Masked Autoencoders Are Scalable Vision Learners

Kaiming He*,† Xinlei Chen* Saining Xie Yanghao Li Piotr Dollár Ross Girshick

*equal technical contribution †project lead

Facebook AI Research (FAIR)

Abstract

This paper shows that masked autoencoders (MAE) are scalable self-supervised learners for computer vision. Our MAE approach is simple: we mask random patches of the input image and reconstruct the missing pixels. It is based on two core designs. First, we develop an asymmetric *encoder-decoder architecture, with an encoder that operates only on the visible subset of patches (without mask tokens), along with a lightweight decoder that reconstructs the original image from the latent representation and mask tokens. Second, we find that masking a high proportion of the input image,* e.g.*, 75%, yields a nontrivial and meaningful self-supervisory task. Coupling these two designs enables us to train large models efficiently and effectively: we accelerate training (by 3× or more) and improve accuracy. Our scalable approach allows for learning high-capacity models that generalize well:* e.g.*, a vanilla ViT-Huge model achieves the best accuracy (87.8%) among methods that use only ImageNet-1K data. Transfer performance in downstream tasks outperforms supervised pre-training and shows promising scaling behavior.*

1. Introduction

Deep learning has witnessed an explosion of architectures of continuously growing capability and capacity [33, 25, 57]. Aided by the rapid gains in hardware, models today can easily overfit one million images [13] and begin to demand hundreds of millions of—often publicly inaccessible—*labeled* images [16].

This appetite for data has been successfully addressed in natural language processing (NLP) by self-supervised pre-training. The solutions, based on autoregressive language modeling in GPT [47, 48, 4] and *masked autoencoding* in BERT [14], are conceptually simple: they remove a portion of the data and learn to predict the removed content. These methods now enable training of generalizable NLP models containing over one hundred billion parameters [4].

The idea of masked autoencoders, a form of more general denoising autoencoders [58], is natural and applicable in computer vision as well. Indeed, closely related research in vision [59, 46] preceded BERT. However, despite significant interest in this idea following the success of BERT, progress of autoencoding methods in vision lags behind NLP. We ask: *what makes masked autoencoding different between vision and language*? We attempt to answer this question from the following perspectives:

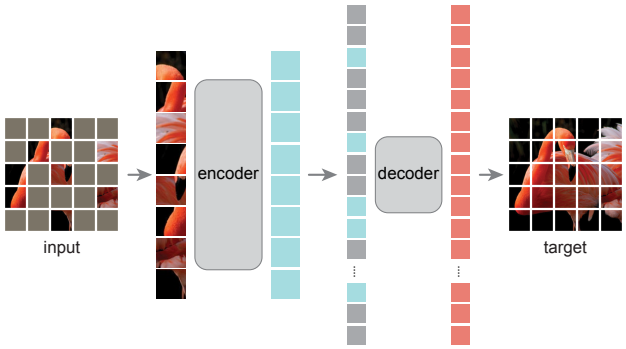

Figure 1. **Our MAE architecture**. During pre-training, a large random subset of image patches (*e.g.*, 75%) is masked out. The encoder is applied to the small subset of *visible patches*. Mask tokens are introduced *after* the encoder, and the full set of encoded patches and mask tokens is processed by a small decoder that reconstructs the original image in pixels. After pre-training, the decoder is discarded and the encoder is applied to uncorrupted images (full sets of patches) for recognition tasks.

(i) Until recently, architectures were different. In vision, convolutional networks [34] were dominant over the last decade [33]. Convolutions typically operate on regular grids and it is not straightforward to integrate 'indicators' such as mask tokens [14] or positional embeddings [57] into convolutional networks. This architectural gap, however, has been addressed with the introduction of Vision Transformers (ViT) [16] and should no longer present an obstacle.

(ii) Information density is different between language and vision. Languages are human-generated signals that are highly semantic and information-dense. When training a model to predict only a few missing words per sentence, this task appears to induce sophisticated language understanding. Images, on the contrary, are natural signals with heavy spatial redundancy—*e.g.*, a missing patch can be recovered from neighboring patches with little high-level un-

掩码自编码器是可扩展的视觉学习器

何恺明*,† 陈鑫磊* 谢赛宁 李扬昊 彼得·多尔·罗斯·吉什克

*同等技术贡献 †项目负责人

Facebook AI Research (FAIR)

摘要

This paper shows that masked autoencoders (MAE) are scalable self-supervised learners for computer vision. Our MAE approach is simple: we mask random patches of the input image and reconstruct the missing pixels. It is based on two core designs. First, we develop an 不对称 *encoder-decoder architecture, with an encoder that operates only on the visible subset of patches (without mask tokens), along with a lightweight decoder that reconstructs the original image from the latent representation and mask tokens. Second, we find that masking a high proportion of the input image,* 例如 *., 75%, yields a nontrivial and meaningful self-supervisory task. Coupling these two designs enables us to train large models efficiently and effectively: we accelerate training (by 3× or more) and improve accuracy. Our scalable approach allows for learning high-capacity models that generalize well:* 例如 *., a vanilla ViT-Huge model achieves the best accuracy (87.8%) among methods that use only ImageNet-1K data. Transfer performance in downstream tasks outperforms supervised pre-training and shows promising scaling behavior.*

1. 引言

深度学习见证了架构的爆炸式增长，其能力和容量不断提升[33, 25, 57]。在硬件快速发展的助力下，如今的模型能够轻松过拟合百万张图像[13]，并开始需求数亿张——通常公开不可获取的——*labeled*图像[16]。

这种对数据的渴求在自然语言处理（NLP）领域已通过自监督预训练得到成功解决。基于GPT中自回归语言建模[47,48,4]和BERT中*masked autoencoding*[14]的解决方案在概念上很简单：它们移除部分数据，并学习预测被移除的内容。这些方法如今能够训练包含超过一千亿参数的可泛化NLP模型[4]。

掩码自编码器的思想，是更通用的去噪自编码器[58]的一种形式，在计算机视觉中也很自然且适用。事实上，密切相关的

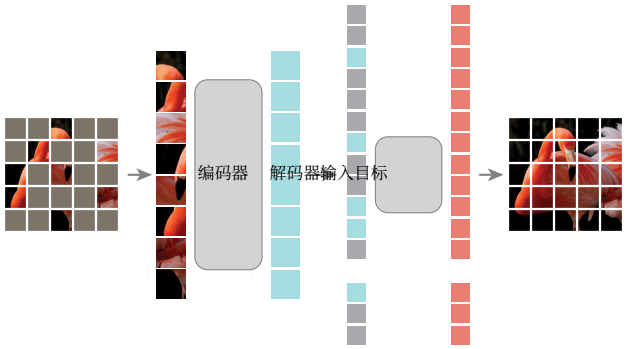

图1. 我们的MAE架构。在预训练期间，图像块的大规模随机子集（*e.g.*，即75%）被遮蔽。编码器仅作用于*visible patches*的小规模子集。掩码标记被引入*after*编码器，完整的编码块集与掩码标记通过小型解码器处理，以像素方式重建原始图像。预训练完成后，解码器被丢弃，编码器被应用于未损坏图像（完整块集）以执行识别任务。

在视觉领域[59,46]的研究早于BERT。然而，尽管在BERT成功后这一理念引起了广泛关注，但视觉中自编码方法的进展仍落后于自然语言处理。我们提出：*what makes masked autoencoding different between vision and language?* 我们尝试从以下角度回答这个问题：

(i) 直到最近，架构还是不同的。在视觉领域，卷积网络[34]在过去十年中占据主导地位[33]。卷积通常操作在规则网格上，将掩码标记[14]或位置嵌入[57]等"指示符"集成到卷积网络中并不直接。然而，随着视觉变换器（ViT）[16]的引入，这种架构差距已得到解决，应不再构成障碍。

（二）信息密度在语言和视觉领域存在差异。语言是人类产生的高度语义化和信息密集的信号。当训练模型仅预测句子中少量缺失词汇时，这项任务似乎能诱导出复杂的语言理解能力。相反地，图像是具有强烈空间冗余的自然信号——{v*}，一个缺失的图像块只需通过相邻区块即可重建，几乎不需要高层次的理解。

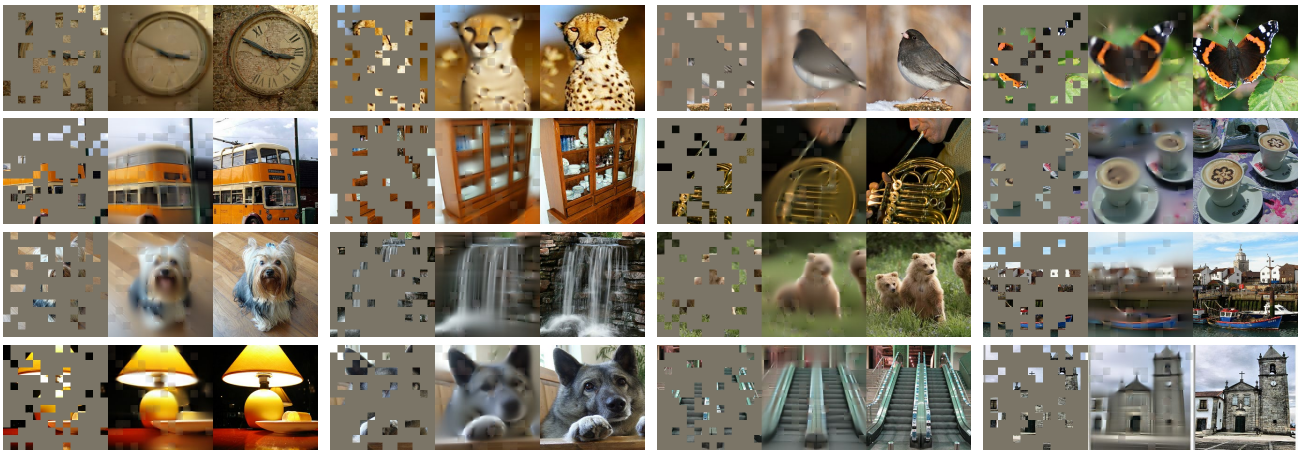

Figure 2. Example results on ImageNet *validation* images. For each triplet, we show the masked image (left), our MAE reconstruction[†] (middle), and the ground-truth (right). The masking ratio is 80%, leaving only 39 out of 196 patches. More examples are in the appendix.
[†]*As no loss is computed on visible patches, the model output on visible patches is qualitatively worse. One can simply overlay the output with the visible patches to improve visual quality. We intentionally opt not to do this, so we can more comprehensively demonstrate the method's behavior.*

Figure 3. Example results on COCO validation images, using an MAE trained on ImageNet (the same model weights as in Figure 2). Observe the reconstructions on the two right-most examples, which, although different from the ground truth, are semantically plausible.

derstanding of parts, objects, and scenes. To overcome this difference and encourage learning useful features, we show that a simple strategy works well in computer vision: masking a *very high* portion of random patches. This strategy largely reduces redundancy and creates a challenging self-supervisory task that requires holistic understanding beyond low-level image statistics. To get a qualitative sense of our reconstruction task, see Figures 2 – 4.

(iii) The autoencoder's *decoder*, which maps the latent representation back to the input, plays a different role between reconstructing text and images. In vision, the decoder reconstructs *pixels*, hence its output is of a lower semantic level than common recognition tasks. This is in contrast to language, where the decoder predicts missing *words* that contain rich semantic information. While in BERT the decoder can be trivial (an MLP) [14], we found that for images, the decoder design plays a key role in determining the semantic level of the learned latent representations.

Driven by this analysis, we present a simple, effective, and scalable form of a masked autoencoder (MAE) for visual representation learning. Our MAE masks random patches from the input image and reconstructs the missing patches in the pixel space. It has an *asymmetric* encoder-decoder design. Our encoder operates only on the visible subset of patches (without mask tokens), and our decoder is lightweight and reconstructs the input from the latent representation along with mask tokens (Figure 1). Shifting the mask tokens to the small decoder in our asymmetric encoder-decoder results in a large reduction in computation. Under this design, a very high masking ratio (*e.g.*, 75%) can achieve a win-win scenario: it optimizes accuracy while allowing the encoder to process only a small portion (*e.g.*, 25%) of patches. This can reduce overall pre-training time by 3× or more and likewise reduce memory consumption, enabling us to easily scale our MAE to large models.

Our MAE learns very high-capacity models that generalize well. With MAE pre-training, we can train data-hungry models like ViT-Large/-Huge [16] on ImageNet-1K with improved generalization performance. With a vanilla ViT-Huge model, we achieve 87.8% accuracy when fine-tuned on ImageNet-1K. This outperforms all previous results that use only ImageNet-1K data. We also evaluate transfer learning on object detection, instance segmentation, and semantic segmentation. In these tasks, our pre-training achieves better results than its supervised pre-training counterparts, and more importantly, we observe significant gains by scaling up models. These observations are aligned with those witnessed in self-supervised pre-training in NLP [14, 47, 48, 4] and we hope that they will enable our field to explore a similar trajectory.

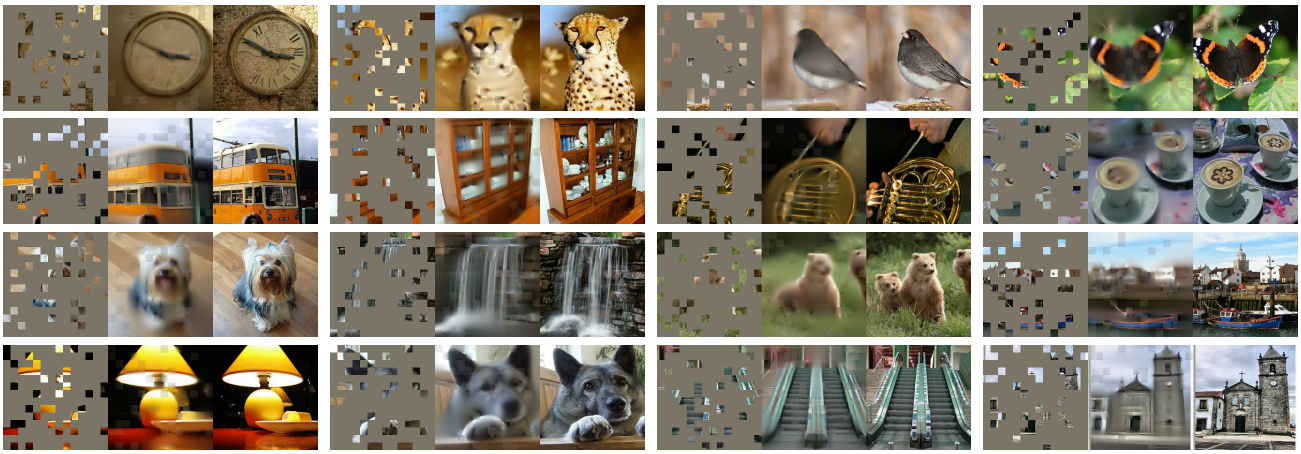

图2. ImageNet *validation* 图像上的示例结果。对于每个三元组，我们展示了掩码图像（左）、我们的MAE重建结果[†](中间)和真实图像（右）。掩码比例为80%，仅保留196个图像块中的39个。更多示例见附录。

[†]*As no loss is computed on visible patches, the model output on visible patches is qualitatively worse. One can simply overlay the output with the visible patches to improve visual quality. We intentionally opt not to do this, so we can more comprehensively demonstrate the method's behavior.*

图3. 在COCO验证图像上的示例结果，使用在ImageNet上训练的MAE（与图2中相同的模型权重）。观察最右侧两个示例的重建结果，尽管与真实值不同，但语义上是合理的。

对部分、物体和场景的整体理解。为克服这一差异并促进学习有用特征，我们证明在计算机视觉中一个简单策略效果显著：对随机图像块进行*very high*比例的掩码。该策略大幅减少了冗余，并创造了一个具有挑战性的自监督任务，需要超越低级图像统计的整体理解。关于重建任务的定性示例，请参见图2至图4。

（iii）自编码器的解码器{v*}，其作用是将潜在表示映射回输入，在重建文本和图像时扮演着不同角色。在视觉领域，解码器重建的是{v*}，因此其输出语义层级低于常见识别任务。这与语言领域形成鲜明对比——解码器预测的是包含丰富语义信息的缺失{v*}。虽然BERT中的解码器可以非常简单（一个多层感知机）[14]，但我们发现对于图像而言，解码器设计对所学潜在表示的语义层级起着决定性作用。

基于这一分析，我们提出了一种简单高效且可扩展的掩码自编码器（MAE）形式，用于视觉表征学习。我们的MAE从输入图像中随机掩码图像块，并在像素空间中重建缺失块。采用*asymmetric*编码器-解码器设计：编码器仅处理可见图像块子集（不含掩码标记），而解码器则轻量级且能够从潜在表示与掩码标记中重建输入（图1）。在我们的非对称编码器-解码器结构中将掩码标记转移到小型解码器，可大幅减少计算量。这种设计下，极高的掩码比例（如75%）能实现双赢：既优化了精度，又让编码器只需处理少量图像块（如25%）。这能使整体预训练时间减少3倍以上，并同步降低内存消耗，使我们能够轻松将MAE扩展至大型模型。

我们的MAE能够学习具有极高容量且泛化能力优异的模型。通过MAE预训练，我们可以在ImageNet-1K数据集上成功训练ViT-Large/-Huge[16]这类数据饥渴型模型，并显著提升其泛化性能。使用标准ViT-Huge模型时，在ImageNet-1K上进行微调可获得87.8%的准确率，这一结果超越了所有仅使用ImageNet-1K数据的既往研究成果。我们还在目标检测、实例分割和语义分割任务上进行了迁移学习评估。在这些任务中，我们的预训练方法均优于有监督预训练方案，更重要的是，我们观察到通过扩大模型规模可带来显著性能提升。这些发现与自然语言处理领域自监督预训练的趋势相吻合[14,47,48,4]，我们期待这将推动计算机视觉领域探索类似的发展路径。

Figure 4. Reconstructions of ImageNet *validation* images using an MAE pre-trained with a masking ratio of 75% but applied on inputs with higher masking ratios. The predictions differ plausibly from the original images, showing that the method can generalize.

2. Related Work

Masked language modeling and its autoregressive counterparts, *e.g.*, BERT [14] and GPT [47, 48, 4], are highly successful methods for pre-training in NLP. These methods hold out a portion of the input sequence and train models to predict the missing content. These methods have been shown to scale excellently [4] and a large abundance of evidence indicates that these pre-trained representations generalize well to various downstream tasks.

Autoencoding is a classical method for learning representations. It has an encoder that maps an input to a latent representation and a decoder that reconstructs the input. For example, PCA and k-means are autoencoders [29]. Denoising autoencoders (DAE) [58] are a class of autoencoders that corrupt an input signal and learn to reconstruct the original, uncorrupted signal. A series of methods can be thought of as a generalized DAE under different corruptions, *e.g.*, masking pixels [59, 46, 6] or removing color channels [70]. Our MAE is a form of denoising autoencoding, but different from the classical DAE in numerous ways.

Masked image encoding methods learn representations from images corrupted by masking. The pioneering work of [59] presents masking as a noise type in DAE. Context Encoder [46] inpaints large missing regions using convolutional networks. Motivated by the success in NLP, related recent methods [6, 16, 2] are based on Transformers [57]. iGPT [6] operates on sequences of pixels and predicts unknown pixels. The ViT paper [16] studies masked patch prediction for self-supervised learning. Most recently, BEiT [2] proposes to predict discrete tokens [44, 50].

Self-supervised learning approaches have seen significant interest in computer vision, often focusing on different pretext tasks for pre-training [15, 61, 42, 70, 45, 17]. Recently, contrastive learning [3, 22] has been popular, *e.g.*, [62, 43, 23, 7], which models image similarity and dissimilarity (or only similarity [21, 8]) between two or more views. Contrastive and related methods strongly depend on data augmentation [7, 21, 8]. Autoencoding pursues a conceptually different direction, and it exhibits different behaviors as we will present.

3. Approach

Our masked autoencoder (MAE) is a simple autoencoding approach that reconstructs the original signal given its partial observation. Like all autoencoders, our approach has an encoder that maps the observed signal to a latent representation, and a decoder that reconstructs the original signal from the latent representation. Unlike classical autoencoders, we adopt an *asymmetric* design that allows the encoder to operate only on the partial, observed signal (without mask tokens) and a lightweight decoder that reconstructs the full signal from the latent representation and mask tokens. Figure 1 illustrates the idea, introduced next.

Masking. Following ViT [16], we divide an image into regular non-overlapping patches. Then we sample a subset of patches and mask (*i.e.*, remove) the remaining ones. Our sampling strategy is straightforward: we sample random patches without replacement, following a uniform distribution. We simply refer to this as "random sampling".

Random sampling with a *high* masking ratio (*i.e.*, the ratio of removed patches) largely eliminates redundancy, thus creating a task that cannot be easily solved by extrapolation from visible neighboring patches (see Figures 2 – 4). The uniform distribution prevents a potential center bias (*i.e.*, more masked patches near the image center). Finally, the highly sparse input creates an opportunity for designing an efficient encoder, introduced next.

MAE encoder. Our encoder is a ViT [16] but applied only on *visible, unmasked patches*. Just as in a standard ViT, our encoder embeds patches by a linear projection with added positional embeddings, and then processes the resulting set via a series of Transformer blocks. However, our encoder only operates on a small subset (*e.g.*, 25%) of the full set. Masked patches are removed; no mask tokens are used. This allows us to train very large encoders with only a fraction of compute and memory. The full set is handled by a lightweight decoder, described next.

MAE decoder. The input to the MAE decoder is the full set of tokens consisting of (i) encoded visible patches, and (ii) mask tokens. See Figure 1. Each mask token [14] is a shared, learned vector that indicates the presence of a miss-

图4. 使用以75%掩码率预训练的MAE对ImageNet *validation*图像进行更高掩码率输入下的重建效果。预测结果与原始图像存在合理差异，表明该方法具有良好的泛化能力。

2. 相关工作

掩码语言模型及其自回归对应方法，{v*}，如BERT [14]和GPT [47, 48, 4]，是自然语言处理中非常成功的预训练方法。这些方法会遮蔽部分输入序列，并训练模型预测缺失内容。研究证明这些方法具备出色的可扩展性[4]，大量证据表明这些预训练表征能够很好地泛化到各种下游任务中。

自编码是一种经典的学习表示的方法。它包含一个将输入映射到潜在表示的编码器，以及一个重建输入的解码器。例如，PCA和k-means都属于自编码器[29]。去噪自编码器(DAE)[58]是一类通过破坏输入信号并学习重建原始未损坏信号的自编码器。一系列方法可被视为不同破坏形式下的广义DAE，{v*}，例如掩码像素[59,46,6]或移除颜色通道[70]。我们的MAE属于去噪自编码形式，但在诸多方面与经典DAE存在差异。

掩码图像编码方法通过从被掩码破坏的图像中学习表示。[59]的开创性工作将掩码作为DAE中的一种噪声类型。Context Encoder[46]使用卷积网络修复大面积缺失区域。受自然语言处理领域成功的启发，相关近期方法[6,16,2]均基于Transformer架构[57]。iGPT[6]在像素序列上进行操作并预测未知像素。ViT论文[16]研究了掩码补丁预测用于自监督学习。最新的BEiT[2]提出预测离散标记[44,50]的方法。

自监督学习方法在计算机视觉领域受到广泛关注，通常侧重于不同的预文本任务进行预训练[15, 61, 42, 70, 45, 17]。近年来对比学习[v3][62, 43, 23, 7]逐渐流行，该方法通过两个或多个视图对图像相似性和差异性（或仅相似性[21, 8]）进行建模。对比学习及相关方法高度依赖数据增强技术[7, 21, 8]。自编码则追求概念上不同的方向，正如我们将要展示的，它呈现出不同的行为特性。

3. 方法

我们的掩码自编码器（MAE）是一种简单的自编码方法，它通过部分观测信号来重建原始信号。与所有自编码器类似，我们的方法包含一个将观测信号映射到潜在表示的编码器，以及一个从潜在表示重建原始信号的解码器。与经典自编码器不同，我们采用{v*}设计：编码器仅处理部分观测信号（不含掩码标记），而轻量级解码器则根据潜在表示和掩码标记来完整重建信号。图1展示了该原理，下文将详细说明。

掩码。遵循ViT[16]的方法，我们将图像划分为规则的非重叠图像块。随后对部分图像块进行采样，并对其余图像块进行掩码（*i.e*，即移除）。我们的采样策略非常直接：按照均匀分布进行无放回的随机图像块采样。我们将其简称为“随机采样”。

以*high*的掩码比例（*i.e*，即被移除图像块的比例）进行随机采样，可有效消除冗余性，从而构建出无法通过可见相邻图像块的外推轻易解决的任务（见图2-4）。均匀分布避免了潜在的中心偏差（*i.e*，即图像中心附近出现更多掩码块的情况）。最终，高度稀疏的输入为设计高效编码器创造了条件——这正是我们接下来要介绍的内容。

MAE编码器。我们的编码器是一个ViT[16]，但仅应用于*visible, unmasked patches*。与标准ViT相同，我们的编码器通过线性投影（添加位置嵌入）来嵌入图像块，然后通过一系列Transformer块处理结果集。然而，我们的编码器仅作用于完整集合的小子集（*e.g*，即25%）。被遮蔽的图像块会被移除；不使用遮蔽标记。这使得我们能够仅用部分计算量和内存来训练非常大的编码器。完整集合由轻量级解码器处理，详见下文。

MAE解码器。MAE解码器的输入是完整令牌集，包含：(i) 已编码可见图像块，以及(ii) 掩码令牌。参见图1。每个掩码令牌[14]是一个共享的学习向量，用于表示缺失内容的存在——

ing patch to be predicted. We add positional embeddings to all tokens in this full set; without this, mask tokens would have no information about their location in the image. The decoder has another series of Transformer blocks.

The MAE decoder is only used during pre-training to perform the image reconstruction task (only the encoder is used to produce image representations for recognition). Therefore, the decoder architecture can be flexibly designed in a manner that is *independent* of the encoder design. We experiment with very small decoders, narrower and shallower than the encoder. For example, our default decoder has $<10\%$ computation per token *vs*. the encoder. With this asymmetrical design, the full set of tokens are only processed by the lightweight decoder, which significantly reduces pre-training time.

Reconstruction target. Our MAE reconstructs the input by predicting the *pixel* values for each masked patch. Each element in the decoder's output is a vector of pixel values representing a patch. The last layer of the decoder is a linear projection whose number of output channels equals the number of pixel values in a patch. The decoder's output is reshaped to form a reconstructed image. Our loss function computes the mean squared error (MSE) between the reconstructed and original images in the pixel space. We compute the loss only on masked patches, similar to BERT [14].[1]

We also study a variant whose reconstruction target is the normalized pixel values of each masked patch. Specifically, we compute the mean and standard deviation of all pixels in a patch and use them to normalize this patch. Using normalized pixels as the reconstruction target improves representation quality in our experiments.

Simple implementation. Our MAE pre-training can be implemented efficiently, and importantly, does not require any specialized sparse operations. First we generate a token for every input patch (by linear projection with an added positional embedding). Next we *randomly shuffle* the list of tokens and *remove* the last portion of the list, based on the masking ratio. This process produces a small subset of tokens for the encoder and is equivalent to sampling patches without replacement. After encoding, we append a list of mask tokens to the list of encoded patches, and *unshuffle* this full list (inverting the random shuffle operation) to align all tokens with their targets. The decoder is applied to this full list (with positional embeddings added). As noted, no sparse operations are needed. This simple implementation introduces negligible overhead as the shuffling and unshuffling operations are fast.

[1]Computing the loss only on masked patches differs from traditional denoising autoencoders [58] that compute the loss on all pixels. This choice is purely result-driven: computing the loss on all pixels leads to a slight decrease in accuracy (*e.g.*, ~0.5%).

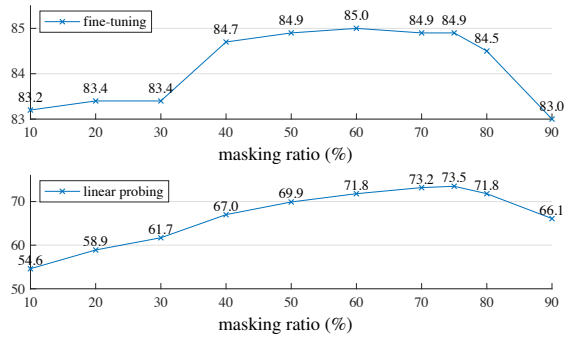

Figure 5. **Masking ratio**. A high masking ratio (75%) works well for both fine-tuning (top) and linear probing (bottom). The y-axes are ImageNet-1K validation accuracy (%) in all plots in this paper.

4. ImageNet Experiments

We do self-supervised pre-training on the ImageNet-1K (IN1K) [13] training set. Then we do supervised training to evaluate the representations with (i) end-to-end fine-tuning or (ii) linear probing. We report top-1 validation accuracy of a single 224×224 crop. Details are in Appendix A.1.

Baseline: ViT-Large. We use ViT-Large (ViT-L/16) [16] as the backbone in our ablation study. ViT-L is very big (an order of magnitude bigger than ResNet-50 [25]) and tends to overfit. The following is a comparison between ViT-L trained from scratch *vs*. fine-tuned from our baseline MAE:

scratch, original [16]	scratch, our impl.	baseline MAE
76.5	82.5	84.9

We note that it is nontrivial to train *supervised* ViT-L from scratch and a good recipe with strong regularization is needed (82.5%, see Appendix A.2). Even so, our MAE pre-training contributes a big improvement. Here fine-tuning is only for 50 epochs (*vs*. 200 from scratch), implying that the fine-tuning accuracy heavily depends on pre-training.

4.1. Main Properties

We ablate our MAE using the default settings in Table 1 (see caption). Several intriguing properties are observed.

Masking ratio. Figure 5 shows the influence of the masking ratio. The optimal ratios are surprisingly high. The ratio of 75% is good for both linear probing and fine-tuning. This behavior is in contrast with BERT [14], whose typical masking ratio is 15%. Our masking ratios are also much higher than those in related works [6, 16, 2] in computer vision (20% to 50%).

The model *infers* missing patches to produce different, yet plausible, outputs (Figure 4). It makes sense of the gestalt of objects and scenes, which cannot be simply completed by extending lines or textures. We hypothesize that this reasoning-like behavior is linked to the learning of useful representations.

Figure 5 also shows that linear probing and fine-tuning results follow *different* trends. For linear probing, the ac-

待预测的补丁。我们为这完整集合中的所有标记添加位置嵌入；若不如此，掩码标记将无法获知自身在图像中的位置信息。解码器则包含另一个系列的Transformer模块。

MAE解码器仅在预训练期间用于执行图像重建任务（只有编码器用于生成图像表示以进行识别）。因此，解码器架构可以灵活设计，使其与编码器设计保持*independent*。我们尝试了非常小的解码器，比编码器更窄更浅。例如，我们的默认解码器对每个token的计算量仅为编码器的<10%*vs*。通过这种非对称设计，完整token集仅由轻量级解码器处理，这显著减少了预训练时间。

重建目标。我们的MAE通过预测每个被遮蔽补丁的*pixel*值来重建输入。解码器输出中的每个元素是一个代表补丁的像素值向量。解码器的最后一层是线性投影，其输出通道数等于补丁中的像素值数量。解码器的输出被重新塑形以形成重建图像。我们的损失函数计算像素空间中重建图像与原始图像之间的均方误差（MSE）。与BERT[14]类似，我们仅针对被遮蔽补丁计算损失。[1]

我们还研究了一种变体，其重建目标是每个掩码补丁的归一化像素值。具体而言，我们计算补丁中所有像素的均值和标准差，并用它们对该补丁进行归一化处理。实验表明，使用归一化像素作为重建目标能提升表征质量。

简单实现。我们的MAE预训练可以高效实现，且重要的是不需要任何专门的稀疏操作。首先我们为每个输入图像块生成一个标记（通过线性投影并添加位置嵌入）。接着我们根据掩码比例对标记列表进行*randomly shuffle*并*remove*列表的最后部分。这个过程为编码器生成一个小的标记子集，相当于无放回地采样图像块。编码完成后，我们将掩码标记列表附加到已编码图像块列表后，并*unshuffle*这个完整列表（反转随机打乱操作）以使所有标记与其目标对齐。解码器应用于这个完整列表（添加了位置嵌入）。如前所述，不需要稀疏操作。这种简单实现引入的开销可忽略不计，因为打乱和反打乱操作速度很快。

[1]Computing the loss only on masked patches differs from traditional denoising autoencoders [58] that compute the loss on all pixels. This choice is purely result-driven: computing the loss on all pixels leads to a slight decrease in accuracy (*e.g.*, ~0.5%).

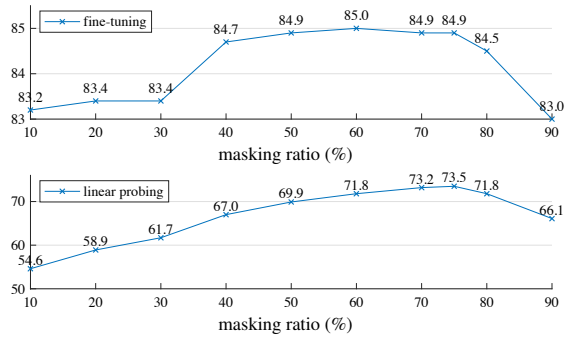

图5. 掩码比例。高掩码比例（75%）在微调（上）和线性探测（下）中均表现良好。本文所有图表中y轴均为ImageNet-1K验证准确率（%）。

4. ImageNet实验

我们在ImageNet-1K（IN1K）[13]训练集上进行自监督预训练，随后通过（i）端到端微调或（ii）线性探测两种方式进行监督训练以评估表征效果。报告采用224{v*}224单裁剪图的Top-1验证准确率，详细设置见附录A.1。

基线：ViT-Large。我们在消融研究中使用ViT-Large（ViT-L/16）[16]作为骨干网络。ViT-L规模非常庞大（比ResNet-50[25]大一个数量级）且容易过拟合。以下是从零开始训练的ViT-L *vs* 与基于我们MAE基线进行微调的对比：

scratch, original [16]	scratch, our impl.	baseline MAE
76.5	82.5	84.9

我们注意到从头开始训练*supervised* ViT-L并非易事，需要采用具有强正则化的优质训练方案（82.5%，参见附录A.2）。即便如此，我们的MAE预训练仍带来了显著提升。此处微调仅进行50个周期（*vs*，而从头训练需200周期），这表明微调精度在很大程度上依赖于预训练效果。

4.1. 主要性质

我们在表1中使用默认设置对MAE进行了消融实验（见标题）。观察到几个有趣的性质。

掩码比例。图5展示了掩码比例的影响。最佳比例出人意料地偏高。75%的比例在线性探测和微调中均表现良好。这一特性与BERT[14]形成鲜明对比，其典型掩码比例为15%。我们的掩码比例也远高于计算机视觉相关研究[6,16,2]中采用的比例（20%至50%）。

模型*infers*缺失补丁以产生不同但合理的输出（图4）。它能够理解物体和场景的整体形态，这种形态无法简单地通过延伸线条或纹理来完成。我们推测这种类推理行为与学习有用表征之间存在关联。

图5还表明，线性探测和微调结果遵循*different*趋势。在线性探测中，ac-

blocks	ft	lin
1	84.8	65.5
2	**84.9**	70.0
4	**84.9**	71.9
8	**84.9**	**73.5**
12	84.4	73.3

(a) **Decoder depth**. A deep decoder can improve linear probing accuracy.

dim	ft	lin
128	**84.9**	69.1
256	84.8	71.3
512	**84.9**	**73.5**
768	84.4	73.1
1024	84.3	73.1

(b) **Decoder width**. The decoder can be narrower than the encoder (1024-d).

case	ft	lin	FLOPs
encoder w/ `[M]`	84.2	59.6	3.3×
encoder w/o `[M]`	**84.9**	**73.5**	**1**×

(c) **Mask token**. An encoder without mask tokens is more accurate and faster (Table 2).

case	ft	lin
pixel (w/o norm)	84.9	73.5
pixel (w/ norm)	**85.4**	**73.9**
PCA	84.6	72.3
dVAE token	85.3	71.6

(d) **Reconstruction target**. Pixels as reconstruction targets are effective.

case	ft	lin
none	84.0	65.7
crop, fixed size	84.7	73.1
crop, rand size	**84.9**	**73.5**
crop + color jit	84.3	71.9

(e) **Data augmentation**. Our MAE works with minimal or no augmentation.

case	ratio	ft	lin
random	75	**84.9**	**73.5**
block	50	83.9	72.3
block	75	82.8	63.9
grid	75	84.0	66.0

(f) **Mask sampling**. Random sampling works the best. See Figure 6 for visualizations.

Table 1. **MAE ablation experiments** with ViT-L/16 on ImageNet-1K. We report fine-tuning (ft) and linear probing (lin) accuracy (%). If not specified, the default is: the decoder has depth 8 and width 512, the reconstruction target is unnormalized pixels, the data augmentation is random resized cropping, the masking ratio is 75%, and the pre-training length is 800 epochs. Default settings are marked in gray.

curacy increases steadily with the masking ratio until the sweet point: the accuracy gap is up to ~20% (54.6% *vs.* 73.5%). For fine-tuning, the results are less sensitive to the ratios, and a wide range of masking ratios (40–80%) work well. All fine-tuning results in Figure 5 are better than training from scratch (82.5%).

Decoder design. Our MAE decoder can be flexibly designed, as studied in Table 1a and 1b.

Table 1a varies the decoder depth (number of Transformer blocks). A sufficiently deep decoder is important for linear probing. This can be explained by the gap between a pixel reconstruction task and a recognition task: the last several layers in an autoencoder are more specialized for reconstruction, but are less relevant for recognition. A reasonably deep decoder can account for the reconstruction specialization, leaving the latent representations at a more abstract level. This design can yield up to 8% improvement in linear probing (Table 1a, 'lin'). However, if fine-tuning is used, the last layers of the encoder can be tuned to adapt to the recognition task. The decoder depth is less influential for improving fine-tuning (Table 1a, 'ft').

Interestingly, our MAE with a *single*-block decoder can perform strongly with fine-tuning (84.8%). Note that a single Transformer block is the minimal requirement to propagate information from visible tokens to mask tokens. Such a small decoder can further speed up training.

In Table 1b we study the decoder width (number of channels). We use 512-d by default, which performs well under fine-tuning and linear probing. A narrower decoder also works well with fine-tuning.

Overall, our default MAE decoder is lightweight. It has 8 blocks and a width of 512-d (gray in Table 1). It only has 9% FLOPs per token *vs.* ViT-L (24 blocks, 1024-d). As such, while the decoder processes all tokens, it is still a small fraction of the overall compute.

encoder	dec. depth	ft acc	hours	speedup
ViT-L, w/ `[M]`	8	84.2	42.4	-
ViT-L	8	84.9	15.4	2.8×
ViT-L	1	84.8	11.6	**3.7**×
ViT-H, w/ `[M]`	8	-	119.6†	-
ViT-H	8	85.8	34.5	3.5×
ViT-H	1	85.9	29.3	**4.1**×

Table 2. **Wall-clock time** of our MAE training (800 epochs), benchmarked in 128 TPU-v3 cores with TensorFlow. The speedup is relative to the entry whose encoder has mask tokens (gray). The decoder width is 512, and the mask ratio is 75%. †: This entry is estimated by training ten epochs.

Mask token. An important design of our MAE is to skip the mask token `[M]` in the encoder and apply it later in the lightweight decoder. Table 1c studies this design.

If the encoder *uses* mask tokens, it performs *worse*: its accuracy drops by 14% in linear probing. In this case, there is a gap between pre-training and deploying: this encoder has a large portion of mask tokens in its input in pre-training, which does not exist in uncorrupted images. This gap may degrade accuracy in deployment. By removing the mask token from the encoder, we constrain the encoder to always see *real* patches and thus improve accuracy.

Moreover, by skipping the mask token in the encoder, we greatly reduce training computation. In Table 1c, we reduce the overall training FLOPs by 3.3×. This leads to a 2.8× wall-clock speedup in our implementation (see Table 2). The wall-clock speedup is even bigger (3.5–4.1×), for a smaller decoder (1-block), a larger encoder (ViT-H), or both. Note that the speedup can be >4× for a masking ratio of 75%, partially because the self-attention complexity is quadratic. In addition, memory is greatly reduced, which can enable training even larger models or speeding up more by large-batch training. The time and memory efficiency makes our MAE favorable for training very large models.

blocks	ft	lin
1	84.8	65.5
2	**84.9**	70.0
4	**84.9**	71.9
8	**84.9**	**73.5**
12	84.4	73.3

(a)解码器深度。较深的解码器可以提高线性探测的准确性。

dim	ft	lin
128	**84.9**	69.1
256	84.8	71.3
512	**84.9**	**73.5**
768	84.4	73.1
1024	84.3	73.1

(b) 解码器宽度。解码器可以比编码器更窄（1024维）。

case	ft	lin	FLOPs
encoder w/ [M]	84.2	59.6	3.3×
encoder w/o [M]	**84.9**	**73.5**	**1**×

(c) 掩码标记。不使用掩码标记的编码器更精确且速度更快（表2）。

case	ft	lin
pixel (w/o norm)	84.9	73.5
pixel (w/ norm)	**85.4**	**73.9**
PCA	84.6	72.3
dVAE token	85.3	71.6

({dv11} 重建目标。将像素作为重建目标是有效的。

case	ft	lin
none	84.0	65.7
crop, fixed size	84.7	73.1
crop, rand size	**84.9**	**73.5**
crop + color jit	84.3	71.9

(e) 数据增强。我们的MAE只需最少或无需增强即可工作。

case	ratio	ft	lin
random	75	**84.9**	**73.5**
block	50	83.9	72.3
block	75	82.8	63.9
grid	75	84.0	66.0

(f) 掩码采样。随机采样效果最佳。可视化结果见图6。

表 1. 基于 ViT-L/16 在 ImageNet-1K 上进行的 MAE 消融实验。我们报告了微调（ft）和线性探测（lin）的准确率（%）。若无特殊说明，默认设置为：解码器深度为 8、宽度为 512，重建目标为非归一化像素，数据增强采用随机缩放裁剪，掩码比例为 75%，预训练时长为 800 轮。默认设置以灰色标注。

准确率随着掩码比例的增加而稳步提高，直至最佳点：准确率差距最高达~20%（54.6% *vs*对比73.5%）。在微调过程中，结果对掩码比例的敏感度较低，且较宽的掩码比例范围（40%-80%）都能取得良好效果。图5中所有微调结果均优于从头开始训练的效果（82.5%）。

解码器设计。我们的MAE解码器可以灵活设计，如表1a和1b中所研究的那样。

表1a改变了解码器的深度（Transformer块的数量）。足够深的解码器对线性探测至关重要。这可以通过像素重建任务和识别任务之间的差距来解释：自编码器中的最后几层更专门用于重建，但与识别任务的相关性较低。一个合理深度的解码器可以解释重建的专门化，使潜在表示保持在更抽象的层次。这种设计可以在线性探测中带来高达8%的提升（表1a中的“lin”）。然而，如果使用微调，编码器的最后几层可以通过调整来适应识别任务。解码器深度对改善微调的影响较小（表1a中的“ft”）。

有趣的是，我们采用*single*块解码器的MAE通过微调可以表现出强劲性能（84.8%）。值得注意的是，单层Transformer块是将可见令牌信息传播到掩码令牌的最低要求。如此小的解码器还能进一步加速训练。

在表1b中，我们研究了解码器的宽度（通道数）。默认使用512维，该设置在微调和线性探测中表现良好。更窄的解码器在微调场景下同样表现优异。

总体而言，我们的默认MAE解码器是轻量级的。它包含8个模块和512维宽度（表1中灰色标注）。每个令牌仅需9%的FLOPs *vs*，而ViT-L（24模块，1024维）则需更多计算量。因此，尽管解码器需要处理所有令牌，其计算量仍只占整体计算的很小部分。

encoder	dec. depth	ft acc	hours	speedup
ViT-L, w/ [M]	8	84.2	42.4	-
ViT-L	8	84.9	15.4	2.8×
ViT-L	1	84.8	11.6	**3.7**×
ViT-H, w/ [M]	8	-	119.6†	-
ViT-H	8	85.8	34.5	3.5×
ViT-H	1	85.9	29.3	**4.1**×

表2. 我们的MAE训练（800轮）在128个TPU-v3核心配合TensorFlow的实测时间。加速比是相对于编码器含掩码标记（灰色）条目的结果。解码器宽度为512，掩码比例为75%。†：此条目通过十轮训练估算得出。

掩码标记。我们MAE的一个重要设计是在编码器中跳过掩码标记[M]，稍后在轻量级解码器中应用它。表1c研究了这一设计。

如果编码器*uses*掩盖标记，它会执行*worse*：在线性探测中准确率下降14%。在这种情况下，预训练与部署之间存在差距：该编码器在预训练时输入中包含大量掩盖标记，而这些在未损坏图像中并不存在。这种差距可能会降低部署时的准确率。通过从编码器中移除掩盖标记，我们约束编码器始终看到*real*补丁，从而提高了准确率。

此外，通过在编码器中跳过掩码标记，我们大幅减少了训练计算量。在表1c中，我们将总体训练FLOPs降低了3.3×。这使我们的实现获得了2.8×的墙钟加速（见表2）。对于更小的解码器（1模块）、更大的编码器（ViT-H）或两者兼有之情况，墙钟加速效果更为显著（3.5-4.1×）。值得注意的是当掩码率为75%时，加速比可达>4×，部分原因是自注意力机制具有二次复杂度。此外，内存占用大幅减少，这使得训练更大型模型或通过大批次训练进一步加速成为可能。这种时间和内存效率优势使我们的MAE非常适用于训练超大型模型。

Figure 6. **Mask sampling strategies** determine the pretext task difficulty, influencing reconstruction quality and representations (Table 1f). Here each output is from an MAE trained with the specified masking strategy. Left: random sampling (our default). Middle: block-wise sampling [2] that removes large random blocks. Right: grid-wise sampling that keeps one of every four patches. Images are from the validation set.

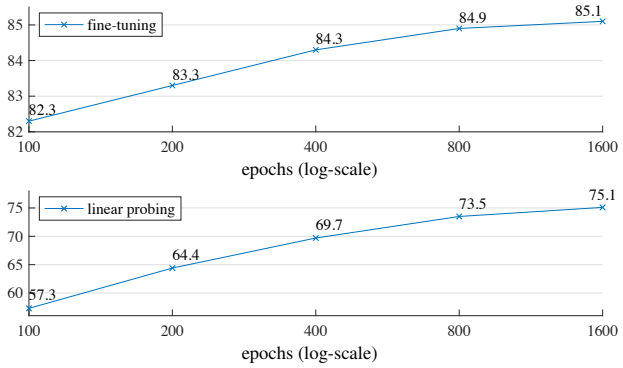

Figure 7. **Training schedules**. A longer training schedule gives a noticeable improvement. Here each point is a full training schedule. The model is ViT-L with the default setting in Table 1.

Reconstruction target. We compare different reconstruction targets in Table 1d. Our results thus far are based on pixels without (per-patch) normalization. Using pixels *with* normalization improves accuracy. This per-patch normalization enhances the contrast locally. In another variant, we perform PCA in the patch space and use the largest PCA coefficients (96 here) as the target. Doing so degrades accuracy. Both experiments suggest that the high-frequency components are useful in our method.

We also compare an MAE variant that predicts *tokens*, the target used in BEiT [2]. Specifically for this variant, we use the DALLE pre-trained dVAE [50] as the tokenizer, following [2]. Here the MAE decoder predicts the token indices using cross-entropy loss. This tokenization improves fine-tuning accuracy by 0.4% *vs*. unnormalized pixels, but has no advantage *vs*. normalized pixels. It also reduces linear probing accuracy. In §5 we further show that tokenization is not necessary in transfer learning.

Our *pixel*-based MAE is much simpler than tokenization. The dVAE tokenizer requires one more pre-training stage, which may depend on extra data (250M images [50]). The dVAE encoder is a large convolutional network (40% FLOPs of ViT-L) and adds nontrivial overhead. Using pixels does not suffer from these problems.

Data augmentation. Table 1e studies the influence of data augmentation on our MAE pre-training.

Our MAE works well using *cropping-only* augmentation, either fixed-size or random-size (both having random horizontal flipping). Adding color jittering degrades the results and so we do not use it in other experiments.

Surprisingly, our MAE behaves decently even if using *no data augmentation* (only center-crop, no flipping). This property is dramatically different from contrastive learning and related methods [62, 23, 7, 21], which heavily rely on data augmentation. It was observed [21] that using cropping-only augmentation reduces the accuracy by 13% and 28% respectively for BYOL [21] and SimCLR [7]. In addition, there is no evidence that contrastive learning can work without augmentation: the two views of an image are the same and can easily satisfy a trivial solution.

In MAE, the role of data augmentation is mainly performed by random masking (ablated next). The masks are different for each iteration and so they generate new training samples regardless of data augmentation. The pretext task is made difficult by masking and requires less augmentation to regularize training.

Mask sampling strategy. In Table 1f we compare different mask sampling strategies, illustrated in Figure 6.

The *block-wise* masking strategy, proposed in [2], tends to remove large blocks (Figure 6 middle). Our MAE with block-wise masking works reasonably well at a ratio of 50%, but degrades at a ratio of 75%. This task is harder than that of random sampling, as a higher training loss is observed. The reconstruction is also blurrier.

We also study *grid-wise* sampling, which regularly keeps one of every four patches (Figure 6 right). This is an easier task and has lower training loss. The reconstruction is sharper. However, the representation quality is lower.

Simple random sampling works the best for our MAE. It allows for a higher masking ratio, which provides a greater speedup benefit while also enjoying good accuracy.

Training schedule. Our ablations thus far are based on 800-epoch pre-training. Figure 7 shows the influence of the training schedule length. The accuracy improves steadily with longer training. Indeed, we have not observed saturation of linear probing accuracy even at 1600 epochs. This behavior is unlike contrastive learning methods, *e.g.*, MoCo v3 [9] saturates at 300 epochs for ViT-L. Note that the MAE encoder only sees 25% of patches per epoch, while in contrastive learning the encoder sees 200% (two-crop) or even more (multi-crop) patches per epoch.

random 75% block 50% grid 75%

图6. 掩码采样策略决定了预训练任务的难度，影响重建质量和表征学习（表1f）。此处每个输出均采用指定掩码策略训练的MAE生成。左：随机采样（我们的默认设置）。中：块状采样[2]，移除大型随机块。右：网格采样，每四个补丁保留一个。图像均来自验证集。

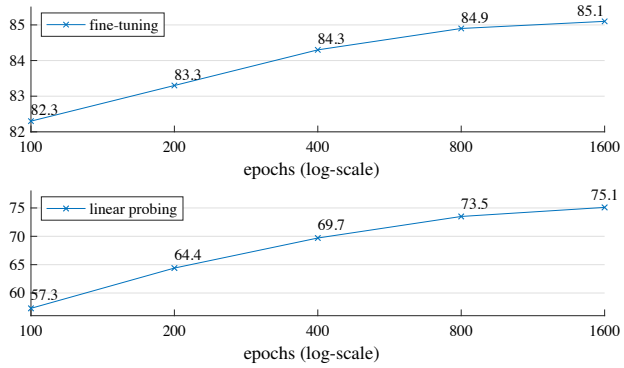

图7：训练计划。更长的训练计划带来显著改进。此处每个点代表一个完整的训练计划。模型为ViT-L，采用表1中的默认设置。

重建目标。我们在表1d中比较了不同的重建目标。到目前为止，我们的结果基于未经（逐块）归一化的像素。使用像素*with*归一化可提高准确率，这种逐块归一化能局部增强对比度。在另一个变体中，我们在块空间执行PCA并使用最大PCA系数（此处为96个）作为目标，但这样做会降低准确率。两项实验均表明，高频成分在我们的方法中具有重要作用。

我们还比较了一种预测BEiT[2]中使用的目标*tokens*的MAE变体。具体针对该变体，我们按照[2]的方法使用DALLE预训练dVAE[50]作为分词器。此处MAE解码器使用交叉熵损失预测词符索引。该分词方式将微调精度提升了0.4%*vs*（非归一化像素），但对*vs*归一化像素没有优势。同时会降低线性探测精度。在§5中我们将进一步证明：在迁移学习中分词处理并非必要。

我们基于*pixel*的MAE比标记化要简单得多。dVAE标记器需要一个额外的预训练阶段，这可能依赖于额外数据（2.5亿张图像[50]）。dVAE编码器是一个大型卷积网络（ViT-L的40% FLOPs），并增加了不小的开销。使用像素则不会遇到这些问题。

数据增强。表1e研究了数据增强对我们的MAE预训练的影响。

我们的MAE在使用*cropping-only*增强（无论是固定尺寸还是随机尺寸，两者都包含随机水平翻转）时表现良好。添加颜色抖动会降低结果质量，因此我们在其他实验中未使用它。

令人惊讶的是，即使仅使用中心裁剪（*no data augmentation* (only center-crop, no flipping)）而不进行翻转，我们的MAE也表现良好。这一特性与对比学习及相关方法[62, 23, 7, 21]形成显著差异——后者极度依赖数据增强。据[21]观察，仅使用裁剪增强会使准确率下降13%

BYOL [21]和SimCLR [7]的比率分别为28%。此外，没有证据表明对比学习可以在没有增强的情况下工作：图像的两个视图是相同的，很容易满足一个平凡的解决方案。

在MAE中，数据增强的作用主要由随机掩码（下文称为"遮蔽"）实现。每次迭代生成的掩码各不相同，因此即使不依赖数据增强技术也能产生新的训练样本。通过掩码使预训练任务变得困难，从而减少对数据增强正则化训练的依赖。

掩码采样策略。在表1f中，我们比较了不同的掩码采样策略（如图6所示）。

*block-wise*掩码策略由[2]提出，倾向于移除大块区域（图6中）。我们采用块状掩码的MAE在50%比例下表现相当不错，但在75%比例时性能下降。该任务比随机采样更具挑战性，因为观察到更高的训练损失。重建结果也更为模糊。

我们还研究了*grid-wise*采样方法，该方法定期保留每四个图像块中的一个（图6右）。这是一个更简单的任务，训练损失较低。重建结果更加清晰锐利，但表征质量有所下降。

简单随机抽样对我们的MAE效果最佳。它允许更高的掩码比例，这不仅提供了更大的加速优势，同时还保持了良好的准确性。

训练计划。我们目前的消融实验基于800个周期的预训练。图7展示了训练时长的影响。随着训练时间的延长，准确率稳步提升。事实上，即使在1600个周期时，我们仍未观察到线性探测准确率出现饱和现象。这种行为与对比学习方法不同，*e.g*例如MoCo v3[9]在ViT-L上300个周期就达到饱和。需要注意的是，MAE编码器每个周期仅处理25%的图像块，而对比学习中的编码器每个周期处理200%（双裁剪）甚至更多（多裁剪）的图像块。

method	pre-train data	ViT-B	ViT-L	ViT-H	ViT-H$_{448}$
scratch, our impl.	-	82.3	82.6	83.1	-
DINO [5]	IN1K	82.8	-	-	-
MoCo v3 [9]	IN1K	83.2	84.1	-	-
BEiT [2]	IN1K+DALLE	83.2	85.2	-	-
MAE	IN1K	83.6	85.9	86.9	**87.8**

Table 3. **Comparisons with previous results on ImageNet-1K**. The pre-training data is the ImageNet-1K training set (except the tokenizer in BEiT was pre-trained on 250M DALLE data [50]). All self-supervised methods are evaluated by end-to-end fine-tuning. The ViT models are B/16, L/16, H/14 [16]. The best for each column is underlined. All results are on an image size of 224, except for ViT-H with an extra result on 448. Here our MAE reconstructs normalized pixels and is pre-trained for 1600 epochs.

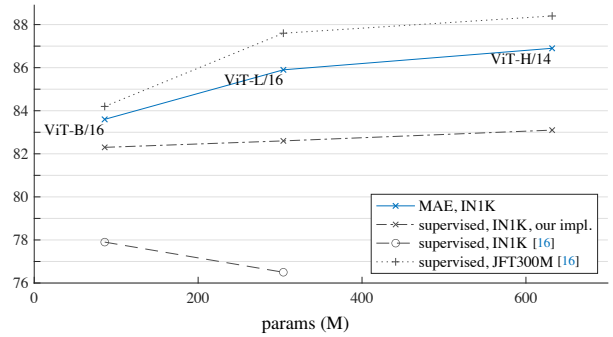

Figure 8. **MAE pre-training *vs*. supervised pre-training**, evaluated by fine-tuning in ImageNet-1K (224 size). We compare with the original ViT results [16] trained in IN1K or JFT300M.

4.2. Comparisons with Previous Results

Comparisons with self-supervised methods. In Table 3 we compare the fine-tuning results of self-supervised ViT models. For ViT-B, all methods perform closely. For ViT-L, the gaps among methods are bigger, suggesting that a challenge for bigger models is to reduce overfitting.

Our MAE can scale up easily and has shown steady improvement from bigger models. We obtain 86.9% accuracy using ViT-H (224 size). By fine-tuning with a 448 size, we achieve **87.8**% accuracy, *using only IN1K data*. The previous best accuracy, among all methods using only IN1K data, is 87.1% (512 size) [67], based on advanced networks. We improve over the state-of-the-art by a nontrivial margin in the highly competitive benchmark of IN1K (no external data). Our result is based on *vanilla* ViT, and we expect advanced networks will perform better.

Comparing with BEiT [2], our MAE is *more accurate* while being *simpler* and *faster*. Our method reconstructs pixels, in contrast to BEiT that predicts tokens: BEiT reported a 1.8% degradation [2] when reconstructing pixels with ViT-B.[2] We do not need dVAE pre-training. Moreover, our MAE is considerably faster (3.5× per epoch) than BEiT, for the reason as studied in Table 1c.

[2]We observed the degradation also in BEiT with ViT-L: it produces 85.2% (tokens) and 83.5% (pixels), reproduced from the official code.

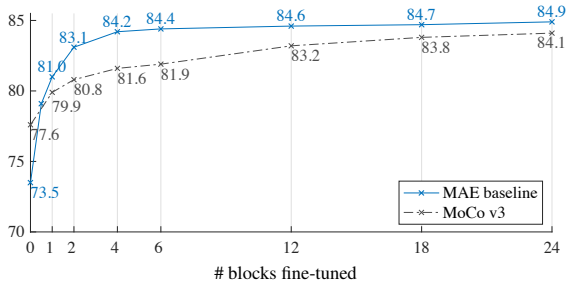

Figure 9. **Partial fine-tuning** results of ViT-L w.r.t. the number of fine-tuned Transformer blocks under the default settings from Table 1. Tuning 0 blocks is linear probing; 24 is full fine-tuning. Our MAE representations are less linearly separable, but are consistently better than MoCo v3 if one or more blocks are tuned.

The MAE models in Table 3 are pre-trained for 1600 epochs for better accuracy (Figure 7). Even so, our total pre-training time is *less* than the other methods when trained on the same hardware. For example, training ViT-L on 128 TPU-v3 cores, our MAE's training time is 31 hours for 1600 epochs and MoCo v3's is 36 hours for 300 epochs [9].

Comparisons with supervised pre-training. In the original ViT paper [16], ViT-L degrades when trained in IN1K. Our implementation of supervised training (see A.2) works better, but accuracy saturates. See Figure 8.

Our MAE pre-training, using only IN1K, can generalize better: the gain over training from scratch is bigger for higher-capacity models. It follows a trend similar to the JFT-300M *supervised* pre-training in [16]. This comparison shows that our MAE can help scale up model sizes.

4.3. Partial Fine-tuning

Table 1 shows that linear probing and fine-tuning results are largely *uncorrelated*. Linear probing has been a popular protocol in the past few years; however, it misses the opportunity of pursuing *strong but non-linear* features—which is indeed a strength of deep learning. As a middle ground, we study a *partial fine-tuning* protocol: fine-tune the last several layers while freezing the others. This protocol was also used in early works, *e.g.*, [65, 70, 42].

Figure 9 shows the results. Notably, fine-tuning only *one* Transformer block boosts the accuracy significantly from 73.5% to 81.0%. Moreover, if we fine-tune only "half" of the last block (*i.e.*, its MLP sub-block), we can get 79.1%, much better than linear probing. This variant is essentially fine-tuning an MLP head. Fine-tuning a few blocks (*e.g.*, 4 or 6) can achieve accuracy close to full fine-tuning.

In Figure 9 we also compare with MoCo v3 [9], a contrastive method with ViT-L results available. MoCo v3 has higher linear probing accuracy; however, all of its partial fine-tuning results are worse than MAE. The gap is 2.6% when tuning 4 blocks. While the MAE representations are less linearly separable, they are stronger *non-linear* features and perform well when a non-linear head is tuned.

method	pre-train data	ViT-B	ViT-L	ViT-H	ViT-H_{448}
scratch, our impl.	-	82.3	82.6	83.1	-
DINO [5]	IN1K	82.8	-	-	-
MoCo v3 [9]	IN1K	83.2	84.1	-	-
BEiT [2]	IN1K+DALLE	83.2	85.2	-	-
MAE	IN1K	83.6	85.9	86.9	**87.8**

表3. 在ImageNet-1K上与先前结果的比较。预训练数据为ImageNet-1K训练集（除BEiT的分词器基于2.5亿DALLE数据[50]预训练外）。所有自监督方法均通过端到端微调进行评估。ViT模型采用B/16、L/16、H/14架构[16]。各列最优结果以下划线标注。除ViT-H额外提供448尺寸结果外，所有结果均基于224图像尺寸。本方法MAE重建归一化像素且经过1600轮预训练。

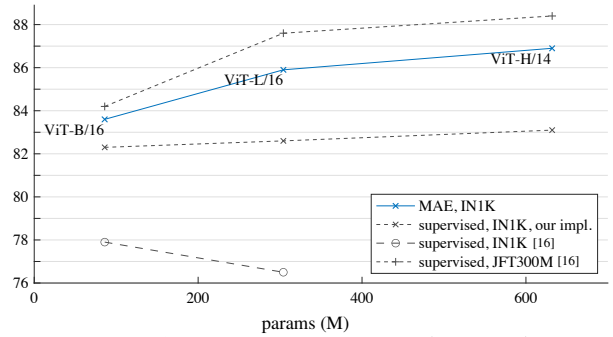

图 8. MAE 预训练 ***vs***。通过 ImageNet-1K（224 尺寸）微调评估的监督式预训练。我们与在 IN1K 或 JFT300M 上训练的原始 ViT 结果 [16] 进行比较。

4.2. 与先前结果的比较

与自监督方法的比较。在表3中，我们比较了自监督ViT模型的微调结果。对于ViT-B，所有方法的表现相近。对于ViT-L，方法之间的差距更大，这表明更大模型面临的挑战是减少过拟合。

我们的MAE可以轻松扩展，并且随着模型增大展现出稳定的性能提升。使用ViT-H（224尺寸）我们获得了86.9%的准确率。通过448尺寸的微调，我们实现了87.8%的准确率，*using only IN1K data*。在所有仅使用IN1K数据的方法中，先前基于先进网络的最佳准确率为87.1%（512尺寸）[67]。在IN1K这个竞争激烈的基准测试中（无外部数据），我们以显著优势超越了现有技术水平。我们的结果基于*vanilla* ViT架构，预计先进网络将获得更好表现。

与BEiT [2]相比，我们的MAE在*more accurate*的同时实现了*simpler*和*faster*。我们的方法重建像素，而BEiT预测标记：BEiT在使用ViT-B重建像素时报告了1.8%的性能下降[2]。[2]我们不需要dVAE预训练。此外，如表1c所研究的，我们的MAE比BEiT快得多（每个周期3.5×）。

[2]We observed the degradation also in BEiT with ViT-L: it produces 85.2% (tokens) and 83.5% (pixels), reproduced from the official code.

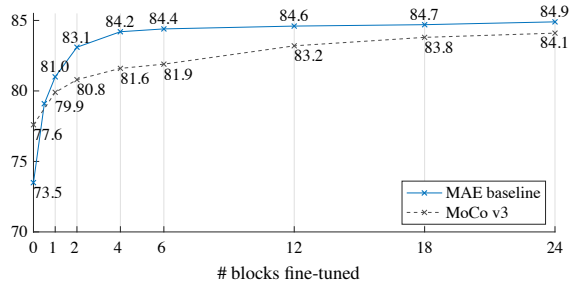

图9. ViT-L在不同微调Transformer块数下的部分微调结果（基于表1默认设置）。微调0个块即线性探测；微调24个块即全参数微调。我们的MAE表征线性可分性较差，但只要微调一个及以上块时，其性能始终优于MoCo v3。

表3中的MAE模型经过1600个epoch的预训练以达到更高精度（图7）。即便如此，在相同硬件条件下，我们的总预训练时间仍比其他方法*less*。例如在128个TPU-v3核心上训练ViT-L时，我们的MAE训练1600个epoch耗时31小时，而MoCo v3训练300个epoch需要36小时[9]。

与监督式预训练的比较。在原始ViT论文[16]中，ViT-L在IN1K数据集上训练时性能下降。我们实现的监督训练（见A.2节）效果更好，但准确率会达到饱和。参见图8。

我们的MAE预训练仅使用IN1K数据，却能展现出更好的泛化能力：对于更高容量的模型，相比从零开始训练带来的性能提升更为显著。这一趋势与文献[16]中JFT-300M *supervised*预训练的效果相似。该对比表明我们的MAE有助于扩展模型规模。

4.3. 部分微调

表1显示，线性探测和微调结果在很大程度上是*uncorrelated*。线性探测在过去几年中一直是一种流行的方案；然而，它错过了追求*strong but non-linear*特征的机会——这确实是深度学习的一个优势。作为折中方案，我们研究了一种*partial fine-tuning*方案：微调最后几层同时冻结其他层。该方案在早期工作中也曾被使用，*e.g*，[65, 70, 42]。

图9展示了结果。值得注意的是，仅微调*one* Transformer模块即可将准确率从73.5%显著提升至81.0%。此外，若仅微调最后一个模块的"半部分"（*i.e*，即其MLP子模块），可获得79.1%的准确率，远优于线性探测方法。该变体本质上是在微调MLP头部。微调少量模块（*e.g*，如4或6个）即可达到接近完整微调的准确率。

在图9中，我们还与MoCo v3[9]进行了比较，这是一个可获得ViT-L结果的对比方法。MoCo v3具有更高的线性探测准确率；然而，其所有部分微调结果均不如MAE。在微调4个区块时，差距达到2.6%。虽然MAE表征的线性可分性较弱，但它们是更强的*non-linear*特征，并且在调整非线性头部时表现优异。

		AP^{box}		AP^{mask}	
method	pre-train data	ViT-B	ViT-L	ViT-B	ViT-L
supervised	IN1K w/ labels	47.9	49.3	42.9	43.9
MoCo v3	IN1K	47.9	49.3	42.7	44.0
BEiT	IN1K+DALLE	49.8	**53.3**	44.4	47.1
MAE	IN1K	**50.3**	**53.3**	**44.9**	**47.2**

Table 4. **COCO object detection and segmentation** using a ViT Mask R-CNN baseline. All entries are based on our implementation. Self-supervised entries use IN1K data *without* labels. Mask AP follows a similar trend as box AP.

These observations suggest that linear separability is not the sole metric for evaluating representation quality. It has also been observed (*e.g.*, [8]) that linear probing is not well correlated with transfer learning performance, *e.g.*, for object detection. To our knowledge, linear evaluation is not often used in NLP for benchmarking pre-training.

5. Transfer Learning Experiments

We evaluate transfer learning in downstream tasks using the pre-trained models in Table 3.

Object detection and segmentation. We fine-tune Mask R-CNN [24] end-to-end on COCO [37]. The ViT backbone is adapted for use with FPN [36] (see A.3). We apply this approach for all entries in Table 4. We report box AP for object detection and mask AP for instance segmentation.

Compared to supervised pre-training, our MAE performs better under all configurations (Table 4). With the smaller ViT-B, our MAE is 2.4 points higher than *supervised* pre-training (50.3 *vs*. 47.9, AP^{box}). More significantly, with the larger ViT-L, our MAE pre-training outperforms supervised pre-training by 4.0 points (53.3 *vs*. 49.3).

The *pixel*-based MAE is better than or on par with the *token*-based BEiT, while MAE is much simpler and faster. Both MAE and BEiT are better than MoCo v3 and MoCo v3 is on par with supervised pre-training.

Semantic segmentation. We experiment on ADE20K [72] using UperNet [63] (see A.4). Table 5 shows that our pre-training significantly improves results over *supervised* pre-training, *e.g.*, by 3.7 points for ViT-L. Our pixel-based MAE also outperforms the token-based BEiT. These observations are consistent with those in COCO.

Classification tasks. Table 6 studies transfer learning on the iNaturalists [56] and Places [71] tasks (see A.5). On iNat, our method shows strong scaling behavior: accuracy improves considerably with bigger models. Our results surpass the previous best results *by large margins*. On Places, our MAE outperforms the previous best results [19, 40], which were obtained via pre-training on billions of images.

Pixels *vs*. tokens. Table 7 compares pixels *vs*. tokens as the MAE reconstruction target. While using dVAE tokens is better than using *unnormalized* pixels, it is statistically similar to using *normalized* pixels across all cases we tested. It again shows that tokenization is not necessary for our MAE.

method	pre-train data	ViT-B	ViT-L
supervised	IN1K w/ labels	47.4	49.9
MoCo v3	IN1K	47.3	49.1
BEiT	IN1K+DALLE	47.1	53.3
MAE	IN1K	**48.1**	**53.6**

Table 5. **ADE20K semantic segmentation** (mIoU) using UperNet. BEiT results are reproduced using the official code. Other entries are based on our implementation. Self-supervised entries use IN1K data *without* labels.

dataset	ViT-B	ViT-L	ViT-H	$ViT\text{-}H_{448}$	prev best
iNat 2017	70.5	75.7	79.3	**83.4**	75.4 [55]
iNat 2018	75.4	80.1	83.0	**86.8**	81.2 [54]
iNat 2019	80.5	83.4	85.7	**88.3**	84.1 [54]
Places205	63.9	65.8	65.9	**66.8**	66.0 [19]†
Places365	57.9	59.4	59.8	**60.3**	58.0 [40]‡

Table 6. **Transfer learning accuracy on classification datasets**, using MAE pre-trained on IN1K and then fine-tuned. We provide system-level comparisons with the previous best results.
†: pre-trained on 1 billion images. ‡: pre-trained on 3.5 billion images.

	IN1K			COCO		ADE20K	
	ViT-B	ViT-L	ViT-H	ViT-B	ViT-L	ViT-B	ViT-L
pixel (w/o norm)	83.3	85.1	86.2	49.5	52.8	48.0	51.8
pixel (w/ norm)	83.6	85.9	86.9	50.3	53.3	48.1	53.6
dVAE token	83.6	85.7	86.9	50.3	53.2	48.1	53.4
△	0.0	-0.2	0.0	0.0	-0.1	0.0	-0.2

Table 7. **Pixels *vs*. tokens** as the MAE reconstruction target. △ is the difference between using dVAE tokens and using normalized pixels. The difference is statistically insignificant.

6. Discussion and Conclusion

Simple algorithms that scale well are the core of deep learning. In NLP, simple self-supervised learning methods (*e.g.*, [47, 14, 48, 4]) enable benefits from exponentially scaling models. In computer vision, practical pre-training paradigms are dominantly supervised (*e.g.* [33, 51, 25, 16]) despite progress in self-supervised learning. In this study, we observe on ImageNet and in transfer learning that an autoencoder—a simple self-supervised method similar to techniques in NLP—provides scalable benefits. Self-supervised learning in vision may now be embarking on a similar trajectory as in NLP.

On the other hand, we note that images and languages are *signals of a different nature* and this difference must be addressed carefully. Images are merely recorded light *without* a semantic decomposition into the visual analogue of words. Instead of attempting to remove objects, we remove random patches that most likely do *not* form a semantic segment. Likewise, our MAE reconstructs pixels, which are *not* semantic entities. Nevertheless, we observe (*e.g.*, Figure 4) that our MAE infers complex, holistic reconstructions, suggesting it has learned numerous visual concepts, *i.e.*, semantics. We hypothesize that this behavior occurs by way of a rich hidden representation inside the MAE. We hope this perspective will inspire future work.

		AP^{box}		AP^{mask}	
method	pre-train data	ViT-B	ViT-L	ViT-B	ViT-L
supervised	IN1K w/ labels	47.9	49.3	42.9	43.9
MoCo v3	IN1K	47.9	49.3	42.7	44.0
BEiT	IN1K+DALLE	49.8	**53.3**	44.4	47.1
MAE	IN1K	**50.3**	**53.3**	**44.9**	**47.2**

表4. 使用ViT Mask R-CNN基线的COCO目标检测与分割结果。所有条目均基于我们的实现。自监督条目使用IN1K数据*without*标签。掩码AP遵循与边界框AP相似的趋势。

这些观察表明，线性可分离性并非评估表示质量的唯一指标。研究还发现（*e.g*，[8]），线性探测与迁移学习性能的相关性不强，*e.g*，例如在目标检测任务中。据我们所知，在自然语言处理领域，线性评估并不常用于预训练模型的基准测试。

5. 迁移学习实验

我们使用表3中的预训练模型评估下游任务中的迁移学习效果。

目标检测与分割。我们在COCO数据集[37]上对Mask R-CNN[24]进行端到端微调。ViT主干网络经过调整可与FPN[36]协同使用（详见A.3节）。表4中所有条目均采用此方法。目标检测报告边界框AP，实例分割报告掩码AP。

与监督式预训练相比，我们的MAE在所有配置下都表现更优（表4）。使用较小的ViT-B时，我们的MAE比*supervised*预训练高出2.4个百分点（50.3 *vs* vs. 47.9，AP^{box}）。更重要的是，使用更大的ViT-L时，我们的MAE预训练以53.3 *vs* vs. 49.3的成绩超越监督式预训练4.0个百分点。

基于*pixel*的MAE优于或与基于*token*的BEiT相当，同时MAE更加简单快速。MAE和BEiT均优于MoCo v3，而MoCo v3与监督预训练效果相当。

语义分割。我们在ADE20K [72]数据集上使用UperNet [63]进行实验（详见A.4节）。表5显示，我们的预训练方法相比*supervised*预训练方法*e.g*有显著提升，ViT-L模型提升了3.7个百分点。基于像素的MAE也优于基于令牌的BEiT，这一发现与COCO数据集上的实验结果一致。

分类任务。表6研究了在iNaturalists [56] 和Places [71]任务上的迁移学习（详见A.5节）。在iNat数据集上，我们的方法展现出强大的扩展能力：模型规模增大时准确率显著提升。我们的结果超越了先前的最佳结果*by large margins*。在Places数据集上，我们的MAE超越了通过数十亿图像预训练获得的先前最佳结果 [19, 40]。

像素 ***vs***。标记。表7比较了像素 *vs*。标记作为MAE重建目标的情况。虽然使用dVAE标记优于使用*unnormalized* 像素，但在我们测试的所有案例中，其统计效果与使用 *normalized* 像素相近。这再次表明，对我们的MAE而言标记化并非必要步骤。

method	pre-train data	ViT-B	ViT-L
supervised	IN1K w/ labels	47.4	49.9
MoCo v3	IN1K	47.3	49.1
BEiT	IN1K+DALLE	47.1	53.3
MAE	IN1K	**48.1**	**53.6**

表5. 使用UperNet的ADE20K语义分割（mIoU）结果。BEiT结果通过官方代码复现得出，其他条目基于我们的实现。自监督条目使用IN1K数据*without*标签。

dataset	ViT-B	ViT-L	ViT-H	ViT-H_{448}	prev best
iNat 2017	70.5	75.7	79.3	**83.4**	75.4 [55]
iNat 2018	75.4	80.1	83.0	**86.8**	81.2 [54]
iNat 2019	80.5	83.4	85.7	**88.3**	84.1 [54]
Places205	63.9	65.8	65.9	**66.8**	66.0 [19]†
Places365	57.9	59.4	59.8	**60.3**	58.0 [40]‡

表6：使用在IN1K上预训练并进行微调的MAE在分类数据集上的迁移学习准确率。我们提供了与先前最佳结果的系统级比较。†: pre-trained on 1 billion images. ‡: pre-trained on 3.5 billion images.

	IN1K			COCO		ADE20K	
	ViT-B	ViT-L	ViT-H	ViT-B	ViT-L	ViT-B	ViT-L
pixel (w/o norm)	83.3	85.1	86.2	49.5	52.8	48.0	51.8
pixel (w/ norm)	83.6	85.9	86.9	50.3	53.3	48.1	53.6
dVAE token	83.6	85.7	86.9	50.3	53.2	48.1	53.4
△	0.0	-0.2	0.0	0.0	-0.1	0.0	-0.2

表7. 使用像素***vs***作为MAE重建目标。△是使用dVAE标记与使用归一化像素之间的差异。该差异在统计学上不显著。

6. 讨论与结论

扩展性良好的简单算法是深度学习的核心。在自然语言处理领域，简单的自监督学习方法（*e.g*，如[47,14,48,4]所示）能够从指数级扩展的模型中获益。尽管自监督学习取得了进展，在计算机视觉领域，实用的预训练范式仍以监督学习为主导（*e.g*，参见[33,51,25,16]）。本研究中，我们在ImageNet和迁移学习任务中发现：自编码器——这种与NLP技术类似的简单自监督方法——能够提供可扩展的性能优势。视觉领域的自监督学习如今可能正踏上与NLP相似的发展轨迹。

另一方面，我们注意到图像与语言具有*signals of a different nature*差异，必须谨慎处理这种差异。图像仅是记录的光信号*without*语义分解为视觉上的词汇类比。我们并非试图移除物体，而是去除最可能*not*构成语义片段的随机图像块。同样地，我们的MAE重建的是*not*语义实体的像素。然而我们观察到（*e.g*，图4），MAE能够推断出复杂且整体的重建结果，表明其已学习到大量视觉概念*i.e*语义信息。我们推测这种行为是通过MAE内部丰富的隐藏表征实现的。希望这一视角能启发未来研究。

Broader impacts. The proposed method predicts content based on learned statistics of the training dataset and as such will reflect biases in those data, including ones with negative societal impacts. The model may generate inexistent content. These issues warrant further research and consideration when building upon this work to generate images.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv:1607.06450*, 2016.

[2] Hangbo Bao, Li Dong, and Furu Wei. BEiT: BERT pre-training of image transformers. *arXiv:2106.08254*, 2021. *Accessed in June 2021*.

[3] Suzanna Becker and Geoffrey E Hinton. Self-organizing neural network that discovers surfaces in random-dot stereograms. *Nature*, 1992.

[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *NeurIPS*, 2020.

[5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021.

[6] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *ICML*, 2020.

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

[8] Xinlei Chen and Kaiming He. Exploring simple Siamese representation learning. In *CVPR*, 2021.

[9] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised Vision Transformers. In *ICCV*, 2021.

[10] Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*, 2020.

[11] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 1995.

[12] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPR Workshops*, 2020.

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

[15] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015.

[16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[17] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018.

[18] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010.

[19] Priya Goyal, Mathilde Caron, Benjamin Lefaudeux, Min Xu, Pengchao Wang, Vivek Pai, Mannat Singh, Vitaliy Liptchinsky, Ishan Misra, Armand Joulin, and Piotr Bojanowski. Self-supervised pretraining of visual features in the wild. *arXiv:2103.01988*, 2021.

[20] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch SGD: Training ImageNet in 1 hour. *arXiv:1706.02677*, 2017.

[21] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent - a new approach to self-supervised learning. In *NeurIPS*, 2020.

[22] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, 2006.

[23] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

[24] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017.

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[26] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *ICCV*, 2021.

[27] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*, 2019.

[28] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *CVPR*, 2021.

[29] Geoffrey E Hinton and Richard S Zemel. Autoencoders, minimum description length, and helmholtz free energy. In *NeurIPS*, 1994.

[30] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *ECCV*, 2016.

[31] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

[32] Insoo Kim, Seungju Han, Ji-won Baek, Seong-Jin Park, Jae-Joon Han, and Jinwoo Shin. Quality-agnostic image recognition via invertible decoder. In *CVPR*, 2021.

[33] Alex Krizhevsky, Ilya Sutskever, and Geoff Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012.

[34] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1989.

[35] Yanghao Li, Saining Xie, Xinlei Chen, Piotr Dollár, Kaiming He, and Ross Girshick. Benchmarking detection transfer learning with vision transformers. *In preparation*, 2021.

更广泛的影响。所提出的方法基于训练数据集学习到的统计规律来预测内容，因此会反映这些数据中的偏见，包括具有负面社会影响的偏见。该模型可能生成不存在的内容。在基于此项工作生成图像时，这些问题需要进一步研究和考量。

参考文献

[1] Jimmy Lei Ba、Jamie Ryan Kiros和Geoffrey E Hinton。层归一化。*arXiv:1607.06450*，2016年。[2] 鲍航博、董力和韦福如。BEiT：图像Transformer的BERT预训练。*arXiv:2106.08254*，2021年。*Accessed in June 2021*。[3] Suzanna Becker和Geoffrey E Hinton。自组织神经网络在随机点立体图中发现表面。*Na- ture*，1992年。[4] Tom Brown、Benjamin Mann、Nick Ryder、Melanie Subbiah、Jared D Kaplan、Prafulla Dhariwal、Arvind Neelakantan、Pranav Shyam、Girish Sastry、Amanda Askell、Sandhini Agarwal、Ariel Herbert-Voss、Gretchen Krueger、Tom Henighan、Rewon Child、Aditya Ramesh、Daniel Ziegler、Jeffrey Wu、Clemens Winter、Chris Hesse、Mark Chen、Eric Sigler、Mateusz Litwin、Scott Gray、Benjamin Chess、Jack Clark、Christopher Berner、Sam McCandlish、Alec Radford、Ilya Sutskever和Dario Amodei。语言模型是小样本学习者。载于*NeurIPS*，2020年。[5] Mathilde Caron、Hugo Touvron、Ishan Misra、Hervé Jégou、Julien Mairal、Piotr Bojanowski和Armand Joulin。自监督视觉Transformer的新兴特性。载于*ICCV*，2021年。[6] 陈马克、Alec Radford、Rewon Child、Jeffrey Wu、Heewoo Jun、David Luan和Ilya Sutskever。基于像素的生成式预训练。载于*ICML*，2020年。[7] 陈霆、Simon Kornblith、Mohammad Norouzi和Geoffrey Hinton。视觉表示对比学习的简单框架。载于*ICML*，2020年。[8] 陈新雷和何恺明。探索简单孪生表示学习。载于*CVPR*，2021年。[9] 陈新雷、谢赛宁和何恺明。自监督视觉Transformer训练的实证研究。载于*ICCV*，2021年。[10] Kevin Clark、Minh-Thang Luong、Quoc V Le和Christopher D Manning。ELECTRA：将文本编码器作为判别器而非生成器进行预训练。载于*ICLR*，2020年。[11] Corinna Cortes和Vladimir Vapnik。支持向量网络。*Machine learning*，1995年。[12] Ekin D Cubuk、Barret Zoph、Jonathon Shlens和Quoc V Le。Randaugment：具有简化搜索空间的实用自动数据增强。载于*CVPR Workshops*，2020年。[13] 邓嘉、董伟、Richard Socher、李飞飞团队Kai Li与Li-Jia Li。ImageNet：大规模分层图像数据库。载于*CVPR*，2009年。[14] Jacob Devlin、张明伟、Kenton Lee和Kristina Toutanova。BERT：用于语言理解的深度双向Transformer预训练。载于*NAACL*，2019年。[15] Carl Doersch、Abhinav Gupta和Alexei A Efros。通过上下文预测的无监督视觉表示学习。载于*ICCV*，2015年。[16] Alexey Dosovitskiy、Lucas Beyer、Alexander Kolesnikov、Dirk Weissenborn、翟晓华、Thomas Unterthiner、Mostafa Dehghani、Matthias Minderer、Georg Heigold、Sylvain Gelly、Jakob Uszkoreit与Neil Houlsby。一幅图像价值16x16词汇：大规模图像识别中的Transformer模型。发表于*ICLR*，2021年。[17] Spyros Gidaris, Praveer Singh与Nikos Komodakis。通过预测图像旋转实现无监督表征学习。发表于*ICLR*，2018年。[18] Xavier Glorot与Yoshua Bengio。理解深度前馈神经网络训练难点。发表于*AISTATS*，2010年。[19] Priya Goyal, Mathilde Caron, Benjamin Lefaudeux, Min Xu, Pengchao Wang, Vivek Pai, Mannat Singh, Vitaliy Liptchinsky, Ishan Misra, Armand Joulin与Piotr Bojanowski。真实场景下的视觉特征自监督预训练。*arXiv:2103.01988*，2021年。[20] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia与Kaiming He。精准大规模小批量SGD：1小时完成ImageNet训练。*arXiv:1706.02677*，2017年。[21] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Remi Munos与Michal Valko。潜在空间自举——自监督学习新方法。发表于*NeurIPS*，2020年。[22] Raia Hadsell, Sumit Chopra与Yann LeCun。通过学习不变映射实现降维。发表于*CVPR*，2006年。[23] 何恺明, Haoqi Fan, 吴育昕与Ross Girshick。无监督视觉表征学习的动量对比方法。发表于*CVPR*，2020年。[24] 何恺明, Georgia Gkioxari, Piotr Dollár与Ross Girshick。Mask R-CNN。发表于*ICCV*，2017年。[25] 何恺明, 张祥雨, 任少卿与孙剑。图像识别中的深度残差学习。发表于*CVPR*，2016年。[26] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo等。鲁棒性的多面性：分布外泛化的批判性分析。发表于*ICCV*，2021年。[27] Dan Hendrycks与Thomas Dietterich。神经网络对常见损坏与扰动的鲁棒性基准测试。发表于*ICLR*，2019年。[28] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt与Dawn Song。自然对抗样本。发表于*CVPR*，2021年。[29] Geoffrey E Hinton与Richard S Zemel。自编码器、最小描述长度与亥姆霍兹自由能。发表于*NeurIPS*，1994年。[30] 黄高, 孙煜, 刘壮, Daniel Sedra与Kilian Q Weinberger。随机深度深度网络。发表于*ECCV*，2016年。[31] Sergey Ioffe与Christian Szegedy。批量归一化：通过减少内部协变量偏移加速深度网络训练。发表于*ICML*，2015年。[32] Insoo Kim, Seungju Han, Ji-won Baek, Seong-Jin Park, Jae-Joon Han与Jinwoo Shin。通过可逆解码器实现质量无关图像识别。发表于*CVPR*，2021年。[33] Alex Krizhevsky, Ilya Sutskever与Geoff Hinton。基于深度卷积神经网络的ImageNet分类。发表于*NeurIPS*，2012年。[34] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard与Lawrence D Jackel。反向传播算法在手写邮政编码识别中的应用。*Neu-ral computation*，1989年。[35] 李阳皓, 谢赛宁, 陈鑫磊, Piotr Dollár, 何恺明与Ross Girshick。基于视觉Transformer的检测迁移学习基准测试。*In preparation*，2021年。

[36] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

[37] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014.

[38] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *ICLR*, 2017.

[39] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

[40] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *ECCV*, 2018.

[41] Xiaofeng Mao, Gege Qi, Yuefeng Chen, Xiaodan Li, Ranjie Duan, Shaokai Ye, Yuan He, and Hui Xue. Towards robust vision transformer. *arXiv:2105.07926*, 2021.

[42] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016.

[43] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv:1807.03748*, 2018.

[44] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *NeurIPS*, 2017.

[45] Deepak Pathak, Ross Girshick, Piotr Dollár, Trevor Darrell, and Bharath Hariharan. Learning features by watching objects move. In *CVPR*, 2017.

[46] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016.

[47] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

[48] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[49] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 2020.

[50] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *ICML*, 2021.

[51] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[52] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

[53] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021.

[54] Hugo Touvron, Alexandre Sablayrolles, Matthijs Douze, Matthieu Cord, and Hervé Jégou. Grafit: Learning fine-grained image representations with coarse labels. In *ICCV*, 2021.

[55] Hugo Touvron, Andrea Vedaldi, Matthijs Douze, and Hervé Jégou. Fixing the train-test resolution discrepancy. *arXiv:1906.06423*, 2019.

[56] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The iNaturalist species classification and detection dataset. In *CVPR*, 2018.

[57] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[58] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, 2008.

[59] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *JMLR*, 2010.

[60] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. In *NeurIPS*, 2019.

[61] Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *ICCV*, 2015.

[62] Zhirong Wu, Yuanjun Xiong, Stella Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 2018.

[63] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, 2018.

[64] Tete Xiao, Mannat Singh, Eric Mintun, Trevor Darrell, Piotr Dollár, and Ross Girshick. Early convolutions help transformers see better. In *NeurIPS*, 2021.

[65] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *NeurIPS*, 2014.

[66] Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv:1708.03888*, 2017.

[67] Li Yuan, Qibin Hou, Zihang Jiang, Jiashi Feng, and Shuicheng Yan. VOLO: Vision outlooker for visual recognition. *arXiv:2106.13112*, 2021.

[68] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 2019.

[69] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018.

[70] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, 2016.

[71] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using Places database. In *NeurIPS*, 2014.

[72] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ADE20K dataset. *IJCV*, 2019.

[36] 林致远（Tsung-Yi Lin）、Piotr Dollár、Ross Girshick、何恺明、Bharath Hariharan、Serge Belongie。面向目标检测的特征金字塔网络。发表于*CVPR*，2017年。 [37] 林致远、Michael Maire、Serge Belongie、James Hays、Pietro Perona、Deva Ramanan、Piotr Dollár、C Lawrence Zitnick。Microsoft COCO：上下文中的常见物体。发表于*ECCV*，2014年。 [38] Ilya Loshchilov、Frank Hutter。SGDR：带热重启的随机梯度下降。发表于*ICLR*，2017年。 [39] Ilya Loshchilov、Frank Hutter。解耦权重衰减正则化。发表于*ICLR*，2019年。 [40] Dhruv Mahajan、Ross Girshick、Vignesh Ramanathan、何恺明、Manohar Paluri、李逸轩、Ashwin Bharambe、Laurens van der Maaten。探索弱监督预训练的极限。发表于*ECCV*，2018年。 [41] 毛晓峰、齐格格、陈月峰、李孝丹、段冉杰、叶少凯、何源、薛辉。迈向鲁棒视觉Transformer。*arXiv:2105.07926*，2021年。 [42] Mehdi Noroozi、Paolo Favaro。通过解决拼图游戏进行无监督视觉表示学习。发表于*ECCV*，2016年。 [43] Aaron van den Oord、Yazhe Li、Oriol Vinyals。基于对比预测编码的表示学习。*arXiv:1807.03748*，2018年。 [44] Aaron van den Oord、Oriol Vinyals、Koray Kavukcuoglu。神经离散表示学习。发表于*NeurIPS*，2017年。 [45] Deepak Pathak、Ross Girshick、Piotr Dollár、Trevor Darrell、Bharath Hariharan。通过观察物体运动学习特征。发表于*CVPR*，2017年。 [46] Deepak Pathak、Philipp Krahenbuhl、Jeff Donahue、Trevor Darrell、Alexei A Efros。上下文编码器：通过修复进行特征学习。发表于*CVPR*，2016年。 [47] Alec Radford、Karthik Narasimhan、Tim Salimans、Ilya Sutskever。通过生成式预训练提升语言理解能力。2018年。 [48] Alec Radford、Jeffrey Wu、Rewon Child、David Luan、Dario Amodei、Ilya Sutskever。语言模型是无监督多任务学习者。2019年。 [49] Colin Raffel、Noam Shazeer、Adam Roberts、Katherine Lee、Sharan Narang、Michael Matena、Yanqi Zhou、Wei Li、Peter J. Liu。基于统一文本到文本Transformer的迁移学习极限探索。*JMLR*，2020年。 [50] Aditya Ramesh、Mikhail Pavlov、Gabriel Goh、Scott Gray、Chelsea Voss、Alec Radford、Mark Chen、Ilya Sutskever。零样本文本到图像生成。发表于*ICML*，2021年。 [51] Karen Simonyan、Andrew Zisserman。面向大规模图像识别的超深卷积网络。发表于*ICLR*，2015年。 [52] Christian Szegedy、Vincent Vanhoucke、Sergey Ioffe、Jonathon Shlens、Zbigniew Wojna。计算机视觉中Inception架构的重新思考。发表于*CVPR*，2016年。 [53] Hugo Touvron、Matthieu Cord、Matthijs Douze、Francisco Massa、Alexandre Sablayrolles、Hervé Jégou。训练数据高效的图像Transformer及注意力蒸馏。发表于*ICML*，2021年。 [54] Hugo Touvron、Alexandre Sablayrolles、Matthijs Douze、Matthieu Cord、Hervé Jégou。Grafit：通过粗标签学习细粒度图像表示。发表于*ICCV*，2021年。 [55] Hugo Touvron、Andrea Vedaldi、Matthijs Douze、Hervé Jégou。修复训练-测试分辨率差异问题。*arXiv:1906.06423*，2019年。

[56] Grant Van Horn、Oisin Mac Aodha、Yang Song、Yin Cui、Chen Sun、Alex Shepard、Hartwig Adam、Pietro Perona与Serge Belongie。iNaturalist物种分类与检测数据集。见*CVPR*，2018年。[57] Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N Gomez、Lukasz Kaiser与Illia Polosukhin。注意力机制即一切。见*NeurIPS*，2017年。[58] Pascal Vincent、Hugo Larochelle、Yoshua Bengio与Pierre-Antoine Manzagol。通过去噪自编码器提取并组合鲁棒特征。见*ICML*，2008年。[59] Pascal Vincent、Hugo Larochelle、Isabelle Lajoie、Yoshua Bengio、Pierre-Antoine Manzagol与Léon Bottou。堆叠去噪自编码器：通过局部去噪准则在深度网络中学习有效表示。*JMLR*，2010年。[60] Haohan Wang、Songwei Ge、Zachary Lipton与Eric P Xing。通过抑制局部预测能力学习鲁棒全局表示。见*NeurIPS*，2019年。[61] Xiaolong Wang与Abhinav Gupta。使用视频进行视觉表示的无监督学习。见*ICCV*，2015年。[62] Zhirong Wu、Yuanjun Xiong、Stella Yu与Dahua Lin。通过非参数实例判别的无监督特征学习。见*CVPR*，2018年。[63] Tete Xiao、Yingcheng Liu、Bolei Zhou、Yuning Jiang与Jian Sun。面向场景理解的统一感知解析。见*ECCV*，2018年。[64] Tete Xiao、Mannat Singh、Eric Mintun、Trevor Darrell、Piotr Dollár与Ross Girshick。早期卷积助力Transformer提升视觉性能。见*NeurIPS*，2021年。[65] Jason Yosinski、Jeff Clune、Yoshua Bengio与Hod Lipson。深度神经网络中的特征可迁移性探究。见*NeurIPS*，2014年。[66] Yang You、Igor Gitman与Boris Ginsburg。卷积网络的大批量训练。*arXiv:1708.03888*，2017年。[67] Li Yuan、Qibin Hou、Zihang Jiang、Jiashi Feng与Shuicheng Yan。VOLO：视觉识别视觉展望器。*arXiv:2106.13112*，2021年。[68] Sangdoo Yun、Dongyoon Han、Seong Joon Oh、Sanghyuk Chun、Junsuk Choe与Youngjoon Yoo。CutMix：通过局部化特征训练强分类器的正则化策略。见*ICCV*，2019年。[69] Hongyi Zhang、Moustapha Cisse、Yann N Dauphin与David Lopez-Paz。mixup：超越经验风险最小化。见*ICLR*，2018年。[70] Richard Zhang、Phillip Isola与Alexei A Efros。彩色图像上色。见*ECCV*，2016年。[71] Bolei Zhou、Agata Lapedriza、Jianxiong Xiao、Antonio Torralba与Aude Oliva。使用Places数据库学习场景识别的深度特征。见*NeurIPS*，2014年。[72] Bolei Zhou、Hang Zhao、Xavier Puig、Tete Xiao、Sanja Fidler、Adela Barriuso与Antonio Torralba。通过ADE20K数据集实现场景语义理解。*IJCV*，2019年。

A. Implementation Details

A.1. ImageNet Experiments

ViT architecture. We follow the standard ViT architecture [16]. It has a stack of Transformer blocks [57], and each block consists of a multi-head self-attention block and an MLP block, both having LayerNorm (LN) [1]. The encoder ends with LN. As the MAE encoder and decoder have different width, we adopt a linear projection layer after the encoder to match it. Our MAE adds positional embeddings [57] (the sine-cosine version) to both the encoder and decoder inputs. Our MAE does *not* use relative position or layer scaling (which are used in the code of [2]).

We extract features from the encoder output for fine-tuning and linear probing. As ViT has a class token [16], to adapt to this design, in our MAE pre-training we append an auxiliary dummy token to the encoder input. This token will be treated as the class token for training the classifier in linear probing and fine-tuning. Our MAE works similarly well without this token (with average pooling).

Pre-training. The default setting is in Table 8. We do *not* use color jittering, drop path, or gradient clip. We use xavier_uniform [18] to initialize all Transformer blocks, following ViT's official code [16]. We use the linear *lr* scaling rule [20]: *lr* = *base_lr*×batchsize / 256.

End-to-end fine-tuning. Our fine-tuning follows common practice of supervised ViT training. The default setting is in Table 9. We use layer-wise *lr* decay [10] following [2].

Linear probing. Our linear classifier training follows [9]. See Table 10. We observe that linear probing requires a very different recipe than end-to-end fine-tuning. In particular, regularization is in general harmful for linear probing. Following [9], we disable many common regularization strategies: we do *not* use mixup [69], cutmix [68], drop path [30], or color jittering, and we set weight decay as zero.

It is a common practice to normalize the classifier input when training a classical linear classifier (*e.g.*, SVM [11]). Similarly, it is beneficial to normalize the pre-trained features when training the linear probing classifier. Following [15], we adopt an extra BatchNorm layer [31] without affine transformation (`affine=False`). This layer is applied on the pre-trained features produced by the encoder, and is before the linear classifier. We note that the layer does *not* break the linear property, and it can be absorbed into the linear classifier after training: it is essentially a re-parameterized linear classifier.[3] Introducing this layer helps calibrate the feature magnitudes across different variants in our ablations, so that they can use the same setting without further *lr* search.

[3] Alternatively, we can pre-compute the mean and std of the features and use the normalized features to train linear classifiers.

config	value
optimizer	AdamW [39]
base learning rate	1.5e-4
weight decay	0.05
optimizer momentum	β_1, β_2=0.9, 0.95 [6]
batch size	4096
learning rate schedule	cosine decay [38]
warmup epochs [20]	40
augmentation	RandomResizedCrop

Table 8. **Pre-training setting.**

config	value
optimizer	AdamW
base learning rate	1e-3
weight decay	0.05
optimizer momentum	β_1, β_2=0.9, 0.999
layer-wise lr decay [10, 2]	0.75
batch size	1024
learning rate schedule	cosine decay
warmup epochs	5
training epochs	100 (B), 50 (L/H)
augmentation	RandAug (9, 0.5) [12]
label smoothing [52]	0.1
mixup [69]	0.8
cutmix [68]	1.0
drop path [30]	0.1 (B/L) 0.2 (H)

Table 9. **End-to-end fine-tuning setting.**

config	value
optimizer	LARS [66]
base learning rate	0.1
weight decay	0
optimizer momentum	0.9
batch size	16384
learning rate schedule	cosine decay
warmup epochs	10
training epochs	90
augmentation	RandomResizedCrop

Table 10. **Linear probing setting.** We use LARS with a large batch for faster training; SGD works similarly with a 4096 batch.

Partial fine-tuning. Our MAE partial fine-tuning (§4.3) follows the setting in Table 9, except that we adjust the number of fine-tuning epochs. We observe that tuning fewer blocks requires a longer schedule. We set the numbers of fine-tuning epochs as {50, 100, 200} and use the optimal one for each number of blocks tuned.

A.2. Supervised Training ViT-L/H from Scratch

We find that it is nontrivial to train *supervised* ViT-L/H *from scratch* on ImageNet-1K. The training is unstable. While there have been strong baselines with publicly available implementations [53] for smaller models, the recipes for the larger ViT-L/H are unexplored. Directly applying the previous recipes to these larger models does not work. A NaN loss is frequently observed during training.

We provide our recipe in Table 11. We use a *wd* of 0.3, a large batch size of 4096, and a long warmup, following the original ViT [16]. We use β_2=0.95 following [6]. We use the regularizations listed in Table 11 and disable others, following [64]. All these choices are for improving training stability. Our recipe can finish training with no NaN loss.

A. 实现细节

A.1. ImageNet 实验

ViT架构。我们遵循标准的ViT架构[16]。它包含堆叠的Transformer块[57]，每个块由多头自注意力块和MLP块组成，两者都采用LayerNorm（LN）[1]。编码器末端设有LN层。由于MAE编码器与解码器具有不同的宽度，我们在编码器后采用线性投影层进行维度匹配。我们的MAE在编码器和解码器输入中都添加了正弦余弦版本的位置嵌入[57]。我们的MAE*not*不使用相对位置编码或层缩放（这些在[2]的代码中被使用）。

我们从编码器输出中提取特征以进行微调和线性探测。由于ViT具有类别标记[16]，为适应这一设计，在MAE预训练中我们向编码器输入添加了一个辅助虚拟标记。该标记在线性探测和微调过程中将被视为训练分类器的类别标记。即使不使用该标记（采用平均池化），我们的MAE同样能良好工作。

预训练。默认设置如表8所示。我们*not*使用颜色抖动、路径丢弃或梯度裁剪。我们使用Xavier uniform [18]初始化所有Transformer模块，遵循ViT官方代码[16]的实现方式。我们采用线性*lr*缩放规则[20]：$lr = base\_lr \times batchsize / 256$。

端到端微调。我们的微调遵循有监督ViT训练的常规做法。默认设置见表9。我们沿用[2]的方法，采用分层*lr*衰减[10]。

线性探测。我们的线性分类器训练遵循[9]。参见表10。我们观察到线性探测需要与端到端微调截然不同的方案。特别是，正则化通常对线性探测有害。遵循[9]的方法，我们禁用了许多常见的正则化策略：我们*not*不使用mixup[69]、cutmix[68]、drop path[30]或颜色抖动，并将权重衰减设置为零。

在训练经典线性分类器（*e.g*，如SVM [11]）时，通常会对分类器输入进行归一化处理。同样地，在训练线性探测分类器时，对预训练特征进行归一化也大有裨益。遵循[15]的做法，我们采用了一个不带仿射变换（affine=False）的额外BatchNorm层[31]。该层作用于编码器产生的预训练特征，且位于线性分类器之前。需要说明的是，该层*not*不会破坏线性特性，并可在训练后被吸收到线性分类器中：本质上这是一个重新参数化的线性分类器[3]。引入该层有助于在我们消融实验中校准不同变体的特征量级，使它们能够使用相同设置而无需额外进行*lr*搜索。

[3]Alternatively, we can pre-compute the mean and std of the features and use the normalized features to train linear classifiers.

config	value
optimizer	AdamW [39]
base learning rate	1.5e-4
weight decay	0.05
optimizer momentum	β_1, β_2=0.9, 0.95 [6]
batch size	4096
learning rate schedule	cosine decay [38]
warmup epochs [20]	40
augmentation	RandomResizedCrop

Table 8. **Pre-training setting.**

config	value
optimizer	AdamW
base learning rate	1e-3
weight decay	0.05
optimizer momentum	β_1, β_2=0.9, 0.999
layer-wise lr decay [10, 2]	0.75
batch size	1024
learning rate schedule	cosine decay
warmup epochs	5
training epochs	100 (B), 50 (L/H)
augmentation	RandAug (9, 0.5) [12]
label smoothing [52]	0.1
mixup [69]	0.8
cutmix [68]	1.0
drop path [30]	0.1 (B/L) 0.2 (H)

Table 9. **End-to-end fine-tuning setting.**

config	value
optimizer	LARS [66]
base learning rate	0.1
weight decay	0
optimizer momentum	0.9
batch size	16384
learning rate schedule	cosine decay
warmup epochs	10
training epochs	90
augmentation	RandomResizedCrop

表10. 线性探测设置。我们使用带大批次的LARS以加速训练；SGD在4096批次下表现类似。

部分微调。我们的MAE部分微调（§4.3）遵循表9中的设置，不同之处在于我们调整了微调周期数。我们观察到调整较少的块需要更长的训练计划。我们将微调周期数设置为{50、100、200}，并为每个调整块数选择最优值。

A.2. 从头开始监督训练 ViT-L/H

我们发现，在ImageNet-1K上训练*supervised* ViT-L/H *from scratch*并非易事。训练过程不稳定。虽然针对较小模型已有基于公开实现[53]的强基线方案，但针对更大规模ViT-L/H的训练方案仍属空白领域。直接沿用先前方案训练这些大型模型无法奏效，训练过程中频繁出现NaN损失值。

我们在表11中提供了我们的配方。我们遵循原始ViT [16]的方法，使用0.3的*wd*、4096的大批量大小和较长的预热期。根据[6]的建议，我们采用β_2=0.95的设置。按照[64]的方案，我们使用表11中列出的正则化方法并禁用其他方法。所有这些选择都是为了提高训练稳定性。我们的配方能够完成训练且不会出现NaN损失。

config	value
optimizer	AdamW
base learning rate	1e-4
weight decay	0.3
optimizer momentum	β_1, β_2=0.9, 0.95
batch size	4096
learning rate schedule	cosine decay
warmup epochs	20
training epochs	300 (B), 200 (L/H)
augmentation	`RandAug` (9, 0.5) [12]
label smoothing [52]	0.1
mixup [69]	0.8
cutmix [68]	1.0
drop path [30]	0.1 (B), 0.2 (L/H)
exp. moving average (EMA)	0.9999

Table 11. **Supervised training ViT from scratch.**

The accuracy is 82.6% for ViT-L (81.5% w/o EMA), and 83.1% for ViT-H (80.9% w/o EMA). Both ViT-L and ViT-H show an overfitting trend if not using EMA.

As a by-product, our recipe for ViT-B has 82.3% accuracy (82.1% w/o EMA), *vs*. 81.8% in [53].

A.3. Object Detection and Segmentation in COCO

We adapt the vanilla ViT for the use of an FPN backbone [36] in Mask R-CNN [24]. ViT has a stack of Transformer blocks that all produce feature maps at a single scale (*e.g*., stride 16). We equally divide this stack into 4 subsets and apply convolutions to upsample or downsample the intermediate feature maps for producing different scales (stride 4, 8, 16, or 32, the same as a standard ResNet [25]). FPN is built on these multi-scale maps.

For fair comparisons among different methods, we search for hyper-parameters for each entry in Table 4 (including all competitors). The hyper-parameters we search for are the learning rate, weight decay, drop path rate, and fine-tuning epochs. We will release code along with the specific configurations. For full model and training details, plus additional experiments, see [35].

A.4. Semantic Segmentation in ADE20K

We use UperNet [63] following the semantic segmentation code of [2]. We fine-tune end-to-end for 100 epochs with a batch size of 16. We search for the optimal *lr* for each entry in Table 5 (including all competitors).

The semantic segmentation code of [2] uses relative position bias [49]. Our MAE pre-training does *not* use it. For fair comparison, we turn on relative position bias *only* during transfer learning, initialized as zero. We note that our BEiT reproduction uses relative position bias in *both* pre-training and fine-tuning, following their code.

A.5. Additional Classification Tasks

We follow the setting in Table 9 for iNaturalist and Places fine-tuning (Table 6). We adjust the *lr* and fine-tuning epochs for each individual dataset.

method	model	params	acc
iGPT [6]	iGPT-L	1362 M	69.0
iGPT [6]	iGPT-XL	6801 M	72.0
BEiT [2]	ViT-L	304 M	52.1†
MAE	ViT-B	86 M	68.0
MAE	ViT-L	304 M	75.8
MAE	ViT-H	632 M	76.6

Table 12. **Linear probing results of masked encoding methods**. Our fine-tuning results are in Table 3. †: our implementation.

dataset	ViT-B	ViT-L	ViT-H	ViT-H$_{448}$	prev best
IN-Corruption ↓ [27]	51.7	41.8	**33.8**	36.8	42.5 [32]
IN-Adversarial [28]	35.9	57.1	68.2	**76.7**	35.8 [41]
IN-Rendition [26]	48.3	59.9	64.4	**66.5**	48.7 [41]
IN-Sketch [60]	34.5	45.3	49.6	**50.9**	36.0 [41]
our supervised training baselines:					
IN-Corruption ↓	45.8	42.3	**41.3**		
IN-Adversarial	27.2	29.6	**33.1**		
IN-Rendition	49.4	**50.9**	50.3		
IN-Sketch	35.6	37.5	**38.0**		

Table 13. **Robustness evaluation on ImageNet variants** (top-1 accuracy, except for IN-C [27] which evaluates mean corruption error). We test the same MAE models (Table 3) on different ImageNet validation sets, *without* any specialized fine-tuning. We provide system-level comparisons with the previous best results.

B. Comparison on Linear Probing Results

In §4.3 we have shown that linear probing accuracy and fine-tuning accuracy are largely *uncorrelated* and they have different focuses about linear separability. We notice that existing masked image encoding methods are generally less competitive in linear probing (*e.g*., than contrastive learning). For completeness, in Table 12 we compare on linear probing accuracy with masking-based methods.

Our MAE with ViT-L has 75.8% linear probing accuracy. This is substantially better than previous masking-based methods. On the other hand, it still lags behind contrastive methods under this protocol: *e.g*., MoCo v3 [9] has 77.6% linear probing accuracy for the ViT-L (Figure 9).

C. Robustness Evaluation on ImageNet

In Table 13 we evaluate the robustness of our models on different variants of ImageNet validation sets. We use the same models fine-tuned on *original* ImageNet (Table 3) and only run inference on the different validation sets, *without* any specialized fine-tuning. Table 13 shows that our method has strong scaling behavior: increasing the model sizes has significant gains. Increasing the image size helps in all sets but IN-C. Our results outperform the previous best results (of specialized systems) by large margins.

In contrast, *supervised* training performs much worse (Table 13 bottom; models described in A.2). For example, with ViT-H, our MAE pre-training is 35% better on IN-A (68.2% vs 33.1%) than the supervised counterpart.

config	value
optimizer	AdamW
base learning rate	1e-4
weight decay	0.3
optimizer momentum	β_1, β_2=0.9, 0.95
batch size	4096
learning rate schedule	cosine decay
warmup epochs	20
training epochs	300 (B), 200 (L/H)
augmentation	`RandAug` (9, 0.5) [12]
label smoothing [52]	0.1
mixup [69]	0.8
cutmix [68]	1.0
drop path [30]	0.1 (B), 0.2 (L/H)
exp. moving average (EMA)	0.9999

表11. 从零开始的监督训练ViT。

ViT-L的准确率为82.6%（不使用EMA时为81.5%），ViT-H的准确率为83.1%（不使用EMA时为80.9%）。若不使用EMA，ViT-L和ViT-H均呈现过拟合趋势。

作为副产品，我们的ViT-B配方达到了82.3%的准确率（未使用EMA时为82.1%），*vs*。[53]中为81.8%。

A.3. COCO中的目标检测与分割

我们将普通的ViT进行调整，使其能够在Mask R-CNN[24]中作为FPN主干网络[36]使用。ViT包含一系列Transformer模块，这些模块均生成单一尺度（*e.g*，步幅16）的特征图。我们将这个堆栈均分为4个子集，并应用卷积对中间特征图进行上采样或下采样，以生成不同尺度（步幅4、8、16或32，与标准ResNet[25]相同）。FPN正是在这些多尺度特征图上构建而成。

为了公平比较不同方法，我们在表4中为每个条目（包括所有竞争对手）搜索超参数。我们搜索的超参数包括学习率、权重衰减、丢弃路径率和微调周期。我们将发布代码及具体配置。完整模型和训练细节以及额外实验请参见[35]。

A.4. ADE20K中的语义分割

我们采用UperNet[63]，遵循[2]的语义分割代码进行端到端微调100个周期，批次大小为16。我们为表5中的每个条目（包括所有竞争对手）搜索最优的{v*}。

[2]的语义分割代码采用了相对位置偏置[49]。我们的MAE预训练*not*使用了该技术。为公平比较，我们在迁移学习期间启用相对位置偏置*only*，并将其初始化为零。需要说明的是，根据原代码实现，我们的BEiT复现版本在*both*预训练和微调阶段均采用了相对位置偏置。

A.5. 附加分类任务

我们遵循表9中针对iNaturalist和Places微调的设置（表6）。我们针对每个单独的数据集调整{v*}和微调周期。

method	model	params	acc
iGPT [6]	iGPT-L	1362 M	69.0
iGPT [6]	iGPT-XL	6801 M	72.0
BEiT [2]	ViT-L	304 M	52.1†
MAE	ViT-B	86 M	68.0
MAE	ViT-L	304 M	75.8
MAE	ViT-H	632 M	76.6

表12. 掩码编码方法的线性探测结果。我们的微调结果见表3。{v*}：我们的实现。

dataset	ViT-B	ViT-L	ViT-H	ViT-H_{448}	prev best
IN-Corruption ↓ [27]	51.7	41.8	**33.8**	36.8	42.5 [32]
IN-Adversarial [28]	35.9	57.1	68.2	**76.7**	35.8 [41]
IN-Rendition [26]	48.3	59.9	64.4	**66.5**	48.7 [41]
IN-Sketch [60]	34.5	45.3	49.6	**50.9**	36.0 [41]
our supervised training baselines:					
IN-Corruption ↓	45.8	42.3	**41.3**		
IN-Adversarial	27.2	29.6	**33.1**		
IN-Rendition	49.4	**50.9**	50.3		
IN-Sketch	35.6	37.5	**38.0**		

表13. 在ImageNet变体上的鲁棒性评估（Top-1准确率，除IN-C[27]评估平均损坏误差外）。我们在不同ImageNet验证集上测试相同MAE模型（表3），{v*}无需专门微调。我们提供了与先前最佳结果的系统级比较。

B. 线性探测结果对比

在§4.3中我们已经证明线性探测精度和微调精度很大程度上*uncorrelated*，并且它们关于线性可分离性有不同的侧重点。我们注意到现有的掩码图像编码方法在线性探测方面通常竞争力较弱（*e.g*，相较于对比学习）。为完整起见，我们在表12中与基于掩码的方法进行了线性探测精度比较。

我们的ViT-L MAE模型在线性探测准确率上达到了75.8%，这显著优于以往的基于掩码的方法。但在此协议下仍落后于对比学习方法：*e.g*，MoCo v3 [9] 的ViT-L模型在线性探测准确率为77.6%（图9）。

C. ImageNet上的鲁棒性评估

在表13中，我们在ImageNet验证集的不同变体上评估了模型的鲁棒性。我们使用在*original* ImageNet上微调的相同模型（表3），仅在不同验证集上进行推理，*without*无需任何专门微调。表13显示我们的方法具有强大的扩展性能：增大模型规模能带来显著收益。增大图像尺寸对所有数据集都有帮助，除了IN-C。我们的结果以较大优势超越了先前（专业系统的）最佳结果。

相比之下，*supervised*训练的表现要差得多（表13底部；模型描述见A.2节）。例如使用ViT-H时，我们的MAE预训练在IN-A上的表现比监督式对应方法高出35%（68.2%对比33.1%）。

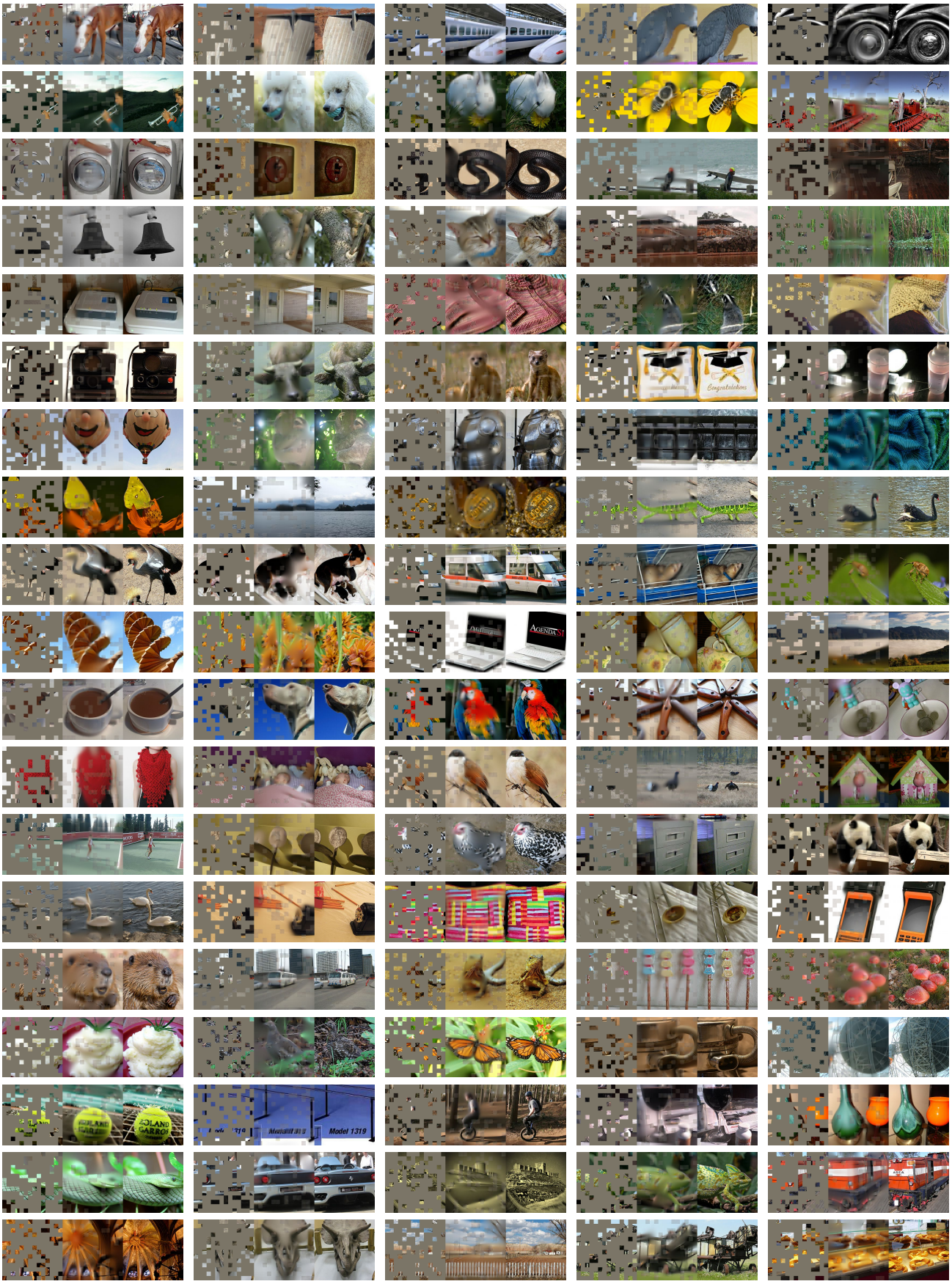

Figure 10. **Uncurated random samples** on ImageNet *validation* images. For each triplet, we show the masked image (left), our MAE reconstruction (middle), and the ground-truth (right). The masking ratio is 75%.

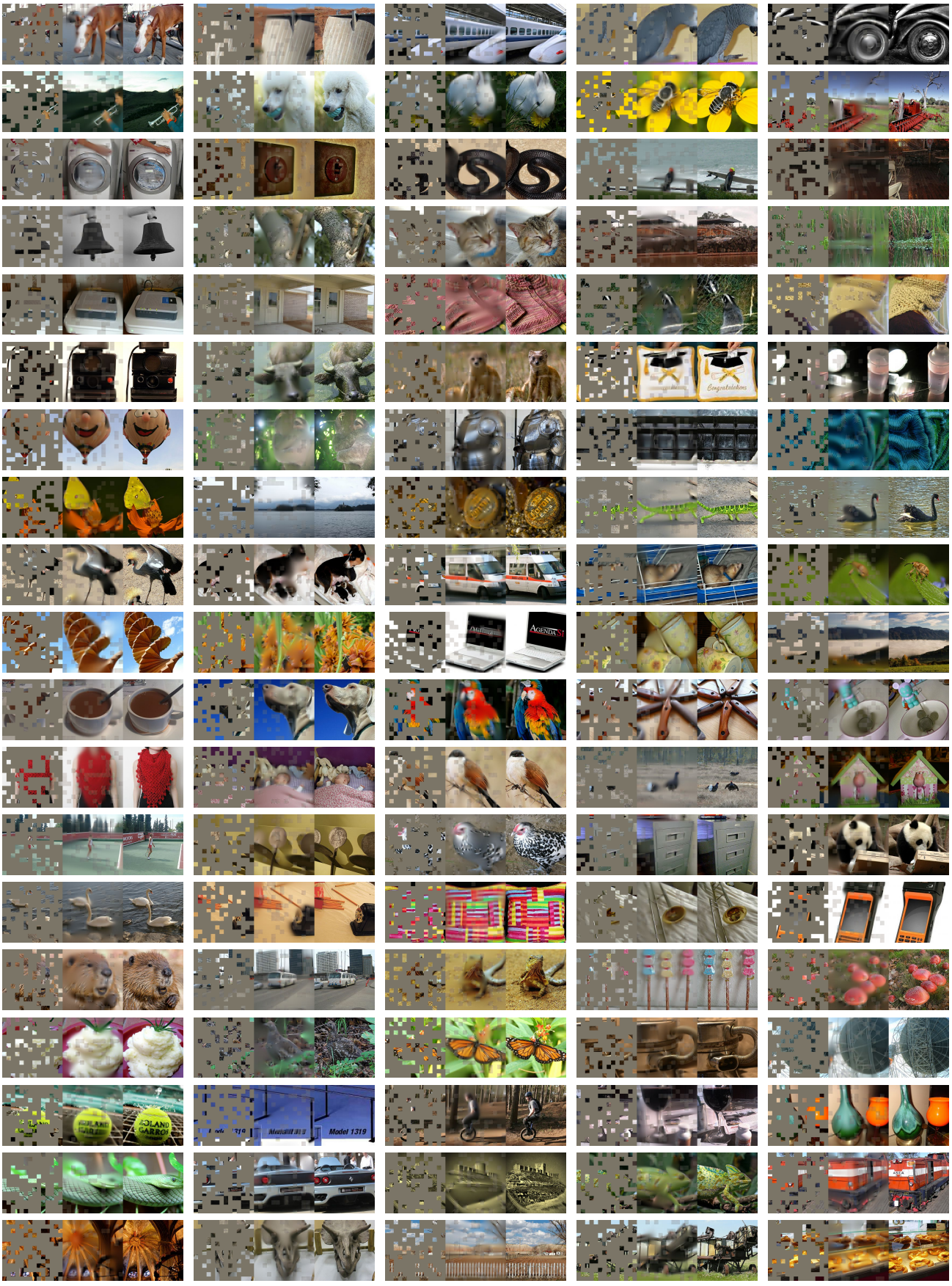

图 10. ImageNet *validation* 图像上未经筛选的随机样本。每个三联图中，我们展示了掩码图像（左）、MAE 重建结果（中）和真实图像（右）。掩码比例为 75%。

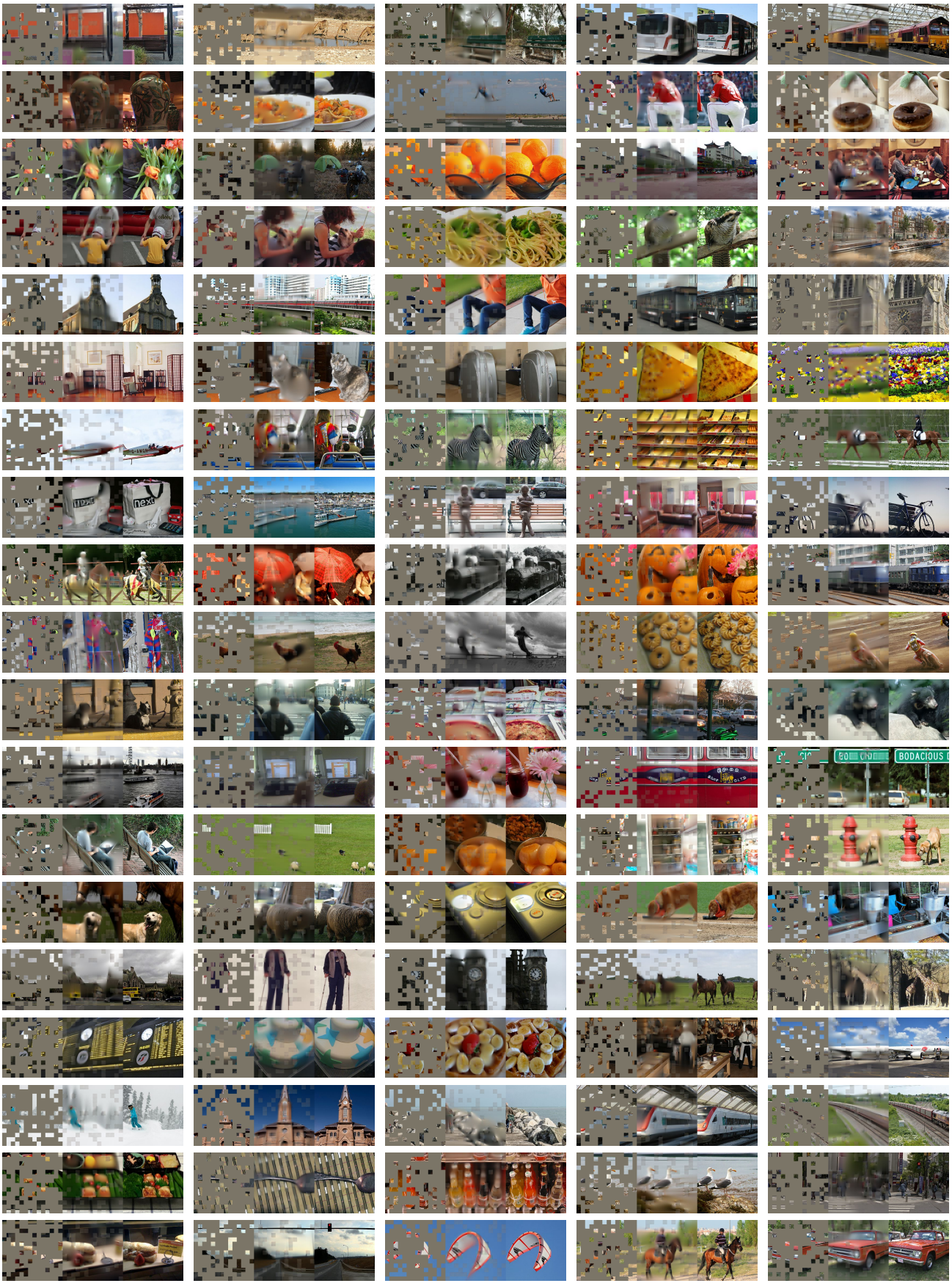

Figure 11. **Uncurated random samples** on COCO validation images, using an MAE trained on ImageNet. For each triplet, we show the masked image (left), our MAE reconstruction (middle), and the ground-truth (right). The masking ratio is 75%.

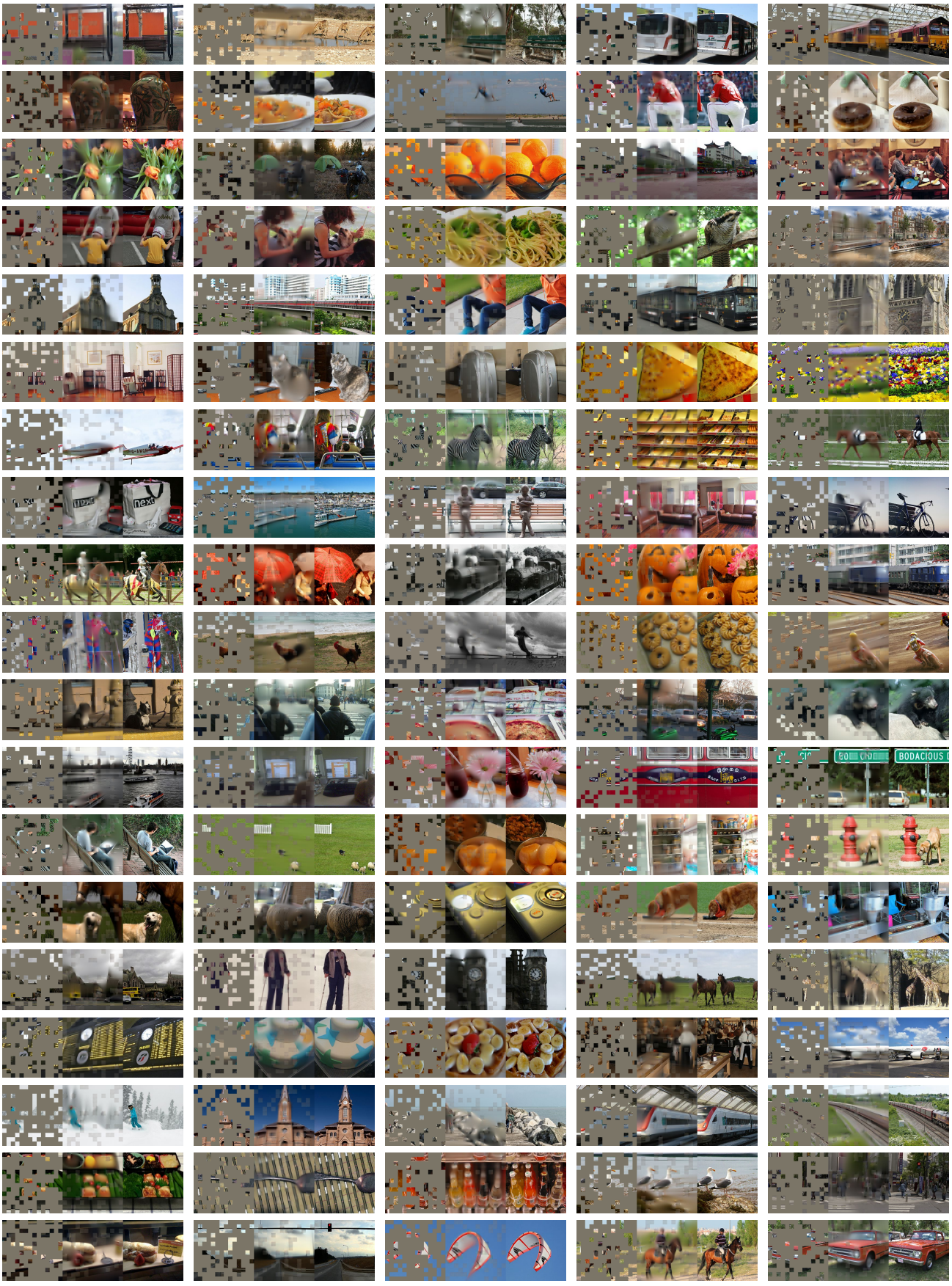

图11. 在COCO验证图像上未经筛选的随机样本，使用在ImageNet上训练的MAE模型。每个三联图中，我们展示了掩码图像（左）、MAE重建结果（中）和真实图像（右）。掩码比例为75%。