] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 3

[20] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 5

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[22] Ioannis Kakogeorgiou, Spyros Gidaris, Bill Psomas, Yannis Avrithis, Andrei Bursuc, Konstantinos Karantzalos, and Nikos Komodakis. What to hide from your students: Attention-guided masked image modeling. *arXiv preprint arXiv:2203.12719*, 2022. 1, 3, 7, 8

[23] Gang Li, Heliang Zheng, Daqing Liu, Chaoyue Wang, Bing Su, and Changwen Zheng. Semmae: Semantic-guided masking for learning masked autoencoders. *arXiv preprint arXiv:2206.10207*, 2022. 1, 3, 4, 5, 7, 8

[24] Zhaowen Li, Zhiyang Chen, Fan Yang, Wei Li, Yousong Zhu, Chaoyang Zhao, Rui Deng, Liwei Wu, Rui Zhao, Ming Tang, et al. Mst: Masked self-supervised transformer for visual representation. *Advances in Neural Information Processing Systems*, 34:13165–13176, 2021. 1, 3, 7, 8

[25] 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. 5

[26] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence

Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 5, 6

[27] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 1

[28] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016. 3

[29] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016. 3

[30] Zachary M Pisano, Joshua S Agterberg, Carey E Priebe, and Daniel Q Naiman. Spectral graph clustering via the expectation-solution algorithm. *Electronic Journal of Statistics*, 16(1):3135–3175, 2022. 2, 4

[31] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021. 8

[32] Yuge Shi, N Siddharth, Philip Torr, and Adam R Kosiorek. Adversarial masking for self-supervised learning. In *International Conference on Machine Learning*, pages 20026–20040. PMLR, 2022. 3, 7

[33] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021. 1, 3, 8

[34] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008. 3

[35] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision (ECCV)*, pages 418–434, 2018. 5

[36] Jiahao Xie, Wei Li, Xiaohang Zhan, Ziwei Liu, Yew Soon Ong, and Chen Change Loy. Masked frequency modeling for self-supervised visual pre-training. *arXiv preprint arXiv:2206.07706*, 2022. 3, 8

[37] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. 1

[38] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9653–9663, 2022. 1, 2, 3, 5, 7, 8

[39] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016. 3

[40] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017. 5, 6, 7

[41] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021. 3, 8