Masked Autoencoders Are Scalable Vision Learners

掩码自编码器是可扩展的视觉学习者

Kaiming He*,† Xinlei Chen* Saining Xie Yanghao Li Piotr Dollár Ross Girshick
Kaiming He*,† Xinlei Chen* Saining Xie Yanghao Li Piotr Dollár Ross Girshick
*equal technical contribution † project lead
* 具有相等的技术贡献 † 项目负责人
Facebook AI Research (FAIR)
Facebook AI Research (FAIR)

Abstract

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 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 pretraining and shows promising scaling behavior.

本文展示了掩码自编码器 (MAE) 是计算机视觉领域可扩展的自监督学习者。我们的 MAE 方法很简单: 我们对输入图像的随机区域进行掩码，并重建缺失的像素。该方法基于两个核心设计。首先，我们开发了一种非对称的编码器-解码器架构，编码器仅在可见的补丁子集上操作 (不使用掩码标记)，同时配备一个轻量级解码器，从潜在表示和掩码标记中重建原始图像。其次，我们发现对输入图像进行高比例的掩码，例如 75%，会产生一个非平凡且有意义的自监督任务。将这两个设计结合起来，使我们能够高效且有效地训练大型模型: 我们加速了训练 (提高了 3× 或更多) 并提高了准确性。我们可扩展的方法允许学习高容量模型，这些模型具有良好的泛化能力: 例如，一个普通的 ViT-Huge 模型在仅使用 ImageNet-1K 数据的方法中达到了最佳准确率 (87.8%)。在下游任务中的迁移性能优于监督预训练，并显示出良好的扩展行为。

1. Introduction

1. 引言

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].

深度学习见证了架构能力和容量的持续增长的爆炸性发展 [33, 25, 57] 。得益于硬件的快速进步，今天的模型可以轻松地对一百万张图像进行过拟合 [13]，并开始需要数亿张-通常是公众无法访问的-标记图像 [16]。

This appetite for data has been successfully addressed in natural language processing (NLP) by self-supervised pretraining. 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].

这种对数据的渴求在自然语言处理 (NLP) 中通过自监督预训练得到了成功解决。基于 GPT [47, 48, 4] 的自回归语言建模和 BERT [14] 的掩码自编码的解决方案在概念上是简单的: 它们去除一部分数据并学习预测被去除的内容。这些方法现在使得训练包含超过一百亿参数的可泛化 NLP 模型成为可能 [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:

掩码自编码器的概念是一种更一般的去噪自编码器 [58]，在计算机视觉中同样自然且适用。事实上，与视觉相关的研究 [59, 46] 在 BERT 之前就已经存在。然而，尽管在 BERT 成功后对这一思想产生了显著的兴趣，但视觉中的自编码方法进展落后于 NLP。我们提出: 掩码自编码在视觉和语言之间有什么不同？我们试图从以下几个角度回答这个问题:

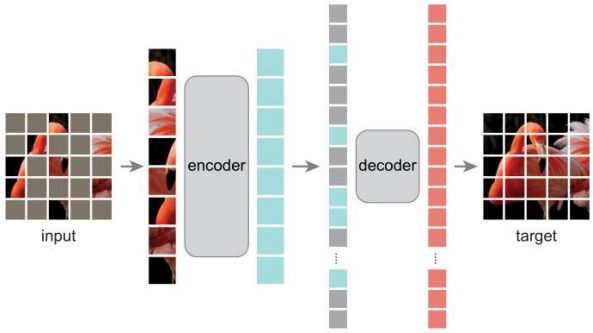

Figure 1. Our MAE architecture. During pre-training, a large random subset of image patches (e.g., 75%) is masked out. The encoder is applied to the small subset of visible patches. Mask tokens are introduced after the encoder, and the full set of encoded patches and mask tokens is processed by a small 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.

图 1. 我们的 MAE 架构。在预训练期间，大量随机选择的图像块 (例如，75%) 被掩盖。编码器应用于可见块的小子集。在编码器之后引入掩码标记，编码的块和掩码标记的完整集合由一个小解码器处理，该解码器重建原始图像的像素。在预训练之后，解码器被丢弃，编码器应用于未损坏的图像 (完整的块集合) 以进行识别任务。

(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.

(i) 直到最近，架构是不同的。在视觉领域，卷积网络 [34] 在过去十年中占主导地位 [33]。卷积通常在规则网格上操作，将 "指示符" 如掩码标记 [14] 或位置嵌入 [57] 集成到卷积网络中并不简单。然而，这一架构差距已通过引入视觉变换器 (ViT) [16] 得到解决，不再构成障碍。

(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 understanding 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.

(ii) 信息密度在语言和视觉之间是不同的。语言是人类生成的信号，具有高度的语义性和信息密度。当训练一个模型仅预测每个句子中少数缺失的单词时，这一任务似乎会引发复杂的语言理解。相反，图像是具有大量空间冗余的自然信号，例如，缺失的区域可以从相邻区域恢复，而几乎不需要对部分、物体和场景进行高级理解。为了克服这种差异并鼓励学习有用的特征，我们展示了一种在计算机视觉中效果良好的简单策略: 对随机区域进行大比例的遮罩。这一策略大大减少了冗余，并创建了一个具有挑战性的自监督任务，要求超越低级图像统计进行整体理解。要获取我们重建任务的定性感受，请参见图 2 - 4。

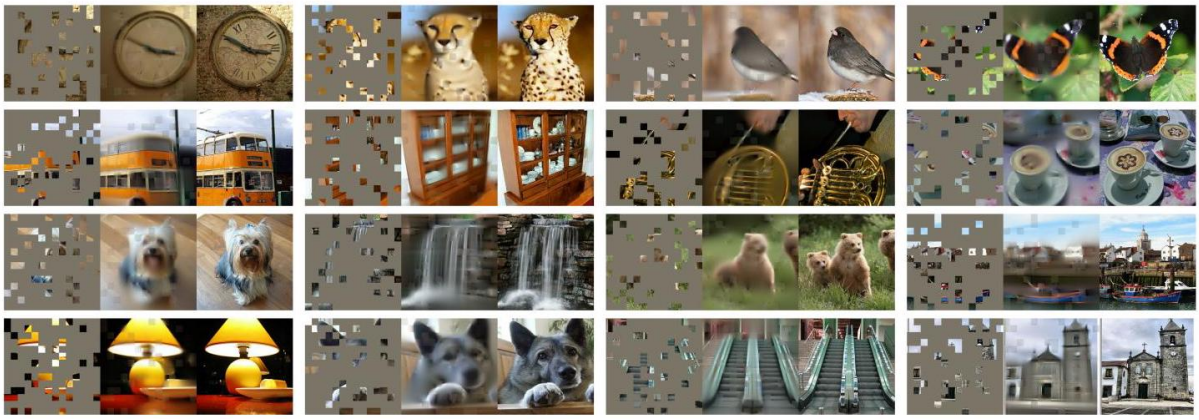

Figure 2. Example results on ImageNet validation images. For each triplet, we show the masked image (left), our MAE reconstruction † (middle), and the ground-truth (right). The masking ratio is 80% , leaving only 39 out of 196 patches. More examples are in the appendix. † As no loss is computed on visible 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.

图 2. 在 ImageNet 验证图像上的示例结果。对于每个三元组，我们展示了被遮罩的图像 (左)，我们的 MAE 重建 † (中)，以及真实值 (右)。遮罩比例为 80% ，仅保留 196 个补丁中的 39 个。更多示例见附录。† 由于在可见补丁上没有计算损失，因此模型在可见补丁上的输出在质量上较差。可以简单地将输出与可见补丁叠加以改善视觉质量。我们故意选择不这样做，以便更全面地展示该方法的行为。

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.

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)。观察右侧两个示例的重建，尽管与真实值不同，但在语义上是合理的。

(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.

(iii) 自编码器的解码器将潜在表示映射回输入，在重建文本和图像之间扮演不同的角色。在视觉中，解码器重建像素，因此其输出的语义水平低于常见的识别任务。这与语言形成对比，在语言中，解码器预测缺失的单词，这些单词包含丰富的语义信息。虽然在 BERT 中解码器可以是简单的 (一个 MLP)[14]，但我们发现对于图像，解码器的设计在确定学习到的潜在表示的语义水平方面起着关键作用。

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.

基于这一分析，我们提出了一种简单、有效且可扩展的掩码自编码器 (MAE) 用于视觉表示学习。我们的 MAE 从输入图像中掩盖随机补丁，并在像素空间中重建缺失的补丁。它具有不对称的编码器-解码器设计。我们的编码器仅在可见的补丁子集上操作 (没有掩码标记)，而我们的解码器轻量且从潜在表示中重建输入，同时包含掩码标记 (图 1)。将掩码标记转移到我们不对称编码器-解码器中的小解码器，导致计算量大幅减少。在这种设计下，极高的掩码比例 (例如，75%) 可以实现双赢的局面: 它优化了准确性，同时允许编码器仅处理一小部分 (例如，25%) 补丁。这可以将整体预训练时间减少 3× 或更多，并同样减少内存消耗，使我们能够轻松地将 MAE 扩展到大型模型。

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.

我们的 MAE 学习了非常高容量的模型，具有良好的泛化能力。通过 MAE 预训练，我们可以在 ImageNet-1K 上训练像 ViT-Large/-Huge [16] 这样的数据需求量大的模型，并提高其泛化性能。使用普通的 ViT-Huge 模型，我们在 ImageNet-1K 上微调时达到了 87.8% 的准确率。这超越了所有仅使用 ImageNet-1K 数据的先前结果。我们还评估了在目标检测、实例分割和语义分割上的迁移学习。在这些任务中，我们的预训练取得了比其监督预训练对应物更好的结果，更重要的是，我们观察到通过扩大模型规模获得了显著的提升。这些观察结果与在自然语言处理中的自监督预训练所见的结果一致 [14, 47, 48, 4] ，我们希望这能使我们的领域探索类似的轨迹。

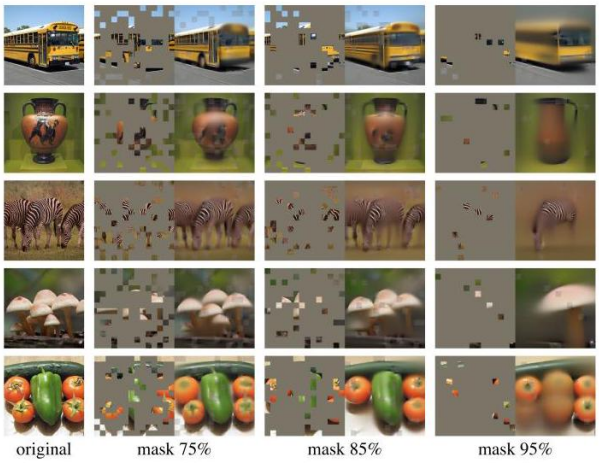

Figure 4. Reconstructions of ImageNet validation images using an MAE pre-trained with a masking ratio of 75% but applied on inputs with higher masking ratios. The predictions differ plausibly from the original images, showing that the method can generalize.

图 4. 使用以 75% 的掩码比例预训练的 MAE 重建的 ImageNet 验证图像，但应用于具有更高掩码比例的输入。预测结果与原始图像在合理范围内有所不同，显示出该方法具有泛化能力。

2. Related Work

2. 相关工作

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.

掩码语言建模及其自回归对应物，例如 BERT [14] 和 GPT [47, 48, 4]，是自然语言处理领域中非常成功的预训练方法。这些方法保留输入序列的一部分，并训练模型预测缺失的内容。这些方法已被证明具有良好的扩展性 [4]，大量证据表明这些预训练表示能够很好地泛化到各种下游任务。

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.

自编码是一种经典的学习表示的方法。它具有一个编码器，将输入映射到潜在表示，以及一个解码器，用于重构输入。例如，主成分分析 (PCA) 和 k 均值聚类都是自编码器 [29]。去噪自编码器 (DAE) [58] 是一类自编码器，它们会破坏输入信号并学习重构原始的、未损坏的信号。一系列方法可以被视为在不同破坏下的广义 DAE，例如，掩蔽像素 [59,46, 6] 或去除颜色通道 [70]。我们的 MAE 是一种去噪自编码形式，但在许多方面与经典的 DAE 不同。

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] .

掩蔽图像编码方法从被掩蔽破坏的图像中学习表示。[59] 的开创性工作将掩蔽视为 DAE 中的一种噪声类型。上下文编码器 [46] 使用卷积网络对大面积缺失区域进行填充。受到自然语言处理 (NLP) 成功的启发，相关的近期方法 [6, 16, 2] 基于变换器 [57]。iGPT [6] 在像素序列上操作并预测未知像素。ViT 论文 [16] 研究了自监督学习中的掩蔽补丁预测。最近，BEiT [2] 提出了预测离散标记 [44, 50] 的方法。

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.

自监督学习方法在计算机视觉中引起了显著的关注，通常集中于不同的预训练前置任务 [15, 61, 42, 70, 45, 17]。最近，对比学习 [3, 22] 变得流行，例如 [62, 43, 23, 7] ，它建模了两个或多个视图之间的图像相似性和不相似性 (或仅相似性 [21, 8])。对比及相关方法在很大程度上依赖于数据增强 [7, 21, 8] 。自编码追求一个概念上不同的方向，并表现出不同的行为，正如我们将要展示的那样。

3. Approach

3. 方法

Our masked autoencoder (MAE) is a simple autoencod-ing 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.

我们的掩码自编码器 (MAE) 是一种简单的自编码方法，它在给定部分观察的情况下重建原始信号。与所有自编码器一样，我们的方法具有一个编码器，该编码器将观察到的信号映射到潜在表示，并且一个解码器，该解码器从潜在表示重建原始信号。与经典自编码器不同，我们采用了不对称设计，使编码器仅在部分观察信号 (没有掩码标记) 上操作，而轻量级解码器则从潜在表示和掩码标记中重建完整信号。图 1 展示了接下来介绍的这一思想。

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".

掩码。遵循 ViT [16]，我们将图像划分为规则的非重叠块。然后我们抽取一部分块并掩盖 (即，移除)

其余的块。我们的抽样策略非常简单: 我们在均匀分布下随机抽取块，不进行替换。我们将其称为 “随机抽样”。

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.

高掩码比例的随机抽样 (即，移除块的比例) 在很大程度上消除了冗余，从而创建了一个无法通过从可见邻近块的外推轻易解决的任务 (见图 2 - 4)。均匀分布防止了潜在的中心偏差 (即，更多被掩盖的块靠近图像中心)。最后，高度稀疏的输入为设计高效的编码器创造了机会，接下来将介绍。

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 编码器。我们的编码器是一个 ViT [16]，但仅应用于可见的、未掩盖的块。就像标准 ViT 一样，我们的编码器通过线性投影嵌入块，并添加位置嵌入，然后通过一系列 Transformer 块处理生成的集合。然而，我们的编码器仅在完整集合的一个小子集 (例如，25%) 上操作。被掩盖的块被移除；不使用掩码标记。这使我们能够仅用一小部分计算和内存训练非常大的编码器。完整集合由接下来描述的轻量级解码器处理。

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 missing 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.

MAE 解码器。MAE 解码器的输入是由 (i) 编码的可见补丁和 (ii) 掩码标记组成的完整标记集。见图 1。每个掩码标记 [14] 是一个共享的、学习到的向量，表示要预测的缺失补丁的存在。我们为这个完整标记集中的所有标记添加位置嵌入；如果没有这些，掩码标记将对其在图像中的位置没有任何信息。解码器还有一系列 Transformer 块。

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.

MAE 解码器仅在预训练期间用于执行图像重建任务 (仅使用编码器生成用于识别的图像表示)。因此，解码器架构可以灵活设计，与编码器设计无关。我们实验了非常小的解码器，其宽度和深度均小于编码器。例如，我们的默认解码器每个标记的计算量为 $< 10\%$ ，与编码器相比。通过这种不对称设计，完整的标记集仅由轻量级解码器处理，这显著减少了预训练时间。

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]

重建目标。我们的 MAE 通过预测每个掩码补丁的像素值来重建输入。解码器输出中的每个元素是一个表示补丁的像素值向量。解码器的最后一层是一个线性投影，其输出通道数等于补丁中的像素值数量。解码器的输出被重塑以形成重建的图像。我们的损失函数计算重建图像与原始图像在像素空间中的均方误差 (MSE)。我们仅在掩码补丁上计算损失，类似于 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.

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，并用它们来归一化该补丁。在我们的实验中，使用归一化像素作为重建目标提高了表示质量。

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 unshuf-fling operations are fast.

简单实现。我们的 MAE 预训练可以高效地实施，重要的是，不需要任何专门的稀疏操作。首先，我们为每个输入补丁生成一个标记 (通过线性投影并添加位置嵌入)。接下来，我们随机打乱标记列表，并根据掩码比例移除列表的最后一部分。这个过程产生了一个小的标记子集供编码器使用，相当于不放回地抽样补丁。编码后，我们将一组掩码标记附加到编码补丁的列表中，并对这个完整列表进行反打乱 (逆转随机打乱操作)，以将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。这个简单的实现引入了微不足道的开销，因为打乱和反打乱操作都很快。

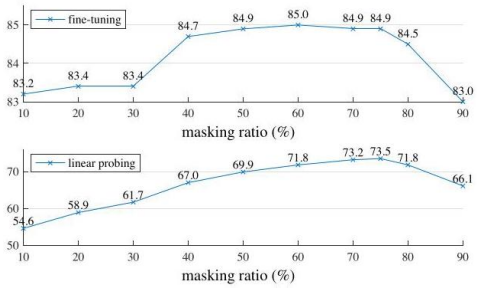

Figure 5. Masking ratio. A high masking ratio (75%) works well for both fine-tuning (top) and linear probing (bottom). The y-axes are ImageNet-1K validation accuracy (%) in all plots in this paper.

图 5. 掩码比例。较高的掩码比例 (75%) 在微调 (上) 和线性探测 (下) 中都表现良好。所有图中的 y 轴为 ImageNet-1K 验证准确率 (%)。

4. ImageNet Experiments

4. ImageNet 实验

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.

我们在 ImageNet-1K (IN1K) [13] 训练集上进行自监督预训练。然后我们进行监督训练，以评估表示，方法为 (i) 端到端微调或 (ii) 线性探测。我们报告单个 224 × 224 裁剪的 top-1 验证准确率。详细信息见附录 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:

基线:ViT-Large。我们在消融研究中使用 ViT-Large (ViT-L/16) [16] 作为主干。ViT-L 非常大 (比 ResNet-50 [25] 大一个数量级)，并倾向于过拟合。以下是从头训练的 ViT-L 与从我们的基线 MAE 微调的 ViT-L 之间的比较:

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 pretraining 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.

我们注意到，从头开始训练监督式 ViT-L 并非易事，需要一个强正则化的良好方案 (82.5%，见附录 A.2)。即便如此，我们的 MAE 预训练仍然带来了显著的改善。在这里，微调仅进行 50 个周期 (而从头开始则为 200 个周期)，这意味着微调的准确性在很大程度上依赖于预训练。

4.1. Main Properties

4.1. 主要特性

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

我们使用表 1 中的默认设置对 MAE 进行了消融实验 (见说明)。观察到几个有趣的特性。

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%).

掩码比例。图 5 显示了掩码比例的影响。最佳比例出乎意料地高。比例 75% 对于线性探测和微调都表现良好。这种行为与 BERT [14] 相对立，后者的典型掩码比例为 15% 。我们的掩码比例也远高于计算机视觉相关工作中的比例 [6, 16, 2] (20% 到 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.

模型推断缺失的补丁以生成不同但合理的输出 (图 4)。它理解物体和场景的整体形态，这不能仅通过延伸线条或纹理来简单完成。我们假设这种类似推理的行为与有用表示的学习有关。

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

图 5 还显示线性探测和微调结果遵循不同的趋势。对于线性探测，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

块	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

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

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

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

[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., $\sim 0.5\%$).

[1] 仅在掩码补丁上计算损失与传统的去噪自编码器 [58] 不同，后者在所有像素上计算损失。这个选择纯粹是结果驱动的: 在所有像素上计算损失会导致准确性略有下降 (例如，$\sim 0.5\%$)。

case	ft	lin	FLOPs
带有 [M] 的编码器	84.2	59.6	3.3 ×
不带 [M] 的编码器	84.9	73.5	1×

(a) Decoder depth. A deep decoder can improve linear probing accuracy. (d) Reconstruction target. Pixels as reconstruction targets are effective.

(a) 解码器深度。深层解码器可以提高线性探测的准确性。(d) 重建目标。作为重建目标的像素是有效的。

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

情况	ft	lin
像素 (不带归一化)	84.9	73.5
像素 (带归一化)	85.4	73.9
主成分分析 (PCA)	84.6	72.3
dVAE 令牌	85.3	71.6

(b) Decoder width. The decoder can be narrower than the encoder (1024-d). (e) Data augmentation. Our MAE works with minimal or no augmentation.

(b) 解码器宽度。解码器可以比编码器 (1024 维) 更窄。(e) 数据增强。我们的 MAE 在最小或没有增强的情况下工作。

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

情况	ft	lin
none	84.0	65.7
裁剪，固定大小	84.7	73.1
裁剪，随机大小	84.9	73.5
裁剪 + 颜色抖动	84.3	71.9

(c) Mask token. An encoder without mask tokens is more accurate and faster (Table 2). (f) Mask sampling. Random sampling works the best. See Figure 6 for visualizations.

(c) 掩码标记。没有掩码标记的编码器更准确且更快 (表 2)。(f) 掩码采样。随机采样效果最佳。请参见图 6 以获取可视化。

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

案例	比例	ft	lin
随机	75	84.9	73.5
块	50	83.9	72.3
块	75	82.8	63.9
网格	75	84.0	66.0

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

$\sim 20\%$(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%) .

表 1. 在 ImageNet-1K 上使用 ViT-L/16 的 MAE 消融实验。我们报告微调 (ft) 和线性探测 (lin) 准确率 (%)。如果未指定，默认设置为: 解码器深度为 8，宽度为 512，重建目标为未归一化像素，数据增强为随机缩放裁剪，掩码比例为 75% ，预训练长度为 800 个周期。默认设置以灰色标记。准确率随着掩码比例的增加而稳步提高，直到达到最佳点: 准确率差距高达 $\sim 20\%$(54.6% 与 73.5%)。对于微调，结果对比例的敏感性较低，广泛的掩码比例 ($40-80\%$) 效果良好。图 5 中的所有微调结果均优于从头开始训练 (82.5%) 。

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

解码器设计。我们的 MAE 解码器可以灵活设计，如表 1a 和 1b 所研究的。

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').

表 1a 变化了解码器的深度 (Transformer 块的数量)。足够深的解码器对于线性探测是重要的。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注于重建，但对于识别的相关性较低。一个合理深度的解码器可以考虑重建的专业化，使潜在表示处于更抽象的层次。这种设计可以在线性探测中带来高达 8% 的提升 (表 1a，'lin')。然而，如果使用微调，编码器的最后几层可以调整以适应识别任务。解码器的深度对提高微调的影响较小 (表 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.

有趣的是，我们的 MAE 使用单块解码器在微调时表现强劲 (84.8%)。请注意，单个 Transformer 块是将信息从可见标记传播到掩码标记的最小要求。如此小的解码器还可以进一步加快训练速度。

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.

在表 1b 中，我们研究了解码器的宽度 (通道数)。我们默认使用 512-d，这在微调和线性探测下表现良好。更窄的解码器在微调时也表现良好。

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.

总体而言，我们的默认 MAE 解码器是轻量级的。它有 8 个块，宽度为 512-d(表 1 中的灰色)。每个标记的 FLOPs 仅为 ViT-L(24 块，1024-d) 的 9%。因此，尽管解码器处理所有标记，但它仍然是整体计算的一小部分。

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 ×

编码器	深度衰减	准确率	小时	加速
ViT-L，带有 [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，带有 [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.

表 2. 我们的 MAE 训练的实际时间 (800 个周期)，在 128 个 TPU-v3 核心上使用 TensorFlow 进行基准测试。加速是相对于具有掩码标记的编码器的条目 (灰色)。解码器宽度为 512，掩码比例为 75%.† : 该条目是通过训练十个周期估算的。

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.

掩码令牌。我们 MAE 的一个重要设计是跳过编码器中的掩码令牌 [M] ，并在轻量级解码器中稍后应用它。表 1c 研究了这一设计。

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 pretraining, 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.

如果编码器使用掩码令牌，它的表现会更差: 在线性探测中，其准确率下降了 14% 。在这种情况下，预训练和部署之间存在差距: 该编码器在预训练时输入中有大量的掩码令牌，而在未损坏的图像中并不存在这种情况。这一差距可能会降低部署时的准确率。通过从编码器中移除掩码令牌，我们限制编码器始终看到真实的补丁，从而提高准确率。

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 $(3.5-4.1\times)$, for a smaller decoder (1-block), a larger encoder (ViT-H), or both. Note that the speedup can be $>4\times$ 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.

此外，通过在编码器中跳过掩码令牌，我们大大减少了训练计算。在表 1c 中，我们将整体训练 FLOPs 减少了 3.3× 。这导致我们实现中的 2.8× 实时加速 (见表 2)。对于较小的解码器 (1-block)、较大的编码器 (ViT-H) 或两者，实时加速甚至更大 $(3.5-4.1\times)$ 。请注意，对于掩码比例为 75% 的情况，加速可以达到 $>4\times$ ，部分原因是自注意力的复杂度是二次的。此外，内存大大减少，这可以使得训练更大的模型成为可能，或者通过大批量训练进一步加速。时间和内存效率使我们的 MAE 适合训练非常大的模型。

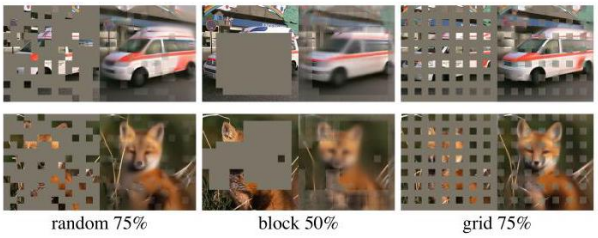

Figure 6. Mask sampling strategies determine the pretext task difficulty, influencing reconstruction quality and representations (Table 1f). Here each output is from an MAE trained with the specified masking strategy. Left: random sampling (our default). Middle: block-wise sampling [2] that removes large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

图 6。掩码采样策略决定了前置任务的难度，影响重建质量和表示 (表 1f)。这里每个输出来自于使用指定掩码策略训练的 MAE。左: 随机采样 (我们的默认设置)。中: 块状采样 [2]，去除大随机块。右: 网格状采样，保留每四个补丁中的一个。图像来自验证集。

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.

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，我们的结果基于没有 (每个补丁) 归一化的像素。使用带有归一化的像素可以提高准确性。这种每个补丁的归一化在局部增强了对比度。在另一种变体中，我们在补丁空间中执行主成分分析 (PCA)，并使用最大的 PCA 系数 (这里是 96) 作为目标。这样做会降低准确性。这两个实验表明，高频成分在我们的方法中是有用的。

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 tokeniza-tion is not necessary in transfer learning.

我们还比较了一种预测标记的 MAE 变体，这是在 BEiT [2] 中使用的目标。具体来说，对于这个变体，我们使用 DALLE 预训练的 dVAE [50] 作为标记器，遵循 [2]。在这里，MAE 解码器使用交叉熵损失预测标记索引。这种标记化在微调准确性上比未归一化像素提高了 0.4%，但与归一化像素相比没有优势。它还降低了线性探测的准确性。在 §5 中，我们进一步表明，在迁移学习中标记化并不是必要的。

Our pixel-based MAE is much simpler than tokeniza-tion. 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.

我们的基于像素的 MAE 比标记化简单得多。dVAE 标记器需要一个额外的预训练阶段，这可能依赖于额外的数据 (250M 图像 [50])。dVAE 编码器是一个大型卷积网络 (40% FLOPs 的 ViT-L)，并增加了非平凡的开销。使用像素不会遭受这些问题。

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

数据增强。表 1e 研究了数据增强对我们 MAE 预训练的影响。

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.

我们的 MAE 在仅使用裁剪增强时表现良好，无论是固定大小还是随机大小 (两者都有随机水平翻转)。添加颜色抖动会降低结果，因此我们在其他实验中不使用它。

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.

出人意料的是，即使不使用数据增强 (仅中心裁剪，不翻转)，我们的 MAE 仍然表现良好。这个特性与对比学习和相关方法 [62, 23, 7, 21] 的显著不同，这些方法严重依赖于数据增强。有观察 [21] 表明，仅使用裁剪增强会分别降低 BYOL [21] 和 SimCLR [7] 的准确性 13% 和 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，容易满足一个平凡的解决方案。

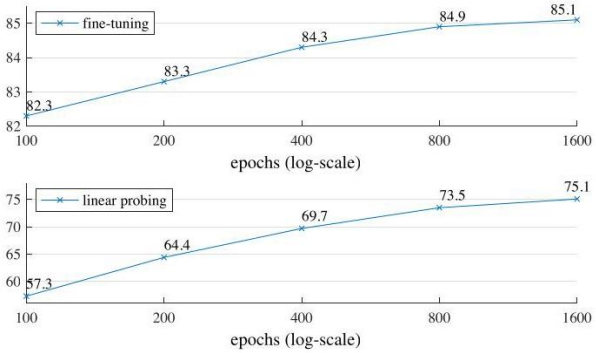

Figure 7. Training schedules. A longer training schedule gives a noticeable improvement. Here each point is a full training schedule. The model is ViT-L with the default setting in Table 1.

图 7. 训练计划。更长的训练计划显著提高了性能。这里每个点代表一个完整的训练计划。模型为 ViT-L，使用表 1 中的默认设置。

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.

在 MAE 中，数据增强的主要作用是通过随机掩蔽 (下文将进行剖析) 来实现的。每次迭代的掩蔽不同，因此它们生成新的训练样本，而不依赖于数据增强。由于掩蔽，前置任务变得更加困难，并且需要较少的增强来规范训练。

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

掩蔽采样策略。在表 1f 中，我们比较了不同的掩蔽采样策略，如图 6 所示。

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.

在 [2] 中提出的块状掩蔽策略倾向于去除大块 (图 6 中)。我们的 MAE 在掩蔽比例为 50% 时表现合理，但在比例为 75% 时性能下降。这个任务比随机采样更困难，因为观察到更高的训练损失。重建图像也更模糊。

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.

我们还研究了网格采样，它定期保留每四个补丁中的一个 (图 6 右)。这是一个更简单的任务，训练损失较低。重建图像更清晰。然而，表示质量较低。

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.

简单的随机采样对我们的 MAE 效果最佳。它允许更高的掩蔽比例，这提供了更大的加速收益，同时也保持了良好的准确性。

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.

训练计划。到目前为止，我们的消融实验基于 800 个周期的预训练。图 7 显示了训练计划长度的影响。随着训练时间的延长，准确率稳步提高。实际上，即使在 1600 个周期时，我们也没有观察到线性探测准确率的饱和。这种行为与对比学习方法不同，例如，MoCo v3 [9] 在 ViT-L 上在 300 个周期时达到饱和。请注意，MAE 编码器每个周期仅看到 25% 个补丁，而在对比学习中，编码器每个周期看到 200% (双裁剪) 甚至更多 (多裁剪) 补丁。

method	pre-train data	ViT-B	ViT-L	ViT-H	ViT-H448
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

方法	预训练数据	ViT-B	ViT-L	ViT-H	ViT-H448
从头开始，我们的实现	-	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.

表 3. 与之前在 ImageNet-1K 上的结果比较。预训练数据是 ImageNet-1K 训练集 (除了 BEiT 中的分词器是在 250M DALLE 数据上预训练的 [50])。所有自监督方法均通过端到端微调进行评估。ViT 模型为 B/16、L/16、H/14 [16]。每列的最佳结果用下划线标出。除 ViT-H 在 448 上有额外结果外，所有结果均在 224 的图像大小下。这里我们的 MAE 重建了归一化的像素，并进行了 1600 个周期的预训练。

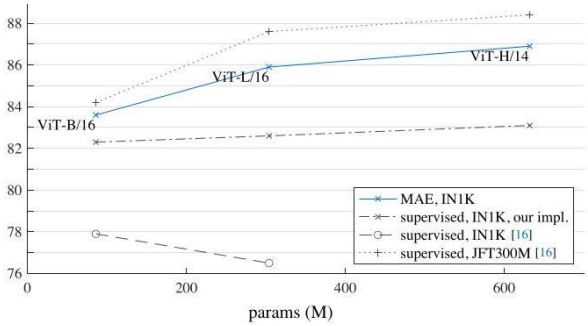

Figure 8. MAE pre-training vs. supervised pre-training, evaluated by fine-tuning in ImageNet-1K (224 size). We compare with the original ViT results [16] trained in IN1K or JFT300M.

图 8. MAE 预训练与监督预训练的比较，通过在 ImageNet-1K(224 大小) 中的微调进行评估。我们与在 IN1K 或 JFT300M 中训练的原始 ViT 结果 [16] 进行了比较。

4.2. Comparisons with Previous Results

4.2. 与之前结果的比较

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.

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，各方法之间的差距更大，这表明对于更大的模型来说，减少过拟合是一个挑战。

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 INIK 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.

我们的 MAE 可以轻松扩展，并且从更大的模型中显示出稳定的改进。我们使用 ViT-H(224 尺寸) 获得了 86.9% 的准确率。通过使用 448 尺寸进行微调，我们仅使用 INIK 数据达到了 **87.8**% 的准确率。在所有仅使用 IN1K 数据的方法中，之前的最佳准确率为 87.1%(512 尺寸)[67]，基于先进的网络。我们在高度竞争的 IN1K 基准测试中以非微不足道的幅度超越了最先进的技术 (没有外部数据)。我们的结果基于普通的 ViT，我们预计先进的网络会表现得更好。

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.

与 BEiT [2] 相比，我们的 MAE 更加准确，同时更简单和更快。我们的方法重建像素，而 BEiT 则预测标记:BEiT 在使用 ViT-B 重建像素时报告了 1.8% 的降级 [2]。我们不需要 dVAE 预训练。此外，我们的 MAE 每个 epoch 的速度明显快于 BEiT，原因如表 1c 所研究。

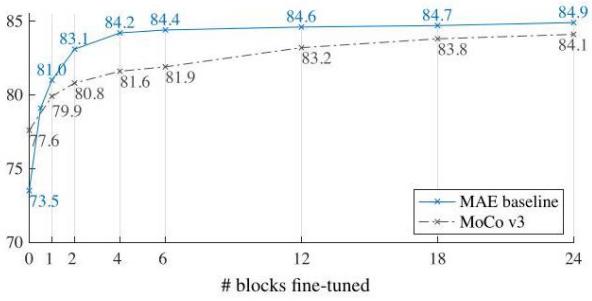

Figure 9. Partial fine-tuning results of ViT-L w.r.t. the number of fine-tuned Transformer blocks under the default settings from Table 1. Tuning 0 blocks is linear probing; 24 is full fine-tuning. Our MAE representations are less linearly separable, but are consistently better than MoCo v3 if one or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

图 9. ViT-L 在默认设置下微调的 Transformer 块数量的部分微调结果，参考表 1。微调 0 个块是线性探测；24 是完全微调。我们的 MAE 表示的线性可分性较差，但如果微调一个或多个块，则始终优于 MoCo v3。

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].

表 3 中的 MAE 模型经过 1600 轮的预训练以提高准确性 (图 7)。即便如此，当在相同硬件上训练时，我们的总预训练时间仍低于其他方法。例如，在 128 个 TPU-v3 核心上训练 ViT-L，我们的 MAE 训练时间为 1600 轮 31 小时，而 MoCo v3 的训练时间为 300 轮 36 小时 [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.

与监督预训练的比较。在原始 ViT 论文 [16] 中，ViT-L 在 IN1K 上训练时性能下降。我们实现的监督训练 (见 A.2) 效果更好，但准确性趋于饱和。见图 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.

我们的 MAE 预训练仅使用 IN1K，能够更好地泛化: 对于更高容量的模型，从头开始训练的增益更大。它遵循与 [16] 中 JFT-300M 监督预训练类似的趋势。这一比较表明我们的 MAE 可以帮助扩大模型规模。

4.3. Partial Fine-tuning

4.3. 部分微调

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].

表 1 显示线性探测和微调结果在很大程度上不相关。线性探测在过去几年中一直是一个流行的协议；然而，它错失了追求强大但非线性特征的机会——这实际上是深度学习的一大优势。作为折中方案，我们研究了一种部分微调协议: 微调最后几层，同时冻结其他层。该协议在早期工作中也曾使用，例如 [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.

图 9 显示了结果。值得注意的是，仅微调一个 Transformer 块显著提高了准确性，从 73.5% 提升到 81.0% 。此外，如果我们仅微调最后一个块的 "半部分" (即其 MLP 子块)，我们可以得到 79.1% ，这比线性探测要好得多。这个变体本质上是微调一个 MLP 头。微调几个块 (例如，4 或 6 个) 可以达到接近完全微调的准确性。

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.

在图 9 中，我们还与 MoCo v3 [9] 进行了比较，后者是一种对比方法，具有可用的 ViT-L 结果。MoCo v3 的线性探测准确性更高；然而，它的所有部分微调结果都不如 MAE。当微调 4 个块时，差距为 2.6% 。虽然 MAE 表示的线性可分性较差，但它们是更强的非线性特征，并且在微调非线性头时表现良好。

[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.

method	pre-train data	APbox		APmask	
		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

方法	预训练数据	APbox		APmask	
		ViT-B	ViT-L	ViT-B	ViT-L
监督	带标签的 IN1K	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.

表 4. 使用 ViT Mask R-CNN 基线的 COCO 目标检测和分割。所有条目均基于我们的实现。自监督条目使用无标签的 IN1K 数据。Mask AP 与框 AP 遵循类似的趋势。

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.

这些观察结果表明，线性可分性并不是评估表示质量的唯一指标。也有观察到 (例如，[8]) 线性探测与迁移学习性能 (例如，目标检测) 并不高度相关。据我们所知，线性评估在 NLP 中并不常用于基准预训练。

5. Transfer Learning Experiments

5. 迁移学习实验

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

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。

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.

目标检测与分割。我们在 COCO [37] 上对 Mask R-CNN [24] 进行了端到端的微调。ViT 主干被调整以与 FPN [36] 一起使用 (见 A.3)。我们对表 4 中的所有条目应用这种方法。我们报告目标检测的框 AP 和实例分割的掩码 AP。

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 pretraining (50.3 vs. 47.9, $\mathrm{AP}^{\mathrm{box}}$). More significantly, with the larger ViT-L, our MAE pre-training outperforms supervised pre-training by 4.0 points (53.3 vs. 49.3).

与监督预训练相比，我们的 MAE 在所有配置下表现更好 (表 4)。使用较小的 ViT-B，我们的 MAE 比监督预训练高出 2.4 分 (50.3 对比 47.9, $\mathrm{AP}^{\mathrm{box}}$)。更显著的是，使用更大的 ViT-L，我们的 MAE 预训练比监督预训练高出 4.0 分 (53.3 对比 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.

基于像素的 MAE 优于或与基于标记的 BEiT 相当，而 MAE 更加简单和快速。MAE 和 BEiT 都优于 MoCo v3，而 MoCo v3 与监督预训练相当。

Semantic segmentation. We experiment on ADE20K [72] using UperNet [63] (see A.4). Table 5 shows that our pretraining significantly improves results over supervised pretraining, 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.

[2] 我们在 ViT-L 的 BEiT 中也观察到了降级: 它生成了 85.2% (标记) 和 83.5% (像素)，这是从官方代码中复现的。

语义分割。我们在 ADE20K [72] 上使用 UperNet [63] 进行实验 (见 A.4)。表 5 显示我们的预训练显著改善了监督预训练的结果，例如，ViT-L 提高了 3.7 分。我们的基于像素的 MAE 也优于基于标记的 BEiT。这些观察结果与 COCO 中的结果一致。

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.

分类任务。表 6 研究了在 iNaturalists [56] 和 Places [71] 任务上的迁移学习 (见 A.5)。在 iNat 上，我们的方法表现出强大的扩展性: 随着模型规模的增大，准确性显著提高。我们的结果大幅超越了之前的最佳结果。在 Places 上，我们的 MAE 超越了通过在数十亿张图像上进行预训练获得的之前最佳结果 [19, 40]。

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.

像素与标记。表 7 比较了像素与标记作为 MAE 重建目标。使用 dVAE 标记优于使用未归一化的像素，但在我们测试的所有情况下，它在统计上与使用归一化像素相似。这再次表明，标记化对于我们的 MAE 并不是必要的。

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

方法	预训练数据	ViT-B	ViT-L
监督学习	带标签的 IN1K	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 Uper-Net. BEiT results are reproduced using the official code. Other entries are based on our implementation. Self-supervised entries use IN1K data without labels.

表 5. 使用 Uper-Net 的 ADE20K 语义分割 (mIoU)。BEiT 的结果是使用官方代码重现的。其他条目基于我们的实现。自监督条目使用没有标签的 IN1K 数据。

dataset	ViT-B	ViT-L	ViT-H	ViT-H448	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]*

数据集	ViT-B	ViT-L	ViT-H	ViT-H448	之前的最佳
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.

表 6. 在分类数据集上使用在 IN1K 上预训练并随后微调的 MAE 的迁移学习准确性。我们提供与之前最佳结果的系统级比较。

† : pre-trained on 1 billion images. ‡ : pre-trained on 3.5 billion images.

† : 在 10 亿张图像上预训练。‡ : 在 35 亿张图像上预训练。

	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

	IN1K			COCO		ADE20K	
	ViT-B	ViT-L	ViT-H	ViT-B	ViT-L	ViT-B	ViT-L
像素 (无归一化)	83.3	85.1	86.2	49.5	52.8	48.0	51.8
像素 (有归一化)	83.6	85.9	86.9	50.3	53.3	48.1	53.6
dVAE 令牌	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.

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

6. Discussion and Conclusion

6. 讨论与结论

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.

可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 (NLP) 中，简单的自监督学习方法 (例如，[47, 14, 48, 4]) 使得从指数级扩展模型中受益成为可能。在计算机视觉中，尽管自监督学习取得了进展，但实用的预训练范式主要是监督的 (例如 [33, 51, 25, 16])。在本研究中，我们在 ImageNet 和迁移学习中观察到，自编码器——一种与 NLP 技术类似的简单自监督方法——提供了可扩展的好处。视觉中的自监督学习可能现在正走上与 NLP 相似的轨迹。

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.

另一方面，我们注意到图像和语言是不同性质的信号，这种差异必须谨慎处理。图像仅仅是记录的光，没有进行语义分解成词的视觉类比。我们不是试图去除对象，而是去除最可能不形成语义片段的随机区域。同样，我们的 MAE 重建的像素并不是语义实体。然而，我们观察到 (例如，图 4) 我们的 MAE 推断出复杂的整体重建，这表明它已经学习了众多视觉概念，即语义。我们假设这种行为是通过 MAE 内部丰富的隐藏表示实现的。我们希望这种视角能够激励未来的工作。

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.

更广泛的影响。所提出的方法基于训练数据集的学习统计来预测内容，因此会反映这些数据中的偏见，包括对社会产生负面影响的偏见。该模型可能生成不存在的内容。这些问题在基于此工作生成图像时需要进一步研究和考虑。

References

参考文献

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer

[1] Jimmy Lei Ba, Jamie Ryan Kiros, 和 Geoffrey E Hinton. 层

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. Ran-daugment: 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 De-hghani, 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 Gir-shick. 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.

[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. Mi-crosoft 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 inpaint-ing. In CVPR, 2016.

[47] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pretraining. 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 Be-longie. 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 au-toencoders: 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.

A. Implementation Details

A. 实施细节

A.1. ImageNet Experiments

A.1. ImageNet 实验

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]).

ViT 架构。我们遵循标准的 ViT 架构 [16]。它有一堆 Transformer 块 [57]，每个块由一个多头自注意力块和一个 MLP 块组成，两者都具有层归一化 (LN) [1]。编码器以 LN 结束。由于 MAE 编码器和解码器的宽度不同，我们在编码器之后采用线性投影层以匹配它。我们的 MAE 在编码器和解码器输入中添加位置嵌入 [57](正弦-余弦版本)。我们的 MAE 不使用相对位置或层缩放 (这些在 [2] 的代码中使用)。

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).

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 具有类标记 [16]，为了适应这一设计，在我们的 MAE 预训练中，我们在编码器输入中附加了一个辅助虚拟标记。该标记将在微调和线性探测中被视为训练分类器的类标记。我们的 MAE 在没有该标记的情况下 (使用平均池化) 同样有效。

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 $\_lr \times$ batchsize / 256 .

预训练。默认设置见表 8。我们不使用颜色抖动、路径丢弃或梯度裁剪。我们使用 xavier_uniform [18] 来初始化所有 Transformer 块，遵循 ViT 的官方代码 [16]。我们使用线性 lr 缩放规则 [20]: $lr =$ base $\_lr \times$ 批量大小 / 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].

端到端微调。我们的微调遵循监督 ViT 训练的常见实践。默认设置见表 9。我们使用逐层学习率衰减 [10]，遵循 [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.

线性探测。我们的线性分类器训练遵循 [9]。见表 10。我们观察到线性探测需要与端到端微调非常不同的策略。特别是，正则化通常对线性探测是有害的。遵循 [9]，我们禁用许多常见的正则化策略: 我们不使用 mixup [69]、cutmix [68]、路径丢弃 [30] 或颜色抖动，并将权重衰减设置为零。

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.

在训练经典线性分类器 (例如，SVM [11]) 时，规范化分类器输入是一种常见做法。同样，在训练线性探测分类器时，规范化预训练特征也是有益的。遵循 [15]，我们采用一个额外的 BatchNorm 层 [31]，不进行仿射变换 (affine=False)。该层应用于编码器生成的预训练特征，并位于线性分类器之前。我们注意到，该层不会破坏线性特性，并且在训练后可以被吸收进线性分类器: 它本质上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的消融实验中校准不同变体之间的特征幅度，以便它们可以在没有进一步 lr 搜索的情况下使用相同的设置。

config	value
optimizer	AdamW [39]
base learning rate	1.5e-4
weight decay	0.05
optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.95$ [6]
batch size	4096
learning rate schedule	cosine decay [38]
warmup epochs [20]	40
augmentation	RandomResizedCrop

配置	值
优化器	AdamW [39]
基础学习率	1.5e-4
权重衰减	0.05
优化器动量	$\beta_1, \beta_2 = 0.9, 0.95$ [6]
批量大小	4096
学习率调度	余弦衰减 [38]
热身周期 [20]	40
数据增强	随机调整大小裁剪

Table 8. Pre-training setting.
表 8. 预训练设置。

config	value
optimizer	AdamW
base learning rate	1e-3
weight decay	0.05
optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.999$
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)

配置	值
优化器	AdamW
基础学习率	1e-3
权重衰减	0.05
优化器动量	$\beta_1, \beta_2 = 0.9, 0.999$
层级学习率衰减 [10, 2]	0.75
批量大小	1024
学习率调度	余弦衰减
预热周期	5
训练周期	100 (B) , 50 (L/H)
数据增强	RandAug (9, 0.5) [12]
标签平滑 [52]	0.1
混合 [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.
表 9. 端到端微调设置。

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

配置	值
优化器	LARS [66]
基础学习率	0.1
权重衰减	0
优化器动量	0.9
批量大小	16384
学习率调度	余弦衰减
热身周期	10
训练周期	90
数据增强	随机缩放裁剪

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

表 10. 线性探测设置。我们使用 LARS 和大批量进行更快的训练；SGD 在 4096 批量下工作类似。

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.

部分微调。我们的 MAE 部分微调 (§4.3) 遵循表 9 中的设置，除了我们调整微调的周期数。我们观察到，微调较少的块需要更长的时间安排。我们将微调周期数设置为 $\{50, 100, 200\}$ ，并为每个微调块的数量使用最佳值。

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

A.2. 从头开始监督训练 ViT-L/H

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.

我们发现从头开始在 ImageNet-1K 上训练监督 ViT-L/H 并非易事。训练不稳定。虽然对于较小的模型已有强有力的基准和公开可用的实现 [53]，但对于较大的 ViT-L/H 的训练方案尚未被探索。直接将之前的方案应用于这些较大的模型并不起作用。在训练过程中经常观察到 NaN 损失。

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 $\beta_2 = 0.95$ 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.

我们在表 11 中提供了我们的配方。我们使用 0.3 的权重衰减，批量大小为 4096，并且进行长时间的预热，遵循原始的 ViT [16]。我们使用 $\beta_2 = 0.95$ ，遵循 [6]。我们使用表 11 中列出的正则化方法，并禁用其他方法，遵循 [64]。所有这些选择都是为了提高训练的稳定性。我们的配方可以在没有 NaN 损失的情况下完成训练。

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

[3] 另外，我们可以预先计算特征的均值和标准差，并使用归一化的特征来训练线性分类器。

config	value
optimizer	AdamW
base learning rate	1e-4
weight decay	0.3
optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.95$
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

配置	值
优化器	AdamW
基础学习率	1e-4
权重衰减	0.3
优化器动量	$\beta_1, \beta_2 = 0.9, 0.95$
批量大小	4096
学习率调度	余弦衰减
预热周期	20
训练周期	300 (B), 200 (L/H)
数据增强	RandAug $(9, 0.5)$ [12]
标签平滑 [52]	0.1
mixup [69]	0.8
cutmix [68]	1.0
drop path [30]	0.1 (B) , 0.2 (L/H)
指数移动平均 (EMA)	0.9999

Table 11. Supervised training ViT from scratch.

表 11. 从头开始的监督训练 ViT。

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.

ViT-L 的准确率为 82.6% (81.5% 不使用 EMA)，ViT-H 的准确率为 83.1% (80.9% 不使用 EMA)。如果不使用 EMA，ViT-L 和 ViT-H 都显示出过拟合的趋势。

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

作为副产品，我们的 ViT-B 配方具有 82.3% 的准确率 (82.1% 不使用 EMA)，而 [53] 中的准确率为 81.8%。

A.3. Object Detection and Segmentation in COCO

A.3. COCO 中的目标检测和分割

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.

我们为 Mask R-CNN [24] 中的 FPN 骨干网 [36] 调整了原始的 ViT。ViT 有一堆 Transformer 块，这些块都在单一尺度 (例如，步幅为 16) 下生成特征图。我们将这堆块平均分成 4 个子集，并应用卷积来上采样或下采样中间特征图，以生成不同的尺度 (步幅为 4、8、16 或 32，与标准 ResNet [25] 相同)。FPN 是基于这些多尺度图构建的。

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].

为了在不同方法之间进行公平比较，我们为表 4 中的每个条目 (包括所有竞争者) 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。我们将发布代码以及具体配置。有关完整模型和训练细节以及额外实验，请参见 [35]。

A.4. Semantic Segmentation in ADE20K

A.4. ADE20K 中的语义分割

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).

我们使用 UperNet [63]，遵循 [2] 的语义分割代码。我们以批量大小为 16 进行端到端的微调，持续 100 个周期。我们为表 5 中的每个条目 (包括所有竞争者) 搜索最佳 *lr* 。

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 pretraining and fine-tuning, following their code.

[2] 的语义分割代码使用相对位置偏差 [49]。我们的 MAE 预训练不使用它。为了公平比较，我们仅在迁移学习期间启用相对位置偏差，并将其初始化为零。我们注意到我们的 BEiT 重现同时在预训练和微调中使用相对位置偏差，遵循他们的代码。

A.5. Additional Classification Tasks

A.5. 额外分类任务

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.

我们遵循表 9 中的设置进行 iNaturalist 和 Places 的微调 (表 6)。我们为每个单独的数据集调整 *lr* 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

方法	模型	参数	准确率
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.

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。† : 我们的实现。

dataset	ViT-B	ViT-L	ViT-H	ViT-H448	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		

数据集	ViT-B	ViT-L	ViT-H	ViT-H448	prev best
IN-腐败 ↓ [27]	51.7	41.8	33.8	36.8	42.5 [32]
IN-对抗 [28]	35.9	57.1	68.2	76.7	35.8 [41]
IN-表现 [26]	48.3	59.9	64.4	66.5	48.7 [41]
IN-草图 [60]	34.5	45.3	49.6	50.9	36.0 [41]
我们的监督训练基准:					
IN-腐败 ↓	45.8	42.3	41.3		
IN-对抗	27.2	29.6	33.1		
IN-表现	49.4	50.9	50.3		
IN-草图	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 Im-ageNet validation sets, without any specialized fine-tuning. We provide system-level comparisons with the previous best results.

表 13. 在 ImageNet 变体上的鲁棒性评估 (top-1 准确率，IN-C [27] 除外，该评估平均损坏错误)。我们在不同的 ImageNet 验证集上测试相同的 MAE 模型 (表 3)，没有任何专门的微调。我们提供与之前最佳结果的系统级比较。

B. Comparison on Linear Probing Results

B. 线性探测结果的比较

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.

在 §4.3 中，我们已经表明线性探测准确性和微调准确性在很大程度上是无关的，它们对线性可分性的关注点不同。我们注意到，现有的掩码图像编码方法在线性探测中通常竞争力较弱 (例如，相较于对比学习)。为了完整性，在表 12 中，我们比较了基于掩码的方法的线性探测准确性。

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).

我们的 MAE 与 ViT-L 的线性探测准确性为 75.8%。这显著优于之前的基于掩码的方法。另一方面，在该协议下，它仍然落后于对比方法: 例如，MoCo v3 [9] 在 ViT-L 上的线性探测准确性为 77.6%(图 9)。

C. Robustness Evaluation on ImageNet

C. 在 ImageNet 上的鲁棒性评估

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.

在表 13 中，我们评估了我们的模型在不同变体的 ImageNet 验证集上的鲁棒性。我们使用在原始 ImageNet 上微调的相同模型 (表 3)，并仅对不同的验证集进行推理，而不进行任何专门的微调。表 13 显示我们的方法具有强大的扩展性: 增加模型大小带来了显著的收益。增加图像大小在所有集合中都有帮助，但 IN-C 除外。我们的结果大幅超越了之前最佳结果 (专门系统的结果)。

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.

相比之下，监督训练的表现要差得多 (表 13 底部；在 A.2 中描述的模型)。例如，使用 ViT-H，我们的 MAE 预训练在 IN-A 上的表现为 35% 优于监督对应模型 (68.2% 对 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. 在 ImageNet 验证图像上的未整理随机样本。对于每个三元组，我们展示了掩码图像 (左)，我们的 MAE 重建 (中)，以及真实值 (右)。掩码比例为 75% 。

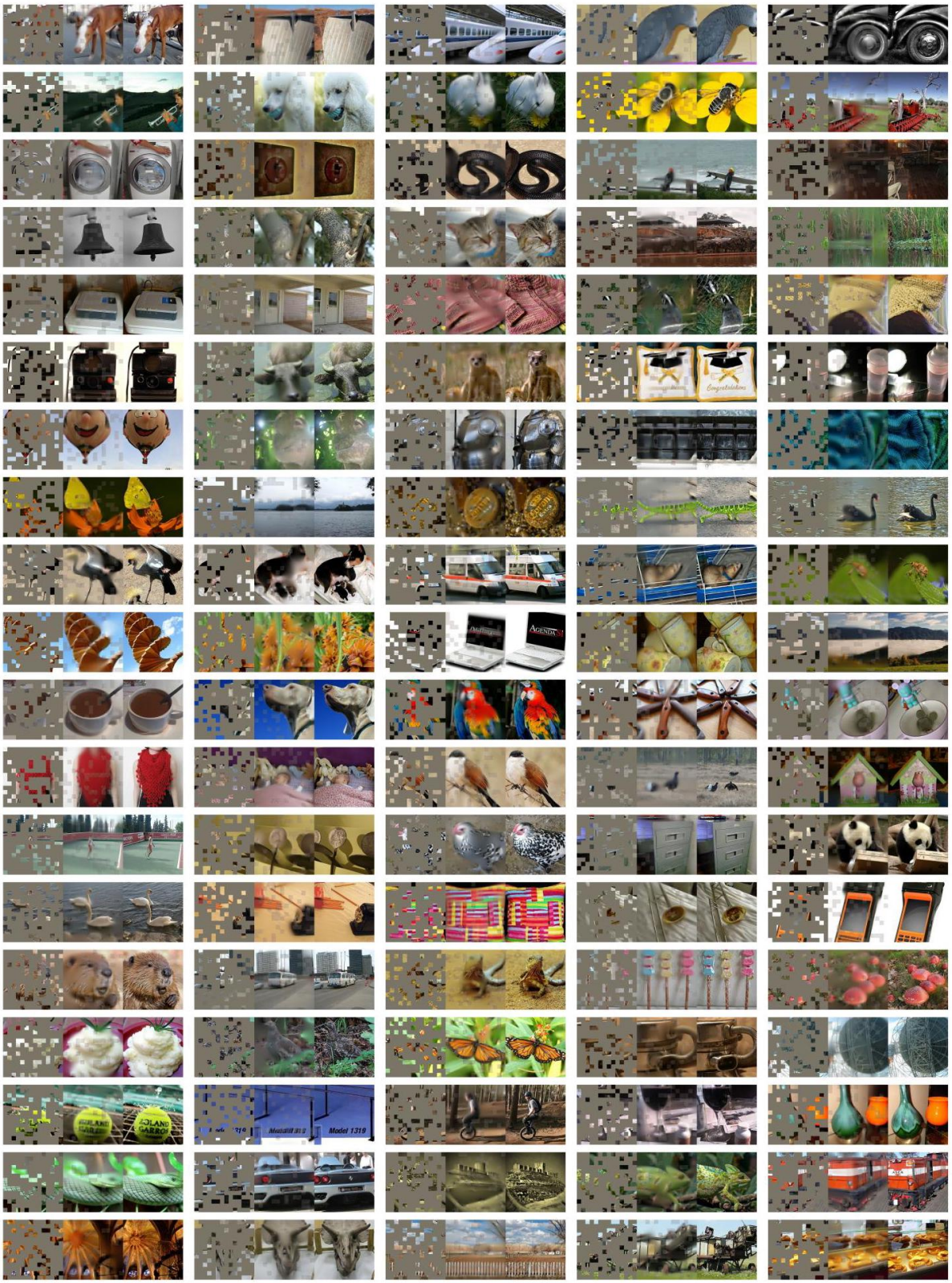

Figure 10. **Uncurated random samples** on ImageNet *validation* images. For each triplet, we show the masked image (left), our MAE

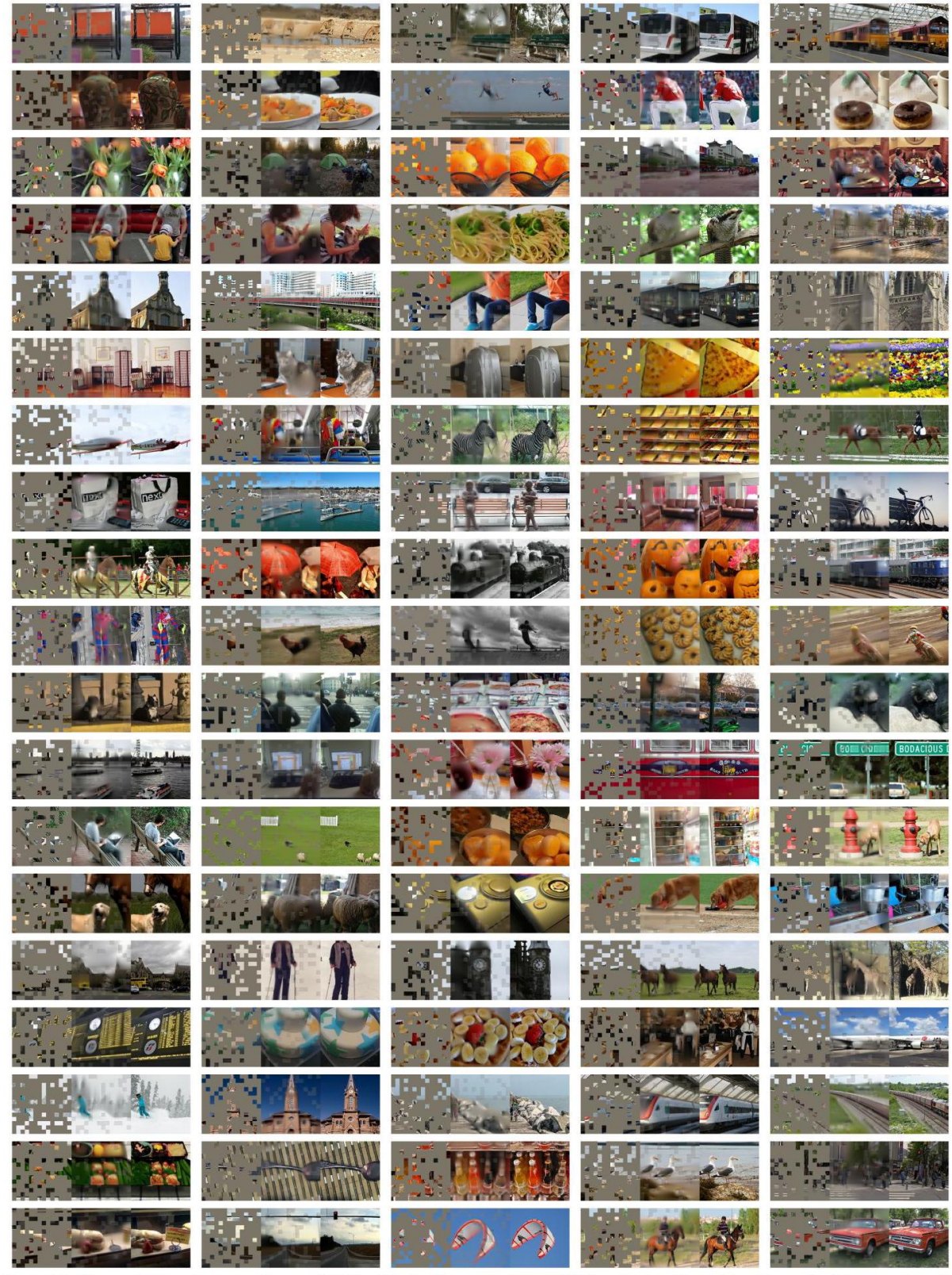

Figure 11. Uncurated random samples on COCO validation images, using an MAE trained on ImageNet. For each triplet, we show the masked image (left), our MAE reconstruction (middle), and the ground-truth (right). The masking ratio is 75%.

图 11. 在 COCO 验证图像上使用在 ImageNet 上训练的 MAE 进行的未整理随机样本。对于每个三元组，我们展示了被遮挡的图像 (左)，我们的 MAE 重建 (中)，以及真实值 (右)。遮挡比例为 75%。