Computational Visual Media
https://doi.org/10.26599/CVM.2025.9450474 Vol. 11, No. 3, June 2025, 483–496

Research Article

FastMAE: Efficient masked autoencoder with offline tokenizer

Meng-Hao Guo[1], Chen Wang[2], Wei Liu[3], and Shi-Min Hu[1] (✉)

Abstract Masked autoencoders (MAEs) have recently achieved great success in computer vision. They can automatically extract representations from unlabeled data and improve the performance of various downstream tasks. However, training an MAE model requires substantial resources, which limits their accessibility to many academic institutions: often laboratories in universities lack the necessary resources. This issue significantly hinders the development of this field. In this paper, we propose FastMAE, an efficient MAE approach. Inspired by the idea of offline tokenizers in natural language processing, FastMAE presents a novel way to build an offline vision tokenizer, which can provide high-level semantics in an efficient way. Benefiting from the offline tokenizer, FastMAE becomes an efficient vision learner. Our experiments demonstrate that FastMAE can achieve 83.6% accuracy with ViT-B in only 18.8 h on 8 NVIDIA Tesla-V100 GPUs, which is 31.3× faster than the original MAE, providing a resource friendly baseline for the computer vision community. Moreover, it also achieves comparable performance to state-of-the-art methods. We hope our research will attract more people to engage in MAE-related research and that we can advance its development together.

Keywords deep learning; computer vision; masked image modeling (MIM)

1 Introduction

The *pretraining followed by fine-tuning* paradigm has been widely adopted in both computer vision and natural language processing (NLP). Self-supervised learning (SSL), aiming to learn a general representation from unlabeled data, is a popular method during the pre-training stage. The crucial step in SSL is to define the pretext task. In recent years, several pretext tasks have emerged, including geometric learning [1], contrastive learning [2–5], generative learning [6, 7]. We focus on one of the generative methods, masked image modeling (MIM).

MIM was inspired by the success of masked language modeling (MLM) in NLP [8], which first randomly masks some words in a sentence and then reconstructs them. BERT [8] pioneers MLM and uses a transformer as the network architecture. BEiT [6], the first work to introduce MIM into computer vision, tries to conduct a visual codebook by using discrete VAE (dVAE) [9] as the reconstruction target to provide semantic guidance. Subsequently, many advanced MIM-based methods have appeared, focusing on various aspects of the problem, including the reconstruction target [10, 11], encoder architecture [12–14], high-dimension input [15], and contrastive learning [16, 17]. In this paper, we concentrate on the reconstruction target.

There are two kinds of reconstruction target: low-level targets and high-level targets. MAE [7] is a representative work considering a low-level target. It changes the reconstruction target from a dVAE to raw pixels, improving performance. Furthermore, MAE does not put masked tokens into the encoder, which significantly reduces the computational overhead and allows models to be easily scaled up to billions of parameters. However, there are obvious gaps between low-level targets and common downstream tasks such as recognition, detection, and segmentation. To address

1 Department of Computer Science, Tsinghua University, Beijing 100084, China. E-mail: M.-H. Guo, gmh20@mails.tsinghua.edu.cn; S.-M. Hu, shimin@tsinghua.edu.cn (✉).
2 University of Pennsylvania, Philadelphia, PA 19104, USA. E-mail: cw.chenwang@outlook.com.
3 Tencent Data Platform, Shenzhen 518057, China. E-mail: wl2223@columbia.edu.
Manuscript received: 2024-01-08; accepted: 2024-12-18

this, researchers have tried to introduce high-level features as reconstruction targets by building discrete codebooks [6, 18], adopting exponential moving average (EMA) teachers [16, 19, 20], and using frozen online teachers [10, 21–24]. However, obtaining high-level reconstruction targets requires additional computation. As can be seen, reconstructing low-level and high-level targets each has its own advantages and disadvantages. The former is efficient and less costly, but neglects critical high-level semantic information for downstream tasks. Recovering high-level targets can provide beneficial semantic information, but requires an online teacher model to build high-level guidance, which brings extra computation and runtime memory overhead, which conflicts with our goal here to reduce computation.

In this paper, we present a novel MIM reconstruction target that can not only capture high-level semantics, but also provide superior efficiency. Our idea is inspired by the success of using an offline codebook in masked language modeling. MLM defines a semantic codebook before pretraining, which allows it to obtain semantics quickly. Motivated by this strategy, we strive to develop an offline visual tokenizer, which can provide high-level semantics in an efficient way. Using this offline visual tokenizer, we can build an efficient and effective MIM model which we call *FastMAE*.

The contributions of this paper are in summary:

- The concept of an offline visual tokenizer, which can efficiently provide a high-level semantic reconstruction target for an MIM method. Benefiting from it, we present an efficient MIM method: FastMAE.
- In order to build the offline tokenizer, we propose a strategy for image discretization, enabling efficient querying of image features without compromising the performance of the MIM method.
- Extensive experiments which demonstrate that our method significantly speeds up the pre-training of MAE (by 31.3×) and achieves comparable results with state-of-the-art (SOTA) methods. More importantly, we provide a lightweight baseline, which requires only 18.8 h on 8 NVIDIA Tesla-V100 GPUs for the pre-training stage. It enables researchers with limited computing resources to participate in and advance this field.

2 Related work

2.1 Masked language modeling

Masked language modeling (MLM) is a popular pre-training approach in NLP. It first randomly masks a few tokens and then recovers their positions in a pre-defined dictionary. BERT [8] was the first work to present masked language modeling. It further proposed next sentence prediction as another pretext task. With the help of these two pretext tasks, BERT successfully learns general representations of words and sentences in context from many unlabeled sentences. It opened the era of large-scale pre-training in NLP. Many NLP MLM-based models followed, such as ALBERT [25] and ELECTRA [26]. The definition of proxy tasks is crucial for the success of MLM. Recovering masked words intuitively brings two clear bonuses: high-level semantics and efficiency.

2.2 Masked image modeling

Inspired by the success of MLM, MIM is a generative self-supervised learning technique designed to apply the achievements of MLM to computer vision. BEiT [6] introduced MIM [27–29] to computer vision. It randomly masks about 40% of the tokens and reconstructs their discrete encodings provided by dVAE. BEiT has provided significant advances in computer vision, particularly in scaling up the model size. Given the outstanding performance of BEiT, numerous works have followed-up, resulting in improvements to MIM in several directions. ConvMAE [12], HiViT [13], and GreenMIM [30] aim to enhance the MIM encoder by introducing hierarchical architectures or convolution operations. Long-sequence MAE [15] explores the input influence during the pre-training stage. CMAE [16] introduces contrastive learning into MIM and enhances its semantic comprehension ability. EVA [31] explores the scaling up capability of MIM. VideoMAE [32] extends MIM to video processing.

This paper focuses on reconstruction targets of MIM models. Many previous works, including MAE [7], PeCo [33], MVP [10], etc., have studied different MIM reconstruction targets. They focus on two main directions. On one hand, existing methods reconstruct low-level targets such as raw pixels and image gradients. However, a significant semantic gap exists between these targets and downstream tasks, such as image classification,

making them less effective. On the other hand, researchers have explored use of high-level semantic features as reconstruction guidance, e.g., in DINO teacher [11], CLIP teacher [10], and EMA teacher [19]. Nevertheless, using an online teacher brings extra computational overhead, which conflicts with our goal of efficient training. In this paper, we propose a new reconstruction target using an offline vision tokenizer, which can combine the efficiency of low-level targets with the semantics of high-level targets.

2.3 Contrastive learning

Aiming to discriminate between different instances, contrastive learning [3, 4, 34–36] is a self-supervised learning method. It was introduced into vision by InstDisc [2], which presents an instance-level non-parametric classifier to learn image representations. The key to the success of contrastive learning lies in the selection of positive and negative samples. Typically, positive samples can be obtained by two different augmentations of the same image, while the use of negative samples has been explored by researchers in different ways. SimCLR [4] increases the number of negative samples by adopting a large batch size, while BYOL [5] argues that negative samples are not necessary for contrastive learning. In addition to exploring positive and negative samples, some other methods consider multi-view contrastive learning [37], cluster-based approaches [38], dense prediction tasks [39], etc.

2.4 Vision transformers

Inspired by the success of transformers in NLP, vision transformers (ViTs) [40–54] were introduced into computer vision as a basic backbone network. Recently, ViTs have quickly conquered various leaderboards and challenged the dominant role of convolutional neural networks (CNNs) in computer vision. ViTs were brought into computer vision by ViT [55], which divides an image into 16 × 16 words and treats images as 1D sequences. This allows ViT to process images with natural language methods. Several attempts have been made to improve ViTs. Swin transformer [40] makes transformers more suitable for vision tasks by computing self-attention in local windows. PVT [42] and PVTv2 [41] enhance vision transformers by introducing an hierarchical architecture. External attention [44] explores a new attention model with linear complexity, reducing the computational overhead of transformers. SwinV2 [56] explores the scaling-up capability of vision transformers. DeiT [57] improves the training recipe for vision transformers. SegFormer [58], PCT [43], UniFormer [59], VideoSwin [60], TransGAN [61], etc., broaden the application field of vision transformers. In this paper, following MAE [7], we mainly focus on how to train a better basic ViT [55].

3 Method

3.1 General description

As Fig. 1 shows, the general formulation of masked image modeling can be summarized as

$$\text{MIM} = \text{Loss}((g(I_{\text{full}}), f(I_{\text{masked}})) \tag{1}$$

where g and f denote different functions, which can be parameterized as neural networks. g is called the *teacher* model, which can be a frozen model, a HoG function, an identity function, etc. f is an autoencoder (AE), which contains both an encoder and decoder. I_{full} denotes an original input image and I_{masked} represents an image degraded by masking some regions. The whole MIM process can be regarded as a representation learning process by reconstructing image features $g(I_{\text{full}})$ from masked image I_{masked}.

In this paper, we wish to develop an efficient MIM method for researchers with limited resources. As explained in the introduction, we observe that reconstruction targets limit the efficiency of current MIM methods, as shown in Table 1. Thus, we aim to address this problem by providing better guidance (through function g). We hope the proposed reconstruction target can efficiently provide high-level semantics for the masked image modeling method.

Specifically, we intend to change function g from an inference process to a query (or index lookup) process with $O(1)$ computational complexity, which can be written as

$$g = \text{inference}(x) \rightarrow g = \text{query}(x) \tag{2}$$

Table 1 Attributes of different reconstruction targets. We desire semantic reconstruction targets that can be quickly obtained

Target	Rapid access	Semantics
Image target	✓	×
Online target	×	✓
Offline target	✓	✓

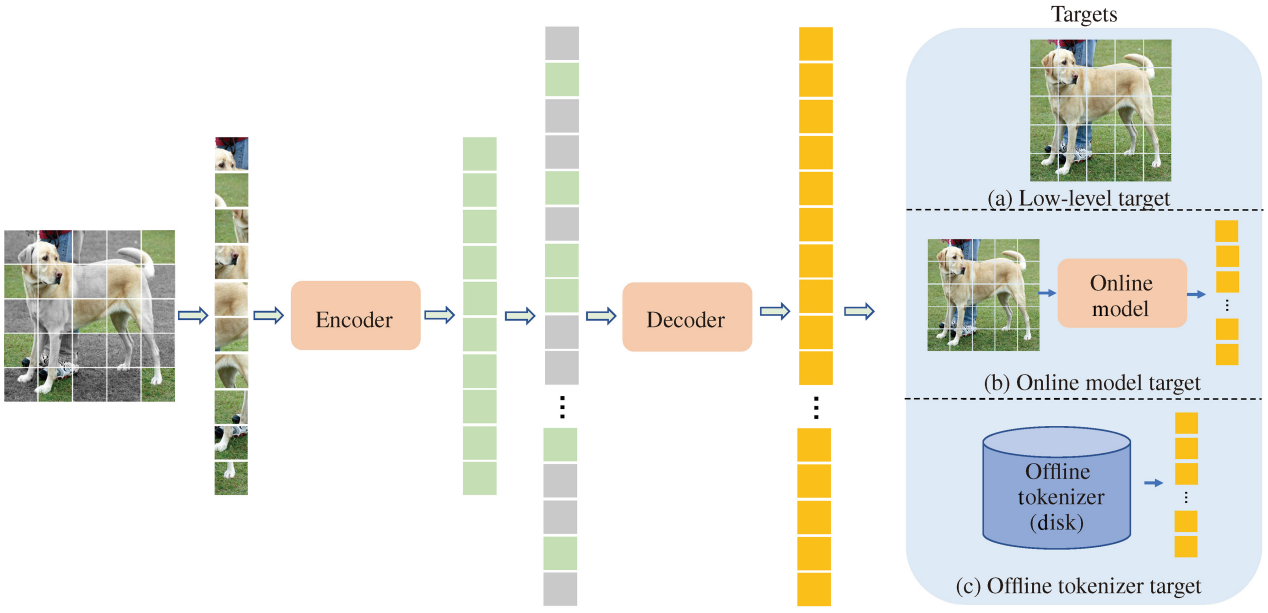

Fig. 1 Pipeline of masked image modeling. The difference between our method and others LIES in the reconstruction target: existing methods usually recover original images or an online teacher target. Our FastMAE reconstructs an offline target.

Here, the right arrow $\rightarrow$ denotes "can be transformed into". To achieve this, we present an offline tokenizer. As Fig. 3 shows, it first preprocesses required tokens and saves them on disk in the offline stage. Then, in the pre-training stage, we can query the related tokens from the disk. In this way, we can reduce computational overhead and run-time memory at the cost of disk storage. The advantages of our offline target are clearly shown in Table 1 and Fig. 2. Our core idea is to exchange expensive computation and run-time memory with cheap disk storage. It is common in computer science, to trade space for time in this way: for instance, bucket sort [62] also adopts a similar idea. In the following, we introduce how we construct an offline tokenizer and pre-train FastMAE.

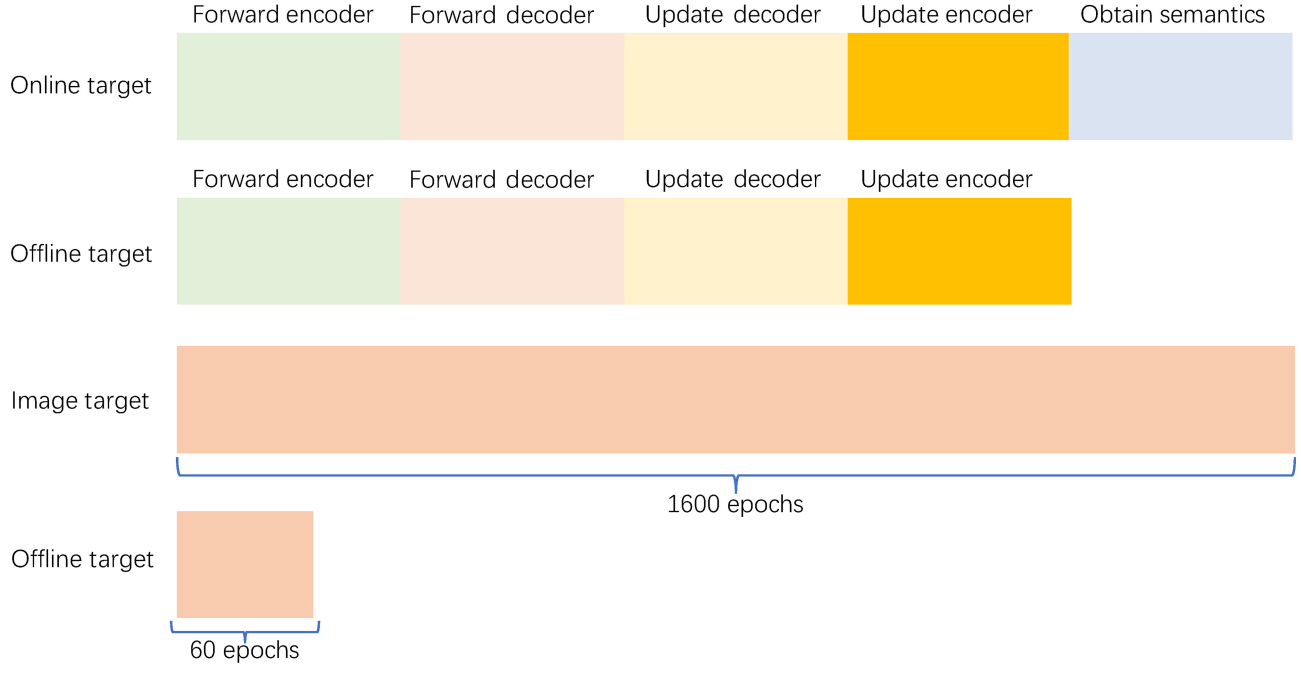

Fig. 2 Time consumed for different reconstruction targets.

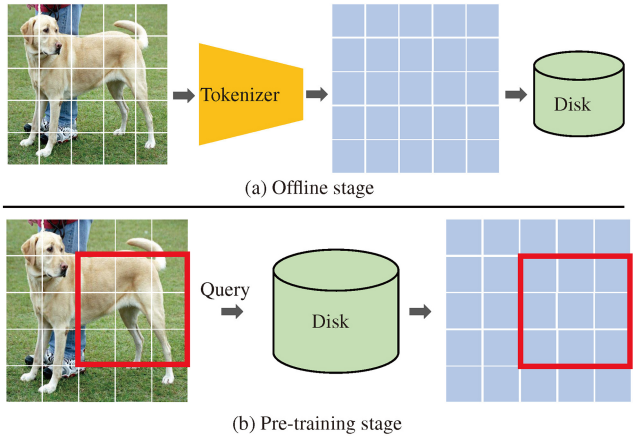

Fig. 3 Offline process and pre-training process. Each white box is a whole and the starting point of each query is at the intersection of the white lines, such as the red box.

3.2 Offline tokenizer

3.2.1 Goals

Before introducing the visual offline tokenizer, let us explore the reasons why the reconstruction targets in MLM can efficiently provide high-level information. We believe the following two points are crucial. Firstly, the tokens in natural language processing are discrete. The discrete property of natural language words implies that tokens are finite and can be encoded in a pre-defined dictionary, e.g., encoding *cat* as 1, *dog* as 2, *human* as 3, etc. After encoding, words can be quickly indexed with $O(1)$ complexity. The second point is that language is an abstract expression of human consciousness; it has been pre-processed by the human brain. Its semantic level is higher than an image. Due to these two characteristics of natural language, MLM can achieve offline encoding and provide high-level information efficiently.

Compared to natural language, images are continuous and raw representations lack human abstraction, so do not enjoy the above two properties. These obstacles prevent MIM from building an offline tokenizer and from acquiring high-level guidance efficiently. Thus, in this paper, our goal is to give images the above two properties, to achieve an efficient masked image modeling method.

3.2.2 High-level semantics

In visual self-supervised learning, providing high-level supervision usually refers to using an intermediate layer of a neural network as supervision instead of raw image supervision. The idea behind this is to use neural networks to encode images instead of the human brain's encoding of natural language.

In general, there are two ways to obtain high-level semantics. The first one is to adopt an online siamese network and exponential moving average (EMA) updating [3, 16], an approach commonly used in contrastive learning. The second is to use an frozen online pre-trained teacher [63], such as MAE [7], iBOT [20], DINO [64], etc., which provides semantics in the pre-training process, an approach commonly used in masked image modeling.

We apply the second strategy and employ a frozen teacher. Differing from the use of online teachers, we use an offline tokenizer to provide high-level information and substitute the inference process with a query process with $O(1)$ complexity. Compared to online teachers, it replaces the computational cost and run-time memory overhead with cheaper disk storage.

3.2.3 Discretizing images

Differing from natural language, raw images are continuous, which makes them unavailable for offline querying. Dosovitskiy [55] proposed regarding a local region (a 16 × 16 patch) as a word. However, as shown in Fig. 4(a) → 4(c), he augments images by randomly scaling and cropping them, resulting in an countless number of possible results, which also cannot be saved and queried offline. Instead, we adopt a different processing strategy to discretize images, as shown in Fig. 4(a) → 4(b) → 4(d). We treat each 16 × 16 region as an inseparable unit, and the starting coordinates of our cropped images in Fig. 4(b) → 4(d) are always located at coordinates at a multiple of 16 such as (0, 0), (0, 16), (16, 0), etc. In this setting, we can randomly choose a cropped view from the set Fig. 4(d) for each training iteration. Due to

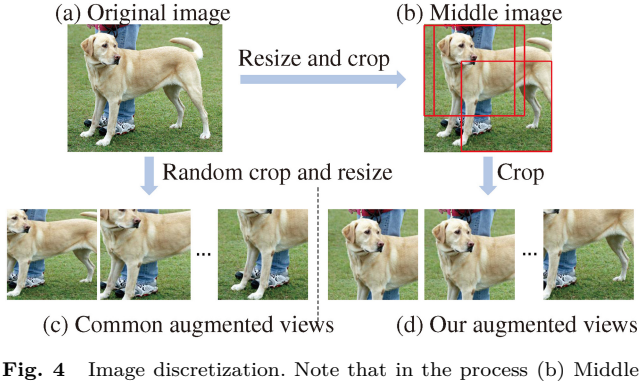

Fig. 4 Image discretization. Note that in the process (b) Middle image → (d) Our augmented view, each starting point coordinate in the red box is a multiple of 16.

the above discretization rule, we can easily index the correspondence between image patches and off-shelf high-level semantic features, as introduced in the next section.

3.2.4 Pre-training

As Fig. 3 shows, the whole process can be divided into two parts, an offline stage and a pre-training stage. In the offline stage (see Fig. 3(a)), we take the resized image (as shown in Fig. 4(b)) as input, and process it by using a tokenizer (the offline teacher). Then, we save the inference results on disk and wait for usage in the pre-training stage. In the pre-training stage, (see Fig. 3(b)), we randomly select a cropped sample from the set in Fig. 4(d) each time as the input view. Then, we can efficiently query the off-shelf high-level semantic features obtained in the offline stage to get high-level guidance quickly, which significantly speeds up the pre-training process. We now given an example using a 224×224 input image. The first cropped sample in Fig. 4(d) corresponds to the region from (0, 0) to (14, 14) in off-the-shelf features. In the same way, the second cropped sample corresponds to the region from (1, 0) to (15, 14) in the off-the-shelf features.

4 Experiments

To demonstrate the effectiveness and efficiency of our proposed FastMAE, we have conducted extensive quantitative and qualitative experiments on various benchmarks, including ImageNet [65], COCO [66], and ADE20K [67]. Experiments were conducted using PyTorch [68] and Jittor [69]. Our main results is that our 60 epochs baseline achieves comparable results to MAE [7] with 1600 training epochs. For 300 epochs training, we show that obtains comparable performance to SOTA models.

4.1 Experimental setting

4.1.1 Pre-training

Following MAE [7], we pretrained our FastMAE on the ImageNet [65] dataset. The input size and patch size were set as 224×224 and 16, respectively, resulting in a sequence of length 196. Then, we randomly masked 75% to obtain a sequence of length 49 as our input. For training, we adopted the AdamW [70] optimizer with an initial learning rate of 1.5×10^{-4}, a batch size 4096, and weight decay 0.05. We applied a cosine annealing schedule and warm-up strategies to adjust the learning rate to improve the training process. We applied various offline tokenizers to provide high-level information such as MAE [7] and DINO [64]. ViT-S/B was selected as the pre-trained encoder. Results after 60 and 300 epochs are reported. The 60 epoch version is intended as a lightweight baseline, which only needs to be trained on 8 NVIDIA Tesla-V100 GPUs for 18.8 h and speeds up MAE by 31.3 times. It serves as a *computing friendly baseline* for researchers with limited computational resources. We use the 300 epoch version for fair comparisons to existing SOTA methods.

4.1.2 Fine-tuning

We kept the setting the same as for MAE [7]. We applied the AdamW [70] optimizer with a weight decay of 0.05, and an initial learning rate of 5×10^{-4}. Drop path rate and layer decay were also applied, and set to 0.1 and 0.65, to avoid overfitting. We fine-tuned our models for 100 epochs with a batch size of 1024. During fine-tuning, we adopted strong data augmentation like MAE, including random clipping, random horizontal flipping, mixup, cutmix, etc. We used the same configuration for fine-tuning ViT-B and ViT-S.

4.1.3 Linear probing

Linear probing means adding a linear classifier on the top of the encoder and only training the added classifier. For a fair comparison, our configuration follows MAE [7]. We trained the linear classifier for 90 epochs using the LARS optimizer with a learning rate of 0.1, and a batch size 16,384.

4.1.4 Segmentation settings

Following the original MAE [7], we conducted semantic segmentation experiments based on UperNet [71]. We adopted the AdamW [70] optimizer with an initial learning rate of 10^{-4}, weight decay 0.05. batch size 16, drop path rate 0.1 and input size 512×512. Poly policy was used to adjust the learning rate and a warm-up strategy was also used. 160,000 training iterations were used for a fair comparison. All segmentation experiments were conducted on the ADE20K [67] dataset.

4.1.5 Detection settings

We used Mask R-CNN [72] as our detection head. With a $1\times$ (12 epochs) training schedule, we applied a AdamW optimizer with an initial learning rate of 0.0003, weight decay 0.05, drop path rate 0.2, and batch size 16. Linear learning rate decay was used to

adjust the learning rate. All models were based on mmdetection [73] and COCO [70] datasets.

4.2 Bag of tricks

4.2.1 All layers matter

Most methods only supervise the output of the last decoder layer. In our experiments, we found that only supervising the last decoder layer does not provide sufficient guidance. We claim in addition to supervising the last decoder layer, all encoder layers need to be supervised. A similar idea was proposed in ConvMAE [12], naming it a *multi-scale decoder*, which supervises the different hierarchical outputs. Instead, we supervise all outputs of encoder layers and provide a stronger signal.

4.2.2 Guidance normalization

Some existing methods do not reconstruct the original target directly, but reconstruct a normalized target. For an $\mathrm{F} \in \mathbb{R}^{N\times C}$ target, MAE [7] normalizes on channel dimension and TEC [21] normalizes on patch dimension. In our experiments, we also found that normalizing guidance can significantly improve the fine-tuned accuracy. We compare different normalizations in Table 4.

4.3 Ablation study

4.3.1 Offline teacher vs. online teacher

We compare offline teachers and online teachers in Table 3 and Table 2. Table 3 shows that there is a little difference (less than 0.1%) between online and offline versions for different teacher models, including DINO [64], MAE [7], iBOT [20], and CLIP [80]. As for other influences, we provide a comprehensive comparison of online and offline teachers in Table 2, which demonstrates that offline teachers have a clear advantage in terms of speed and run-time memory. For example, an offline teacher achieves 2.7 times speedup when using ViT-Large as the online teacher and 1.8 times speedup when using ViT-Base as the online teacher. Furthermore, the offline teacher mode also saves 15%–45% of run-time memory. Of course, we also note that the offline teacher consumes about 1 TB of disk (which only costs about $70 for an SSD, much cheaper than the saved computational resources and run-time memory). We note that the offline teacher does not bring additional overhead as the teacher model becomes larger, which is beneficial when using larger models as a guide such as EVA [31].

Table 3 Ablation study on teacher models. DINO-B means the DINO method with ViT-B backbone. -L means ViT-L model. All encoder models are ViT-B

Teacher	Offline	Online	Accuracy (%)
MAE-B	×	✓	84.4
MAE-B	✓	×	84.3
MAE-L	×	✓	84.6
DINO-B	×	✓	84.5
DINO-B	✓	×	84.5
CLIP-B	×	✓	84.4
CLIP-B	✓	×	84.4
iBOT-B	×	✓	84.7
iBOT-B	✓	×	84.6

Table 4 Ablation study on bag of tricks, using ViT-B as our encoder with 300 training epochs and offline DINO-B as teacher model. ALM = all layers matter

Patch Norm	Channel Norm	ALM	Accuracy (%)
×	×	×	83.9
✓	×	×	84.2
×	✓	×	84.1
×	×	✓	84.1
×	✓	✓	84.4
✓	×	✓	84.5

4.3.2 Choice of teacher

Table 3 shows results of experiments to compare different teachers, including different models and different sizes. We can see that teachers based on contrastive learning (DINO) have slightly better results than teachers based on MIM (MAE). This indicates that the features learned by contrastive learning have higher semantics. Furthermore, a larger

Table 2 Ablation study on online teacher and offline teacher. Our model is faster than MAE because FastMAE only has a one layer decoder. All student backbones were ViT-B

Method	Teacher	Epoch	Online	Offline	Time (s)	Run-time memory (MB)	Disk (TB)	Accuracy (%)
MAE	Image	300	—	—	0.14	7857	0	82.9
FastMAE	MAE-B	300	×	✓	0.09	5371	0.94	84.3
FastMAE	MAE-B	300	✓	×	0.16	6225	0	84.4
FastMAE	MAE-L	300	×	✓	0.09	5372	1.25	84.6
FastMAE	MAE-L	300	✓	×	0.25	7805	0	84.7

teacher can provide more semantic information and achieve better results than the base teacher.

4.3.3 Epoch, decoder depth, architecture

The number of epochs is a common hyperparameter in self-supervised learning; usually, more epochs will lead to better results. The same applies to FastMAE, as shown in Table 5, FastMAE achieves 83.6% accuracy after 60 epochs, and after 300 epochs, it achieves 84.6% accuracy, As for the decoder depth, we performed an ablation study from a 1-layer decoder to a 4-layer decoder and found that decoder depth only slightly influences fine-tuned accuracy. Therefore, in order to save computation and run-time memory, we choose a one layer transformer as our decoder by default. Table 5 also compare different sizes of student architecture (ViT-B vs. ViT-S). It indicates that our method is suitable for different scales of student models and with larger model achieves better performance.

4.3.4 Bag of tricks

The bag of tricks mainly comprises feature normalization and all layers matter. Table 4 considers two normalization methods, channel normalization, and patch normalization. We found that patch normalization gives a slightly better result than channel normalization and patch normalization achieves a 0.3% improvement compared to using the original features. Therefore, we choose patch normalization by default. Furthermore, we find that using all layers matter brings about 0.2% improvement to FastMAE.

4.4 Comparison to existing methods

To demonstrate the effectiveness of our method, we also compared FastMAE to existing MIM methods on common visual tasks, including image recognition, object detection, semantic segmentation, and instance segmentation.

4.4.1 Image classification.

For fairness of comparison, we adopted the same training strategies and settings, including optimizer, batch size, learning rate, etc., as for the representative work MAE [7]. We show the results of our method after 60 epochs and 300 epochs. Table 6 compares existing methods to our FastMAE on the ImageNet validation set. It demonstrates that FastMAE after 300 epochs surpasses other popular methods trained for more epochs, including MAE [7], SimMIM [75], iBOT [20], PeCo [33], ConvMAE [12], etc. in terms of fine-tuned accuracy metrics. Figure 5 clearly shows that FastMAE is 31.3 times faster than MAE [7] while reaching 83.6% top-1 accuracy. Our experiments on 8 NVIDIA Tesla-V100 GPUs show that FastMAE only needs 18.8 h to achieve 83.6% top-1 accuracy, but MAE takes almost 600 h.

Table 5 Ablation study on encoder models and training epochs. All teachers are offline teachers

Encoder	Epochs	Teacher	Accuracy (%)
ViT-B	60	DINO-B	83.5
ViT-B	300	DINO-B	84.3
ViT-B	60	iBOT-B	83.6
ViT-B	300	iBOT-B	84.6
ViT-S	300	iBOT-B	81.8

4.4.2 Semantic segmentation

Aiming to assign each pixel a semantic label, semantic segmentation is a fundamental vision task. Like for image classification, for fairness of comparison, we also used the same training and inference settings used for MAE [7]. For instance, we fine-tuned our method for 100 epochs with batch size 16. Furthermore, we used UperNet [71] as our segmentation head, to keep the same setting as for previous methods. We conducted semantic segmentation experiments on the ADE20K dataset. As Table 7 shows, FastMAE after 60 epochs achieves comparable results to those of popular models including MAE [7]. Our 300-epoch model, achieves 50.7 mIoU and outperforms all other models including iBOT [20] and PeCo [33].

4.4.3 Object detection and instance segmentation

For object detection, we chose Mask R-CNN for

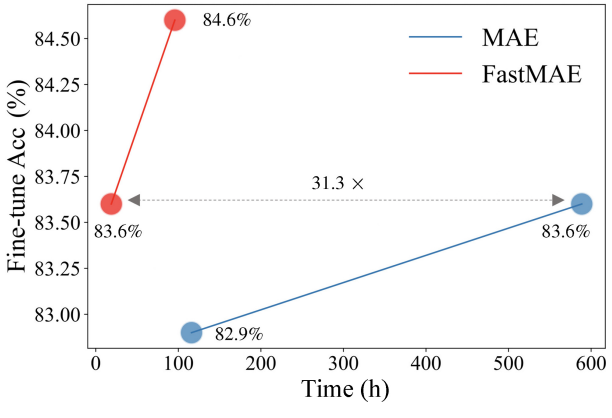

Fig. 5 Comparison of FastMAE to MAE. Horizontal axis: training time. Vertical axis: accuracy after fine-tuning. FastMAE only needs 18.8 h to achieve 83.6% top-1 accuracy while MAE takes almost 600 h. To reach the same accuracy, MAE takes 31.3× longer than FastMAE.

Table 6 Comparison to state-of-the-art methods on the ImageNet validation set. FT accuracy = fine-tuned accuracy. Linear accuracy denotes linear probing accuracy. ⋆ denotes offline teacher

Method	Epochs	Guidance	Pre-training data	Architecture	FT accuracy (%)	Linear accuracy (%)
Supervised	300	Label	IN1K	ViT-B	81.8	—
MoCo v3 [18]	300	EMA Teacher	IN1K	ViT-B	83.0	76.2
DINO [18]	400	EMA Teacher	IN1K	ViT-B	83.3	77.3
BEiT [6]	300	DALLE	IN1K+D250M	ViT-B	83.2	37.6
MAE [7]	300	Image	IN1K	ViT-B	82.9	61.5
MAE [7]	1600	Image	IN1K	ViT-B	83.6	67.8
CIM [74]	300	Image	IN1K	ViT-B	83.3	67.8
MaskFeat [11]	800	HoG	IN1K	ViT-B	84.0	—
CAE [18]	300	Encoder	IN1K	ViT-B	83.6	64.1
iBOT [20]	1600	EMA Teacher	IN1K	ViT-B	84.0	77.1
SimMIM [75]	800	Image	IN1K	ViT-B	83.8	56.7
SIM [76]	1600	EMA Teacher	IN1K	ViT-B	84.1	**78.0**
SdAE [77]	300	EMA Teacher	IN1K	ViT-B	84.1	64.9
MVP [10]	300	CLIP	OpenAI400M	ViT-B	84.4	75.4
ConvMAE [12]	400	Image	IN1K	ConViT-B	84.4	66.9
PeCo [33]	300	Image	IN1K	ViT-B	84.1	—
GreenMIM [30]	800	Image	IN1K	Swin-B	83.8	
Data2Vec [78]	800	EMA Teacher	IN1K	ViT-B	84.2	—
HiViT [13]	800	Image	IN1K	HiViT-B	84.2	—
BoostedMAE [19]	800	EMA Teacher	IN1K	ViT-B	84.2	66.1
FastMIM [79]	400	Image	IN1K	ViT-B	83.6	—
FastMAE	60	iBOT⋆	IN1K	ViT-B	83.6	69.3
FastMAE	300	iBOT⋆	IN1K	ViT-B	**84.6**	72.8

Table 7 Comparison to state-of-the-art methods on the ADE20K validation set. For fairness, we used ViT-B as the backbone for all methods

Method	Epochs	SSL	mIoU
DeiT [57]	300	×	47.0
MoCo V3 [18]	300	✓	47.2
DINO [64]	400	✓	47.2
BEiT [6]	300	✓	44.7
BEiT [6]	800	✓	45.6
MAE [7]	300	✓	45.8
MAE [7]	1600	✓	48.1
CIM [74]	300	✓	43.5
iBOT [20]	1600	✓	50.0
CAE [18]	300	✓	48.3
CAE [18]	1600	✓	50.2
PeCo [33]	800	✓	48.5
BootMAE [19]	800	✓	49.2
FastMAE	60	✓	47.8
FastMAE	300	✓	**50.7**

our detection method, as used in previous methods. All other settings such as batch size, tuning epochs, learning rate, were the same as used by MAE [7] for fairness of comparison. We present object detection and instance segmentation results in Table 8. Under Mask R-CNN 1×, our 60-epoch model achieves better performance than a 300 epoch BEiT [6] model for both object detection and instance segmentation. Our 300-epoch model clearly outperforms the MAE 1600-epoch model for both object detection and instance segmentation.

Table 8 Object detection and instance segmentation on COCO 2017 dataset. Mask R-CNN 1× denotes models are based on Mask R-CNN [72] and we train them for 12 epochs. AP^b and AP^m refer to bounding box AP and mask AP respectively. † represents our implementation. For fair comparison, ViT-B is the backbone for all methods

Method	Mask R-CNN 1×						
	Epochs	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
MoCo v3	300	45.5	67.1	49.4	41.5	65.5	44.4
DINO	400	46.8	68.6	50.9	41.5	65.3	44.5
BEiT	300	39.5	60.6	43.0	35.9	57.7	38.5
BEiT	800	42.1	63.3	46.0	37.8	60.1	40.6
PeCo	800	47.8	—	—	42.6	—	—
MAE†	1600	47.6	68.4	52.1	42.3	65.6	45.7
FastMAE	60	46.7	67.7	50.9	41.5	64.6	44.4
FastMAE	300	**49.0**	**69.9**	**53.7**	**43.3**	**66.8**	**46.7**

5 Conclusions

In this paper, we have presented an efficient masked image modeling method: FastMAE. The core idea of FastMAE is that an offline teacher is a more efficient approach than an online teacher for masked image modeling. Benefiting from FastMAE, we can complete the pre-training phase of masked image modeling in only 18.8 h with 8 NVIDIA Tesla-V100 GPUs, which enables more researchers with limited computing resources to carry out related research, promoting development of this field.

Acknowledgements

This work was supported by the National Science and Technology Major Project (Grant No. 2021ZD0112902), the National Natural Science Foundation of China (Grant Nos. 623B2057 and 62220106003), Tsinghua University Initiative Scientific Research Program, and Tsinghua–Tencent Joint Laboratory for Internet Innovation Technology. The authors sincerely appreciate the dedicated effort and valuable feedback from the anonymous reviewers and editor, which significantly improved the manuscript.

Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article. The author Shi-Min Hu is the Editor-in-Chief of this journal.

References

[1] Gidaris, S.; Singh, P.; Komodakis, N. Unsupervised representation learning by predicting image rotations. *arXiv preprint* arXiv:1803.07728, 2018.

[2] Wu, Z.; Xiong, Y.; Yu, S. X.; Lin, D. Unsupervised feature learning via non-parametric instance discrimination. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3733–3742, 2018.

[3] He, K.; Fan, H.; Wu, Y.; Xie, S.; Girshick, R. Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 9729–9738, 2020.

[4] Chen, Ting, Kornblith, S.; Norouzi, M.; Hinton, G. A simple framework for contrastive learning of visual representations. *arXiv preprint* arXiv:2002.05709, 2020.

[5] 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. Bootstrap your own latent-a new approach to self-supervised learning. *arXiv preprint* arXiv:2006.07733, 2020.

[6] Bao, H.; Dong, L.; Piao, S.; Wei, F. Beit: BERT pre-training of image transformers. *arXiv preprint* arXiv:2106.08254, 2021.

[7] He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; Girshick, R. Masked autoencoders are scalable vision learners. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 16000–16009, 2022.

[8] Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint* arXiv:1810.04805, 2018.

[9] Ramesh, A.; Pavlov, M.; Goh, G.; Gray, S.; Voss, C.; Radford, A.; Chen, M.; Sutskever, I. Zero-shot text-to-image generation. *arXiv preprint* arXiv:2102.12092, 2021.

[10] Wei, L.; Xie, L.; Zhou, W.; Li, H.; Tian, Q. MVP: Multimodality-guided visual pre-training. In: *Computer Vision – ECCV 2022. Lecture Notes in Computer Science, Vol. 13676.* Avidan, S.; Brostow, G.; Cissé, M.; Farinella, G. M.; Hassner, T. Eds. Springer Cham, 337–353, 2022.

[11] Wei, C.; Fan, H.; Xie, S.; Wu, C. Y.; Yuille, A.; Feichtenhofer, C. Masked feature prediction for self-supervised visual pre-training. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 14648–14658, 2022.

[12] Gao, P.; Ma, T.; Li, H.; Dai, J.; Qiao, Y. ConvMAE: Masked convolution meets masked autoencoders. *arXiv preprint* arXiv:2205.03892, 2022.

[13] Zhang, X.; Tian, Y.; Xie, L.; Huang, W.; Dai, Q.; Ye, Q.; Tian, Q. HiViT: A simpler and more efficient design of hierarchical vision transformer. In: Proceedings of the 11th International Conference on Learning Representations, 2023.

[14] Tian, K.; Jiang, Y.; Diao, Q.; Lin, C.; Wang, L.; Yuan, Z. Designing BERT for convolutional networks: Sparse and hierarchical masked modeling. *arXiv preprint* arXiv:2301.03580, 2023.

[15] Hu, R.; Debnath, S.; Xie, S.; Chen, X. Exploring long-sequence masked autoencoders. *arXiv preprint* arXiv:2210.07224, 2022.

[16] Huang, Z.; Jin, X.; Lu, C.; Hou, Q.; Cheng, M. M.; Fu, D.; Shen, X.; Feng, J. Contrastive masked autoencoders are stronger vision learners. *IEEE Transactions on*

Pattern Analysis and Machine Intelligence Vol. 46, No. 4, 2506–2517, 2024.

[17] Lu, C.-Z.; Jin, X.; Huang, Z.; Hou, Q.; Cheng, M.-M.; Feng, J. CMAE-V: Contrastive masked autoencoders for video action recognition. *arXiv preprint* arXiv:2301.06018, 2023.

[18] Chen, X.; Ding, M.; Wang, X.; Xin, Y.; Mo, S.; Wang, Y.; Han, S.; Luo, P.; Zeng, G.; Wang, J. Context autoencoder for self-supervised representation learning. *arXiv preprint* arXiv:2202.03026, 2022.

[19] Dong, X.; Bao, J.; Zhang, T.; Chen, D.; Zhang, W.; Yuan, L.; Chen, D.; Wen, F.; Yu, N. Bootstrapped masked autoencoders forVision BERT pretraining. In: *Computer Vision – ECCV 2022. Lecture Notes in Computer Science, Vol. 13690.* Avidan, S.; Brostow, G.; Cissé, M.; Farinella, G. M.; Hassner, T. Eds. Springer Cham, 247–264, 2022.

[20] Zhou, J.; Wei, C.; Wang, H.; Shen, W.; Xie, C.; Yuille, A.; Kong, T. iBOT: Image bert pre-training with online tokenizer. *arXiv preprint* arXiv:2111.07832, 2021.

[21] Gao, S.; Zhou, P.; Cheng, M.-M.; Yan, S. Towards sustainable self-supervised learning. *arXiv preprint* arXiv:2210.11016, 2022.

[22] Hou, Z.; Sun, F.; Chen, Y.-K.; Xie, Y.; Kung, S.-Y. MILAN: Masked image pretraining on language assisted representation. *arXiv preprint* arXiv:2208.06049, 2022.

[23] Tian, Y.; Xie, L.; Wang, Z.; Wei, L.; Zhang, X.; Jiao, J.; Wang, Y.; Tian, Q.; Ye, Q. Integrally pre-trained transformer pyramid networks. *arXiv preprint* arXiv:2211.12735, 2022.

[24] Zhou, P.; Zhou, Y.; Si, C.; Yu, W.; Ng, T. K.; Yan, S. Mugs: A multi-granular self-supervised learning framework. *arXiv preprint* arXiv:2203.14415, 2022.

[25] Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; Soricut, R. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint* arXiv:1909.11942, 2019.

[26] Clark, K.; Luong, M.-T.; Le, Q. V.; Manning, C. D. ELECTRA: Pre-training text encoders as discriminators rather than generators. *arXiv preprint* arXiv:2003.10555, 2020.

[27] Li, Y.; Fan, H.; Hu, R.; Feichtenhofer, C.; He, K. Scaling language-image pre-training via masking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 23390–23400, 2023.

[28] Ji, G.-P.; Zhuge, M.; Gao, D.; Fan, D.-P.; Sakaridis, C.; Gool, L. V. Masked vision-language transformer in fashion. *arXiv preprint* arXiv:2210.15110, 2022.

[29] Wang, D.; Wang, Q.; Min, W.; Gai, D.; Han, Q.; Li, L.; Geng, Y. SAM-driven MAE pre-training and background-aware meta-learning for unsupervised vehicle re-identification. *Computational Visual Media* Vol. 10, No. 4, 771–789, 2024.

[30] Huang, L.; You, S.; Zheng, M.; Wang, F.; Qian, C.; Yamasaki, T. Green hierarchical vision transformer for masked image modeling. *arXiv preprint* arXiv: 2205.13515, 2022.

[31] Fang, Y.; Wang, W.; Xie, B.; Sun, Q.; Wu, L.; Wang, X.; Huang, T.; Wang, X.; Cao, Y. EVA: Exploring the limits of masked visual representation learning at scale. *arXiv preprint* arXiv:2211.07636, 2022.

[32] Tong, Z.; Song, Y.; Wang, J.; Wang, L. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *arXiv preprint* arXiv:2203.12602, 2022.

[33] Dong, X.; Bao, J.; Zhang, T.; Chen, D.; Zhang, W.; Yuan, L.; Chen, D.; Wen, F.; Yu. N. PeCo: Perceptual codebook for BERT pre-training of vision transformers. *arXiv preprint* arXiv:2111.12710, 2021.

[34] Bardes, A.; Ponce, J.; LeCun, Y. VICRegL: Self-supervised learning of local visual features. *arXiv preprint* arXiv:2210.01571, 2022.

[35] Ma, H.; Li, M.; Yang, J.; Patashnik, O.; Lischinski, D.; Cohen-Or, D.; Huang, H. CLIP-Flow: Decoding images encoded in CLIP space. *Computational Visual Media* Vol. 10, No. 6, 1157–1168, 2024.

[36] Li, J.; Huang, Y.; Wu, M.; Zhang, B.; Ji, X.; Zhang, C. CLIP-SP: Vision-language model with adaptive prompting for scene parsing. *Computational Visual Media* Vol. 10, No. 4, 741–752, 2024.

[37] Tian, Y.; Krishnan, D.; Isola, P. Contrastive multiview coding. In: *Computer Vision – ECCV 2020. Lecture Notes in Computer Science, Vol. 12356.* Vedaldi, A.; Bischof, H.; Brox, T.; Frahm, J.-M. Eds. Springer Cham, 776–794. Springer, 2020.

[38] Caron, M.; Misra, I.; Mairal, J.; Goyal, P.; Bojanowski, P.; Joulin, A. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint* arXiv:2006.09882, 2021.

[39] Xie, Z.; Lin, Y.; Zhang, Z.; Cao, Y.; Lin, S.; Hu, H. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 16684–16693, 2021.

[40] Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 9992–10002, 2021.

[41] Wang, W.; Xie, E.; Li, X.; Fan, D. P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; Shao, L. PVT v2: Improved baselines with pyramid vision transformer. *Computational Visual Media* Vol. 8, No. 3, 415–424, 2022.

[42] Wang, W.; Xie, E.; Li, X.; Fan, D.-P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; Shao, L. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 548–558, 2021.

[43] Guo, M. H.; Cai, J. X.; Liu, Z. N.; Mu, T. J.; Martin, R. R.; Hu, S. M. PCT: Point cloud transformer. *Computational Visual Media* Vol. 7, No. 2, 187–199, 2021.

[44] Guo, M.-H.; Liu, Z. N.; Mu, T. J.; Hu, S. Beyond self-attention: External attention using two linear layers for visual tasks. *arXiv preprint* arXiv:2105.02358, 2007.

[45] Yuan, L.; Chen, Y.; Wang, T.; Yu, W.; Shi, Y.; Jiang, Z.-H.; Tay, F. E.; Feng, J.; Yan, S. Tokens-to-token ViT: Training vision transformers from scratch on ImageNet. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 558–567, 2021.

[46] Guo, M.-H.; Lu, C.-Z.; Liu, Z.-N.; Cheng, M.-M.; Hu, S.-M. Visual attention network. *Computational Visual Media* Vol. 9, No. 4, 733–752, 2023.

[47] Guo, M.-H.; Lu, C.-Z.; Hou, Q.; Liu, Z.; Cheng, M.-M.; Hu, S.-M. SegNeXt: Rethinking convolutional attention design for semantic segmentation. *arXiv preprint* arXiv:2209.08575, 2022.

[48] Dai, Z.; Liu, H.; Le, Q. V.; Tan, M. CoATNet: Marrying convolution and attention for all data sizes. *arXiv preprint* arXiv:2106.04803, 2021.

[49] Geng, Z.; Guo, M.-H.; Chen, H.; Li, X.; Wei, K.; Lin, Z. Is attention better than matrix decomposition? *arXiv preprint* arXiv:2109.04553, 2021.

[50] Han, K.; Xiao, A.; Wu, E.; Guo, J.; Xu, C.; Wang, Y. Transformer in transformer. In: Proceedings of the 35th International Conference on Neural Information Processing Systems, 15908–15919, 2021.

[51] Xu, Y.; Zhang, Q.; Zhang, J.; Tao, D. ViTAE: Vision transformer advanced by exploring intrinsic inductive bias. *arXiv preprint* arXiv:2106.03348, 2021.

[52] Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In: *Computer Vision – ECCV 2020. Lecture Notes in Computer Science, Vol. 12346.* Vedaldi, A.; Bischof, H.; Brox, T.; Frahm, J.-M. Eds. Springer Cham, 213–229, 2020.

[53] Guo, M.-H.; Xu, J.; Zhang, Y.; Song, J.; Peng, H.; Deng, Y.-X.; Dong, X.; Nakayama, K.; Geng, Z.; Wang, C.; et al. R-bench: Graduate-level multi-disciplinary benchmarks for LLM & MLLM complex reasoning evaluation. *arXiv preprint* arXiv:2505.02018, 2025.

[54] Guo, M.-H.; Chu, X.; Yang, Q.; Mo, Z.-H.; Shen, Y.; Li, P.-L.; Lin, X.; Zhang, J.; Chen, X.-S.; Zhang, Y.; et al. RBench-V: A primary assessment for visual reasoning models with multi-modal outputs. *arXiv preprint* arXiv:2505.16770, 2025.

[55] Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint* arXiv:2010.11929, 2020.

[56] Liu, Z.; Hu, H.; Lin, Y.; Yao, Z.; Xie, Z.; Wei, Y.; Ning, J.; Cao, Y.; Zhang, Z.; Dong, L.; et al. Swin transformer v2: Scaling up capacity and resolution. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 12009–12019, 2022.

[57] Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; Jґegou, H. Training data-efficient image transformers & distillation through attention. In: Proceedings of the 38th International Conference on Machine Learning, 10347–10357, 2021.

[58] Xie, E.; Wang, W.; Yu, Z.; Anandkumar, A.; Alvarez, J. M.; Luo, P. SegFormer: Simple and efficient design for semantic segmentation with transformers. *arXiv preprint* arXiv:2105.15203, 2021.

[59] Li, K.; Wang, Y.; Gao, P.; Song, G.; Liu, Y.; Li, H.; Qiao, Y. UniFormer: Unified transformer for efficient spatiotemporal representation learning. *arXiv preprint* arXiv:2201.04676, 2022.

[60] Liu, Z.; Ning, J.; Cao, Y.; Wei, Y.; Zhang, Z.; Lin, S.; Hu, H. Video swin transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 3192–3201, 2022.

[61] Jiang, Y.; Chang, S.; Wang, Z. TransGAN: Two pure transformers can make one strong GAN, and that can scale up. *arXiv preprint* arXiv:2102.07074, 2021.

[62] Leiserson, C. E.; Rivest, R. L.; Cormen, T. H.; Stein, C. *Introduction to Algorithms, Vol. 3*. MIT Press, 1994.

[63] Xue, H.; Gao, P.; Li, H.; Qiao, Y.; Sun, H.; Li, H.; Luo, J. Stare at what you see: Masked image modeling without reconstruction. *arXiv preprint* arXiv:2211.08887, 2022.

[64] Caron, M.; Touvron, H.; Misra, I.; Jégou, H.; Mairal, J.; Bojanowski, P.; Joulin, A. Emerging properties in self-supervised vision transformers. *arXiv preprint* arXiv:2104.14294, 2021.

[65] Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Kai, L.; Li, F.-F.

ImageNet: A large-scale hierarchical image database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 248–255, 2009.

[66] Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C. L. Microsoft COCO: Common objects in context. In: *Computer Vision – ECCV 2014. Lecture Notes in Computer Science, Vol. 8693.* Fleet, D.; Pajdla, T.; Schiele, B.; Tuytelaars, T. Eds. Springer Cham, 740– 755, 2014.

[67] Zhou, B.; Zhao, H.; Puig, X.; Fidler, S.; Barriuso, A.; Torralba, A. Scene parsing through ADE20K dataset. In: Proceedings of the IEEE conference on computer vision and pattern recognition, 633–641, 2017.

[68] Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. Pytorch: An imperative style, high-performance deep learning library. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems, 8026–8037, 2019.

[69] Hu, S. M.; Liang, D.; Yang, G. Y.; Yang, G. W.; Zhou, W. Y. Jittor: A novel deep learning framework with meta-operators and unified graph execution. *Science China Information Sciences* Vol. 63, No. 12, Article No. 222103, 2020.

[70] Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. *arXiv preprint* arXiv:1711.05101, 2017.

[71] Xiao, T.; Liu, Y.; Zhou, B.; Jiang, Y.; Sun, J. Unified perceptual parsing for scene understanding. In: *Computer Vision – ECCV 2018. Lecture Notes in Computer Science, Vol. 11209.* Ferrari, V.; Hebert, M.; Sminchisescu, C.; Weiss, Y. Eds. Springer Cham, 418–434, 2018.

[72] He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, 2961–2969, 2017.

[73] Chen, K.; Wang, J.; Pang, J.; Cao, Y.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Xu, J.; et al. MMDetection: Open MMLab detection toolbox and benchmark. *arXiv preprint* arXiv:1906.07155, 2019.

[74] Fang, Y.; Dong, L.; Bao, H.; Wang, X.; Wei, F. Corrupted image modeling for self-supervised visual pre-training. *arXiv preprint* arXiv:2202.03382, 2022.

[75] Xie, Z.; Zhang, Z.; Cao, Y.; Lin, Y.; Bao, J.; Yao, Z.; Dai, Q.; Hu, H. SimMIM: A simple framework for masked image modeling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 9653–9663, 2022.

[76] Tao, C.; Zhu, X.; Huang, G.; Qiao, Y.; Wang, X.; Dai, J. Siamese image modeling for self-supervised vision representation learning. *arXiv preprint* arXiv: 2206.01204, 2022.

[77] Chen, Y.; Liu, Y.; Jiang, D.; Zhang, X.; Dai, W.; Xiong, H.; Tian, Q. SdAE: Self-distillated masked autoencoder. *arXiv preprint* arXiv:2208.00449, 2022.

[78] Baevski, A.; Hsu, W. N.; Xu, Q.; Babu, A.; Gu, J.; Auli, M. data2vec: A general framework for self-supervised learning in speech, vision and language. In: Proceedings of the 39th International Conference on Machine Learning, 1298–1312, 2022.

[79] Guo, J.; Han, K.; Wu, H.; Tang, Y.; Wang, Y.; Xu, C. FastMIM: Expediting masked image modeling pre-training for vision. *arXiv preprint* arXiv:2212.06593, 2022.

[80] Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning transferable visual models from natural language supervision. In: Proceedings of the 38th International Conference on Machine Learning, 8748–8763, 2021.

Meng-Hao Guo is a Ph.D. candidate under the supervision of Prof. Shi-Min Hu in Tsinghua University. His research interests include computer vision and computer graphics. He has published papers various journals and conferences such as IEEE TPAMI, ACM TOG, NeurIPS, CVPR, and CVMJ.

Chen Wang is a Ph.D. student at the University of Pennsylvania. He received his bachelor and master degrees in computer science from Tsinghua University. His research interests include computer graphics and computer vision.

Wei Liu received his Ph.D. degree in electrical engineering and computer science from Columbia University, in 2012. He is currently a distinguished scientist of Tencent and the director of Ads Multimedia AI with the Tencent Data Platform. He was a research staff member of the IBM T. J. Watson Research Center from 2012 to 2015. He has long been devoted to fundamental research and technological development in core fields of AI, including deep learning, machine learning, reinforcement learning, computer vision, information retrieval, big data, etc. To date, he has more than 280 peer-reviewed technical papers, and more than 30 US patents. He currently serves on the editorial boards of IEEE TPAMI, IEEE TNNLS, and IEEE Intelligent Systems. He is a fellow of the IAPR and IMA, and an elected member of the ISI.

Shi-Min Hu is currently a professor of computer science at Tsinghua University. He received his Ph.D. degree from Zhejiang University in 1996. His research interests include geometry processing, image & video processing, rendering, computer animation, and CAD. He has published more than 100 papers in journals and refereed conferences. He is Editor-in-Chief of *Computational Visual Media*, and on the editorial boards of several journals, including *Computer Aided Design* and *Computer & Graphics*.

To submit a manuscript, please go to `https://jcvm.org`.