International Journal of Computer Vision (2024) 132:208–223
https://doi.org/10.1007/s11263-023-01852-4

Context Autoencoder for Self-supervised Representation Learning

Xiaokang Chen[1] · Mingyu Ding[2,3] · Xiaodi Wang[4] · Ying Xin[4] · Shentong Mo[4] · Yunhao Wang[4] · Shumin Han[4] · Ping Luo[2] · Gang Zeng[1] · Jingdong Wang[4]

Received: 27 January 2023 / Accepted: 12 July 2023 / Published online: 28 August 2023

Abstract
We present a novel masked image modeling (MIM) approach, context autoencoder (CAE), for self-supervised representation pretraining. We pretrain an encoder by making predictions in the encoded representation space. The pretraining tasks include two tasks: masked representation prediction—predict the representations for the masked patches, and masked patch reconstruction—reconstruct the masked patches. The network is an encoder–regressor–decoder architecture: the encoder takes the visible patches as input; the regressor predicts the representations of the masked patches, which are expected to be aligned with the representations computed from the encoder, using the representations of visible patches and the positions of visible and masked patches; the decoder reconstructs the masked patches from the predicted encoded representations. The CAE design encourages the separation of learning the encoder (representation) from completing the pertaining tasks: masked representation prediction and masked patch reconstruction tasks, and making predictions in the encoded representation space empirically shows the benefit to representation learning. We demonstrate the effectiveness of our CAE through superior transfer performance in downstream tasks: semantic segmentation, object detection and instance segmentation, and classification. The code will be available at https://github.com/Atten4Vis/CAE.

Keywords Self-supervised representation learning · Masked image modeling · Context autoencoder

1 Introduction

We study the masked image modeling (MIM) task for self-supervised representation learning. It aims to learn an encoder through masking some patches of the input image and making predictions for the masked patches from the visible patches. It is expected that the resulting encoder pretrained through solving the MIM task is able to extract the patch representations taking on semantics that are transferred to solving downstream tasks.

The typical MIM methods, such as BEiT (Bao et al., 2021), the method studied in the ViT paper (Dosovitskiy et al., 2021), and iBoT (Zhou et al., 2021), use a single ViT architecture to solve the pretraining task i.e., reconstructing the patch tokens or the pixel colors. These methods mix the two tasks: learning the encoder (representation) and reconstructing the masked patch. The subsequent method, masked autoencoder (MAE) (He et al., 2022) adopts an encoder-decoder architecture, partially decoupling the two tasks. As a result, the representation quality is limited. Most previous methods, except iBoT (Zhou et al., 2021), lack an explicit modeling between encoded representations of visible patches and masked patches.

We present a context autoencoder (CAE) approach, illustrated in Fig. 1, for improving the encoding quality. We pretrain the encoder through making predictions for the masked patches in the encoded representation space. The pretraining task is a combination of masked representation prediction and masked patch reconstruction. the pretraining network is an encoder–regressor–decoder architecture. The encoder takes only the visible patches as input and learns the representations only for the visible patches. The regressor predicts the masked patch representations, which is expected to be aligned with the representations of the masked patches computed from the encoder, from the visible patch represen-

Communicated by Chunyuan Li.

✉ Jingdong Wang
wangjingdong@outlook.com

[1] Peking University, Beijing, China
[2] University of Hong Kong, Pok Fu Lam , Hong Kong
[3] UC Berkeley, Berkeley, CA, USA
[4] Baidu, Beijing, China

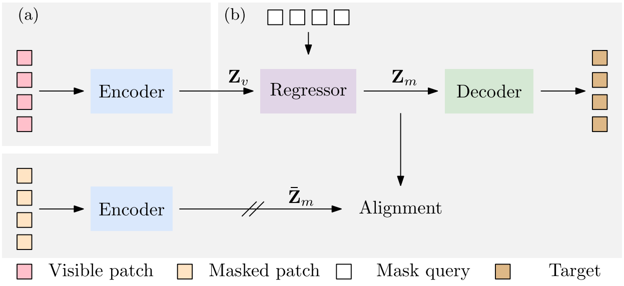

Fig. 1 The pipeline of context autoencoder. Our approach **a** feeds visible patches into the encoder and extracts their representations $\mathbf{Z}_v$ and then (**b**) completes the pretext tasks: predict the representations $\mathbf{Z}_m$ of the masked patches from the visible patches in the encoded representation space through latent contextual regressor and prediction alignment, and reconstruct the masked patches from the predicted representations $\mathbf{Z}_m$ of masked patches. The pretrained encoder in **a** is applied to downstream tasks by simply replacing the pretext task part (**b**) with the downstream task part. // means stop gradient

tations. The decoder reconstructs the masked patches from the predicted masked patch representations without receiving the representations of the visible patches.

The prediction in the encoded representation space from the visible patches to the masked patches generates a plausible semantic guess for the masked patches, which lies in the same semantic space for the visible patches. We assume that the prediction is easier if the encoded representations take higher semantics and that the accurate prediction encourages that the encoded representations take on a larger extent of semantics, empirically validated by the experiments.

The CAE design also encourages the separation of learning the encoder and completing the pretraining tasks: the responsibility of representation learning is mainly taken by the encoder and the encoder is only for representation learning. The reasons include: the encoder in the top stream in Fig. 1 operates only on visible patches, only focusing on learning semantic representations; the regression is done on the encoded representation space, as a mapping between the representations of the visible patches and the masked patches; the decoder operates only on the predicted representations of the masked patches.

We present the empirical performance of our approach on downstream tasks, semantic segmentation, object detection and instance segmentation, and classification. The results show that our approach outperforms supervised pretraining, contrastive self-supervised pretraining, and other MIM methods.

2 Related Work

Self-supervised representation learning has been widely studied in computer vision, including: context prediction (Doersch et al., 2015; Tian et al., 2021), clustering-based methods (Xie et al., 2016; Yang et al., 2016; Caron et al., 2018; Asano et al., 2019; Zhuang et al., 2019; Huang et al., 2019; Caron et al., 2019; Goyal et al., 2021), contrastive self-supervised learning (Li et al., 2022b; van den Oord et al., 2018; Henaff, 2020; Wang et al., 2022), instance discrimination (Dosovitskiy et al., 2014, 2015), image discretization (Gidaris et al., 2020, 2022), masked image modeling (Li et al., 2021; Fang et al., 2022; Tian et al., 2022), and information maximization (Ermolov et al., 2021; Zbontar et al., 2021; Bardes et al., 2021). The following mainly reviews closely-related methods.

Autoencoding Traditionally, autoencoders were used for dimensionality reduction or feature learning (LeCun, 1987; Gallinari et al., 1987; Hinton & Zemel, 1994; Hinton & Salakhutdinov, 2006; Ranzato et al., 2007; Vincent et al., 2008; Kingma & Welling, 2013). The denoising autoencoder (DAE) is an autoencoder that receives a corrupted data point as input and is trained to estimate the original, uncorrupted data point as its output. The variants or modifications of DAE were adopted for self-supervised representation learning, e.g., corruption by masking pixels (Vincent et al., 2010; Pathak et al., 2016; Chen et al., 2020), removing color channels (Zhang et al., 2016), shuffling image patches (Noroozi & Favaro, 2016), denoising pixel-level noise (Atito et al., 2021) and so on.

Contrastive self-supervised learning Contrastive self-super vised learning, referring in this paper to the self-supervised approaches comparing random views with contrastive loss or simply MSE loss that are related as shown in Garrido et al. (2022), has been popular for self-supervised representation learning (Chen et al., 2020; He et al., 2020; Tian et al., 2020; Chen et al., 2021; Grill et al., 2020; Caron et al., 2021; Chen & He, 2021; Caron et al., 2020; Wu et al., 2018; Peng et al., 2022). The basic idea is to maximize the similarity between the views augmented from the same image and optionally minimize the similarity between the views augmented from different images. Random cropping is an important augmentation scheme, and thus typical contrastive self-supervised learning methods (e.g., MoCo v3) tend to learn knowledge mainly from the central regions of the original images. Some dense variants (Wang et al., 2021; Xie et al., 2021a) eliminate the tendency in a limited degree by considering an extra contrastive loss with dense patches.

Masked image modeling Motivated by BERT for masked language modeling (Devlin et al., 2019), the method studied in Dosovitskiy et al. (2021) and BEiT (Bao et al., 2021) use the ViT structure to solve the masked image modeling task, e.g., estimate the pixels or the discrete tokens. The follow-up work, iBOT (Zhou et al., 2021), combines the MIM method (BEiT) and a contrastive self-supervised approach (DINO (Caron et al., 2021)). But they do not have explicitly an encoder for representation learning or a decoder for pretraining task completion, and the ViT structure is essentially a mixture of encoder and decoder, limiting the representation learning quality.

Several subsequent MIM methods are developed to improve the encoder quality, such as designing pretraining architectures: Masked Autoencoder (MAE) (He et al., 2022), SplitMask (El-Nouby et al., 2021), and Simple MIM (SimMIM) (Xie et al., 2021b); adopting new reconstruction targets: Masked Feature Prediction (MaskFeat) (Wei et al., 2021), Perceptual Codebook for BEiT (PeCo) (Dong et al., 2021), and data2vec (Baevski et al., 2022). The technical report[1] of our approach was initially published as an arXiv paper (Chen et al., 2022), and was concurrent to data2vec (Baevski et al., 2022), MAE (He et al., 2022), and other methods, such as (El-Nouby et al., 2021; Xie et al., 2021b). After that, MIM methods have developed rapidly, e.g., extended to frequency/semantic domain (Xie et al., 2022; Liu et al., 2022; Wei et al., 2022; Li et al., 2022e), combined with contrastive self-superivsed learning (Tao et al., 2022; Jing et al., 2022; Yi et al., 2022; Huang et al., 2022), efficient pretraining (Zhang et al., 2022a; Huang et al., 2022; Chen et al., 2022), mask strategy design (Kakogeorgiou et al., 2022; Li et al., 2022a, d), scalability of MIM (Xie et al., 2022), and interpretation of MIM (Xie et al., 2022; Li et al., 2022c; Kong & Zhang, 2022).

The core idea of our approach is making predictions in the encoded representation space. We jointly solve two pretraining tasks: masked representation prediction–predict the representations for the masked patches, where the representations lie in the representation space output from the encoder, and masked patch reconstruction - reconstruct the masked patches.

Our approach is clearly different from MAE (He et al., 2022) (Fig. 2 (top)). Our approach introduces an extra pretraining task, masked representation prediction, and encourages the separation of two roles: learning the encoder and completing pretraining tasks; in contrast, MAE partially mixes the two roles, and has no explicit prediction of masked patch representations.

On the other hand, our approach differs from data2vec (Baevski et al., 2022) and iBoT (Zhou et al., 2021) (Fig. 2 (bottom)). Similar to BEiT, in data2vec and iBoT, there is no explicit module separation of learning the encoder and estimating the mask patch representations, and the target representations are formed from the full view (as the teacher) with the same network as the student network for processing the masked view and predicting the masked patch representations (except a centering process in iBoT for the teacher following DINO). In contrast, our approach is simple: form the target representations merely from the output of the encoder, and the encoder-regressor design is straightforward and explainable: the regressor predicts the representations of masked patches to match the representations computed directly from the encoder.

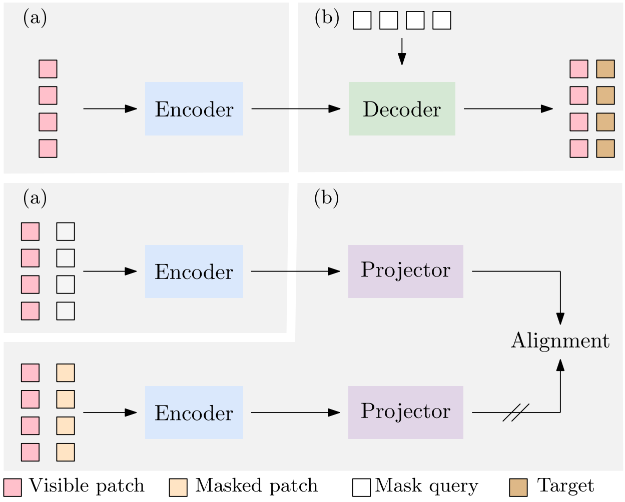

Fig. 2 The pipeline of MAE (top), and the MIM part of iBoT (bottom). The centering module is not depicted in the bottom stream. The pretrained encoder in **a** is applied to downstream tasks by simply replacing the pretext task part (**b**) with the downstream task part. // means stop gradient

3 Approach

3.1 Architecture

Our context autoencoder (CAE) is a masked image modeling approach. The network shown in Fig. 1 is an encoder-regressor-decoder architecture. The key is to make predictions from visible patches to masked patches in the encoded representation space. The pretraining tasks include: masked representation prediction and masked patch reconstruction.

We randomly split an image into two sets of patches: visible patches $\mathbf{X}_v$ and masked patches $\mathbf{X}_m$. The encoder takes the visible patches as input; the regressor predicts the representations of the masked patches, which are expected to be aligned with the representations computed from the encoder, from the representations of the visible patches conditioned on the positions of masked patches; the decoder reconstructs the masked patches from the predicted encoded representations.

Encoder The encoder $\mathcal{F}$ maps the visible patches $\mathbf{X}_v$ to the latent representations $\mathbf{Z}_v$. It only handles the visible patches. We use the ViT to form our encoder. It first embeds the visible patches by linear projection as patch embeddings, and adds the positional embeddings $\mathbf{P}_v$. Then it sends the combined embeddings into a sequence of transformer blocks that are based on self-attention, generating $\mathbf{Z}_v$.

Regressor The latent contextual regressor $\mathcal{H}$ predicts the latent representations $\mathbf{Z}_m$ for the masked patches from the

[1] https://arxiv.org/abs/2202.03026.

latent representations $\mathbf{Z}_v$ of the visible patches output from the encoder conditioned on the positions of the masked patches. We form the latent contextual regressor $\mathcal{H}$ using a series of transformer blocks that are based on cross-attention.

The initial queries $\mathbf{Q}_m$, called mask queries, are mask tokens that are learned as model parameters and are the same for all the masked patches. The keys and the values are the same before linear projection and consist of the visible patch representations $\mathbf{Z}_v$ and the output of the previous cross-attention layer (mask queries for the first cross-attention layer). The corresponding positional embeddings of the masked patches are considered when computing the cross-attention weights between the queries and the keys. In this process, the latent representations $\mathbf{Z}_v$ of the visible patches are not updated.

Decoder The decoder $\mathcal{G}$ maps the latent representations $\mathbf{Z}_m$ of the masked patches to some forms of masked patches, $\mathbf{Y}_m$. The decoder, similar to the encoder, is a stack of transformer blocks that are based on self-attention, followed by a linear layer predicting the targets. The decoder only receives the latent representations of the masked patches (the output of the latent contextual regressor), and the positional embeddings of the masked patches as input without directly using the information of the visible patches.

3.2 Objective Function

Masking Following BEiT (Bao et al., 2021), we adopt the random block-wise masking strategy (illustrated in Fig. 3) to split the input image into two sets of patches, visible and masked patches. For each image, 98 of 196 (14 × 14) patches are masked.

Targets The targets $\bar{\mathbf{Z}}_m$ for the representations of the masked patches are formed as follows. We feed the masked patches $\mathbf{X}_m$ into the encoder, which is the same as the one for encoding visible patches, and generate the representations $\bar{\mathbf{Z}}_m$ of the masked patches as the representation targets.

The targets $\bar{\mathbf{Y}}_m$ for the patch reconstruction are formed by the discrete tokenizer, e.g., the tokenizer trained with d-VAE on ImageNet-1K without using the labels or the DALL-E tokenizer (trained with d-VAE on 400M images) (Ramesh et al., 2021) used in BEiT (Bao et al., 2021). The input image is fed into the tokenizer, assigning a discrete token to each patch for forming the reconstruction targets $\bar{\mathbf{Y}}_m$.

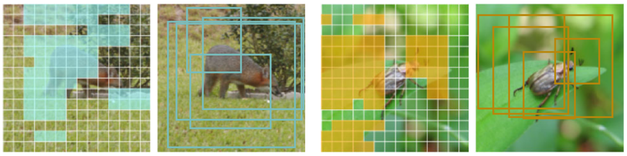

Fig. 3 Illustration of random block-wise sampling (1st and 3rd images) and random cropping (2nd and 4th images). The colored regions are masked regions. The boxes correspond to cropped regions. Random block-wise sampling is used in our approach. Random cropping is a key data-augmentation scheme for contrastive self-supervised pretraining

Loss function The loss function consists of a reconstruction loss: $\ell_y(\mathbf{Y}_m, \bar{\mathbf{Y}}_m)$, and an alignment loss: $\ell_z(\mathbf{Z}_m, \bar{\mathbf{Z}}_m)$, corresponding to masked patch reconstruction and masked representation prediction, respectively. The whole loss is a weighted sum:

$$\ell_y(\mathbf{Y}_m, \bar{\mathbf{Y}}_m) + \lambda\, \ell_z(\mathbf{Z}_m, \mathrm{sg}[\bar{\mathbf{Z}}_m]). \tag{1}$$

We use the MSE loss for $\ell_z(\mathbf{Z}_m, \bar{\mathbf{Z}}_m)$ and the cross-entropy loss for $\ell_y(\mathbf{Y}_m, \bar{\mathbf{Y}}_m)$. sg[·] stands for stop gradient. λ is 2 in our experiments.

4 Discussions

4.1 Analysis

Predictions are made in the encoded representation space Our CAE attempts to make predictions in the encoded representation space: predict the representations for the masked patches from the encoded representations of the visible patches. In other words, it is expected that the output representations of the latent contextual regressor also lie in the encoded representation space, which is ensured by prediction alignment. This encourages the learned representation to take on a large extent of semantics for prediction from visible patches to masked patches, benefiting the representation learning of the encoder.

We empirically verify that the predicted representations lie in the encoded representation space through image reconstruction. We train the CAE using the pixel colors as the prediction targets, for two cases: with and without the alignment, i.e., masked representation prediction. For reconstruction, we feed all the patches (without masking, all the image patches are visible) of an image (from the ImageNet validation set) into the pretrained encoder, then skip the latent contextual regressor and directly send all the encoded patch representations to the pretrained decoder for reconstructing the whole image.

Figure 4 provides reconstruction results for several examples randomly sampled from the ImageNet-1K validation set. One can see that our approach can successfully reconstruct the images, implying that the input and output representations of latent contextual regressor are in the same space. In contrast, without the alignment, the reconstructed images are noisy, indicating the input and output representations of latent contextual regressor are in different spaces. The results suggest that the explicit prediction alignment is critical for ensuring that predictions are made in the encoded representation space.

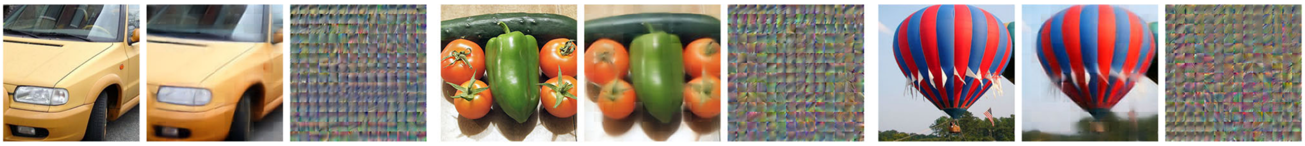

Fig. 4 Illustrating that predictions are made in the representation space. We reconstruct the image by feeding the full image (1st, 4th, and 7th) into the pretrained CAE encoder and then the pretrained CAE decoder outputting the reconstructed image (2nd, 5th, and 8th). It can be seen that the image can be constructed with the semantics kept when skipping latent contextual regressor, verifying the input and the predicted representations lie in the same space. We also show the reconstructed images (3rd, 6th, and 9th) from the encoder and the decoder pretrained without the alignment constraint. We can see that those images are meaningless, indicating that the alignment constraint is critical for ensuring that predictions are made in the representation space

Representation alignment in CAE and contrastive self-supervised learning Representation alignment is also used in contrastive self-supervised learning methods, such as MoCo, BYOL, SimCLR, and methods mixing contrastive self-supervised learning and masked image modeling, such as iBOT, and MST. The alignment loss could be the MSE loss or the contrastive loss that CAE may also take advantage of.

In the CAE, the alignment is imposed over the representations $\mathbf{Z}_m = \mathcal{H}(\mathcal{F}(\mathbf{X}_v))$ - predicted from the representations $\mathcal{F}(\mathbf{X}_v)$ of visible patches through the regressor $\mathcal{H}$, and the representations $\bar{\mathbf{Z}}_m = \mathcal{F}(\mathbf{X}_m)$ - computed from the encoder $\mathcal{F}$. Both $\mathbf{Z}_m$ *and* $\bar{\mathbf{Z}}_m$ *are about the masked patches, and lie in the representation space output from the encoder*.

Differently, the alignment in the most contrastive self-supervised learning methods is imposed over the representations $\{\mathcal{P}(\mathcal{F}(\mathbf{V}_1)), \mathcal{P}(\mathcal{F}(\mathbf{V}_2)), \cdots, \mathcal{P}(\mathcal{F}((\mathbf{V}_N))\}$, where $\mathcal{P}$ is a projector, and some views may be processed with the EMA version of the encoder and the projector. The N representations to be aligned are about *different views* $\{\mathbf{V}_1, \mathbf{V}_2, \cdots, \mathbf{V}_N\}$ (in iBoT and MST, the views are masked views and full views), and are not directly output from the encoder. It is not quite clear how the projector works, and it is reported in (Qi et al., 2023) that the projector is a part-to-whole process mapping the object part representation to the whole object representation for contrastive self-supervised learning.

4.2 Connection

Relation to autoencoder The original autoencoder (LeCun, 1987; Gallinari et al., 1987; Hinton & Zemel, 1994) consists of an encoder and a decoder. The encoder maps the input into a latent representation, and the decoder reconstructs the input from the latent representation. The denoising autoencoder (DAE) (Vincent et al., 2010), a variant of autoencoder, corrupts the input by adding noises and still reconstructs the non-corrupted input.

Our CAE encoder is similar to the original autoencoder and also contains an encoder and a decoder. Different from the autoencoder where the encoder and the decoder process the whole image, our encoder takes a portion of patches as input and our decoder takes the estimated latent representations of the other portion of patches as input. Importantly, the CAE makes predictions in the latent space from the visible patches to the masked patches.

Relation to BEiT, iBoT and MAE The CAE encoder processes the visible patches, to extract their representations, without making predictions for masked patches. Masked representation prediction is made through the regressor and the prediction alignment, ensuring that the output of the regressor lies in the representation space same with the encoder output. The decoder only processes the predicted representations of masked patches. Our approach encourages that the encoder takes the responsibility of and is only for representation learning.

In contrast, BEiT (Bao et al., 2021) and the MIM part of iBOT do not separate the representation extraction role and the task completion role and uses a single network, with both the visible and masked patches as the input, simultaneously for the two roles. In MAE (He et al., 2022), the so-called decoder may play a partial role for representation learning as the representations of the visible patches are also updated in the MAE decoder. Unlike CAE, MAE, iBoT, BEiT do not explicitly predict the representations of masked patches from the representations of visible patches (that lie in the encoded representation space) for masked patches.

When the pretrained encoder is applied to downstream tasks, one often replaces the pretext task completion part using the downstream task layer, e.g., segmentation layer or detection layer. The separation of representation learning (encoding) and pretext task completion helps that downstream task applications take good advantage of representation pretraining.

We provide the computational graph for CAE, BEiT (Bao et al., 2021), denoising autoencoder, Masked Autoencoder (He et al., 2022) and SplitMask (El-Nouby et al., 2021) (one stream) in Fig. 5. Compared to our CAE, the main issue of MAE is that the so-called decoder $\mathcal{R}$ might have also the encoding role, i.e., learning semantic representations of the visible patches.

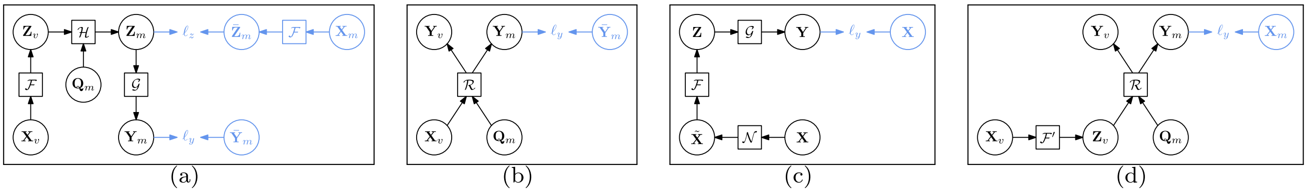

Fig. 5 The computational graphs for **a** a context autoencoder (CAE), **b** BEiT (Bao et al., 2021), **c** a denoising autoencoder (DAE), and **d** MAE (He et al., 2022) and the one stream in SplitMask (El-Nouby et al., 2021). The parts in cornflowerbluecornflower blue are for loss function. **a** The encoder $\mathcal{F}$ receives visible patches $\mathbf{X}_v$ and outputs their latent representations $\mathbf{Z}_v$. The latent contextual regressor $\mathcal{H}$ predicts the latent representations $\mathbf{Z}_m$ for masked patches from $\mathbf{Z}_v$. The decoder predicts the targets $\mathbf{Y}_m$ for masked patches from $\mathbf{Z}_m$. ℓ_z and ℓ_y are the loss functions. During training, the gradient is stopped for $\bar{\mathbf{Z}}_m$. See the detail in Sect. 3. **b** The input includes both visible patches $\mathbf{X}_v$ and mask queries $\mathbf{Q}_m$ representing masked patches, and the representations for them are updated within the function $\mathcal{R}$. **c** The function $\mathcal{N}$ is a noising function generating the noisy version $\tilde{\mathbf{X}}$ from the input $\mathbf{X}$. $\mathcal{F}$ and $\mathcal{G}$ are the normal encoder and decoder, respectively. **d** The two functions, $\mathcal{F}'$ and $\mathcal{R}$, are both based on self-attention. $\mathcal{F}'$ (called encoder in MAE) only processes the visible patches $\mathbf{X}_v$, and $\mathcal{R}$ (called decoder in MAE) processes both the latent representations $\mathbf{Z}_v$ of the visible patches and the mask queries ($\mathbf{Q}_m$) and updates them simultaneously. For simplicity, the positional embeddings are not included in computational graphs. **a** CAE and **c** DAE perform the encoding and MIM task completion roles explicitly and separately, **b** BEiT and **d** MAE perform the encoding and MIM task completion roles implicitly and simultaneously

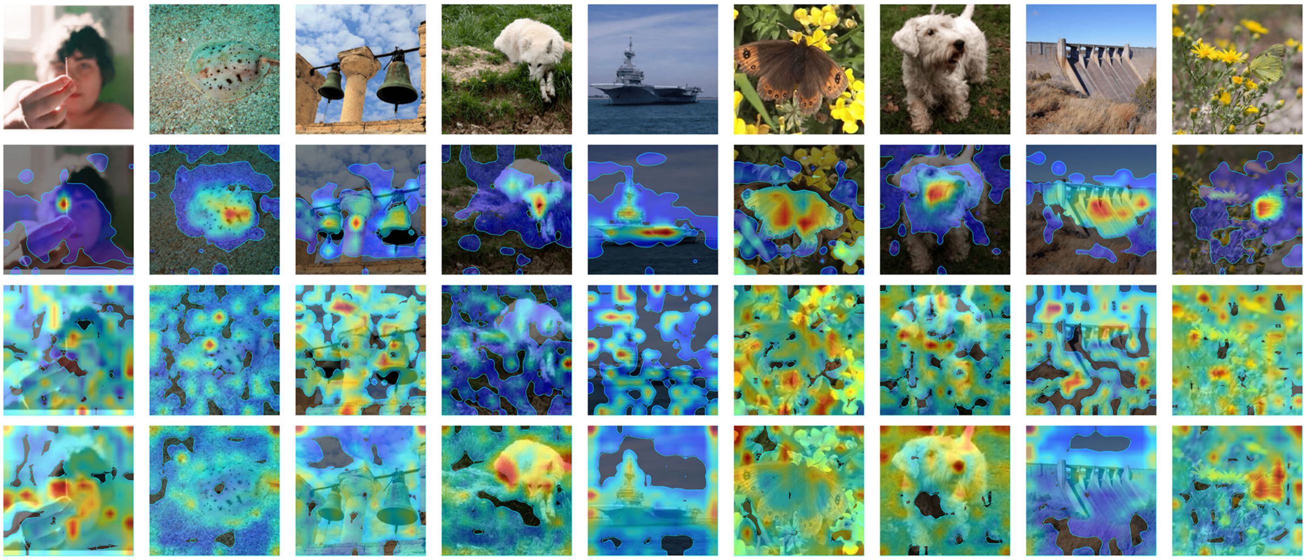

Fig. 6 Illustrating the attention map averaged over 12 attention heads between the class token and the patch tokens in the last layer of the ViT encoder pretrained on ImageNet-1K. The region inside the blue contour is obtained by thresholding the attention weights to keep 50% of the mass. The four rows are: (1) input image, (2) MoCo v3, a typical contrastive self-supervised learning method, (3) MAE, and (4) our CAE. One can see that MoCo v3 tends to focus mainly on the centering regions and little on other patches, and our CAE tends to consider almost all the patches

Comparison to contrastive self-supervised learning Typical contrastive self-supervised learning methods, e.g., SimCLR (Chen et al., 2020) and MoCo (He et al., 2020; Chen et al., 2021), pretrain the networks by solving the pretext task, maximizing the similarities between augmented views (e.g., random crops) from the same image and minimizing the similarities between augmented views from different images.

It is shown in Chen et al. (2020) that random cropping plays an important role in view augmentation for contrastive self-supervised learning. Through analyzing random crops (illustrated in Fig. 3), we observe that the center pixels in the original image space have large chances to belong to random crops. We suspect that the global representation, learned by contrastive self-supervised learning for a random crop possibly with other augmentation schemes, tends to focus mainly on the center pixels in the original image, so that the representations of different crops from the same image can be possibly similar. Figure 6 (the second row) shows that the center region of the original image for the typical contrastive self-supervised learning approach, MoCo v3, is highly attended. The part in random crops corresponding to the center of the original image is still attended as shown in Fig. 7.

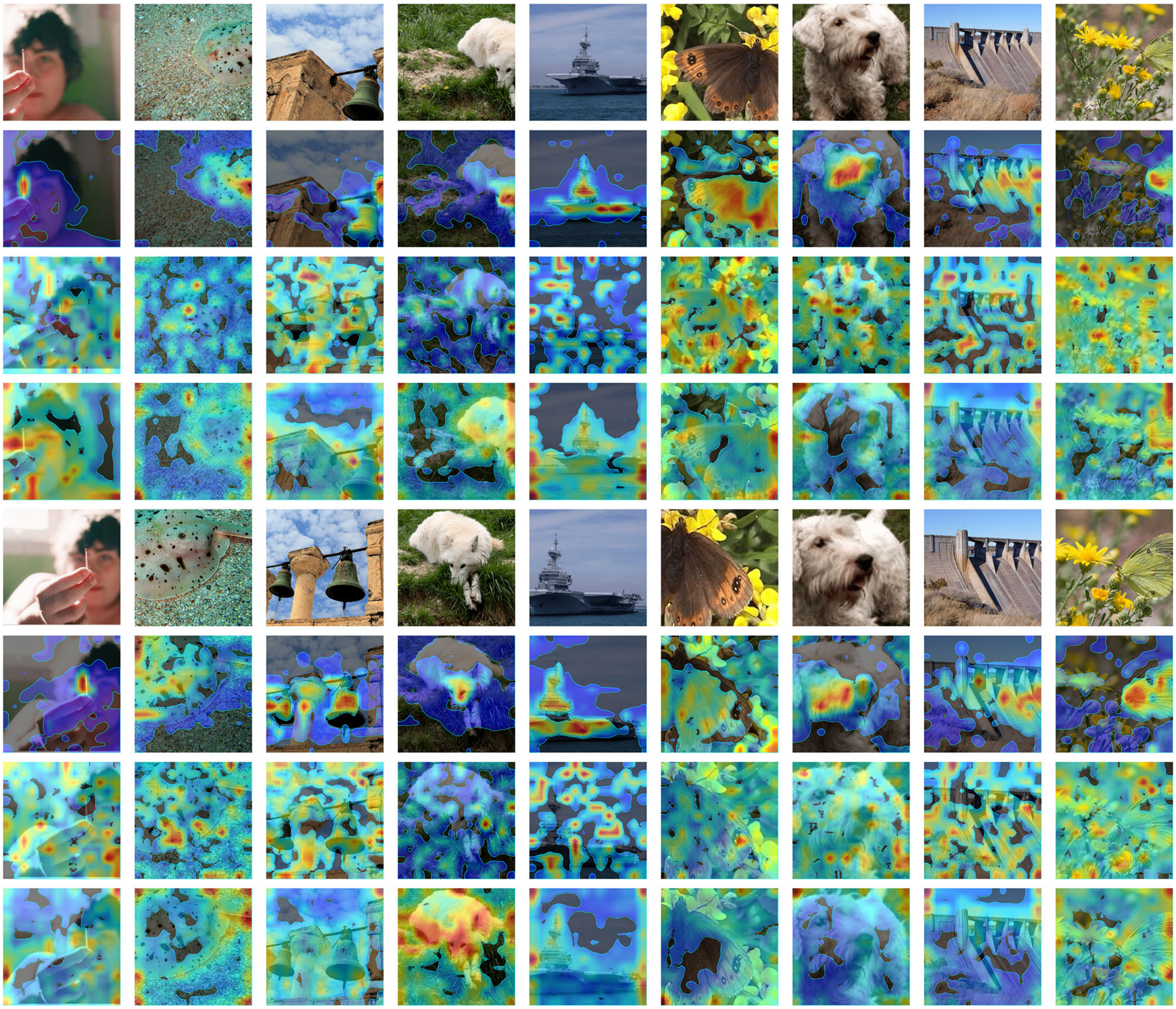

Fig. 7 The attention maps over two sets of randomly cropped images (the 1st the 5th rows) for MoCo v3 (the 2nd the 6th rows), MAE (the 3rd the 7th rows), and our CAE (the 4th the 8th rows) pretrained on ImageNet-1K. The contrastive self-supervised learning method, MoCo v3, tends to focus mainly on the object region and little on other regions. In contrast, MIM-based models, CAE and MAE, tend to consider almost all the patches. The attention maps over the original images are shown in Fig. 6

In contrast, our CAE method (and other MIM methods) randomly samples the patches from the augmented views to form the visible and masked patches. All the patches are possible to be randomly masked for the augmented views and accordingly the original image. Thus, the CAE encoder needs to learn good representations for all the patches, to make good predictions for the masked patches from the visible patches. Figure 6 (the third row) illustrates that almost all the patches in the original images are considered in our CAE encoder.

Considering that the instances of the 1000 categories in ImageNet-1K locate mainly around the center of the original images (Russakovsky et al., 2015), typical contrastive self-supervised learning methods, e.g., MoCo v3, learn the knowledge mainly about the 1000 categories, which is similar to supervised pretraining. But our CAE and other MIM methods are able to learn more knowledge beyond the 1000 categories from the non-center image regions. This indicates that the CAE has the potential to perform better for downstream tasks.

4.3 Interpretation

Intuitive Interpretation for CAE. Humans are able to hallucinate what appears in the masked regions and how they appear according to the visible regions. We speculate that humans do this possibly in a way similar as the following

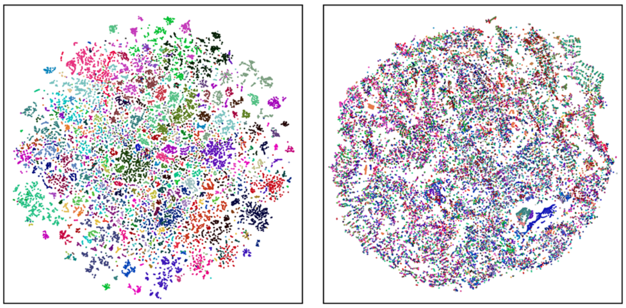

Fig. 8 t-SNE visualization (one color for one category) of representations extracted from the images in ADE20K. Left: ViT pretrained with our CAE; Right: ViT with random weights

example: given that only the region of the dog's head is visible and the remaining parts are missing, one can (a) recognize the visible region to be about a dog, (b) predict the regions where the other parts of the dog appear, and (c) guess what the other parts look like.

Our CAE encoder is in some sense like the human recognition step (a). It understands the content by mapping the visual patches into latent representations that lie in the subspace that corresponds to the category dog.[2] The latent contextual regressor is like step (b). It produces a plausible hypothesis for the masked patches, and describes the regions corresponding to the other parts of the dog using latent representations. The CAE decoder is like step (c), mapping the latent representations to the targets. It should be noted that the latent representations might contain other information besides the semantic information, e.g., the part information and the information for making predictions.

We adopt t-SNE (Van der Maaten & Hinton, 2008) to visualize the high-dimensional patch representations output from our CAE encoder on ADE20K (Zhou et al., 2017) in Fig. 8. ADE20K has a total of 150 categories. For each patch in the image, we set its label to be the category that more than half of the pixels belong to. We collect up to 1000 patches for each category from sampled 500 images. As shown in the figure, the latent representations of CAE are clustered to some degree for different categories (though not perfect as our CAE is pretrained on ImageNet-1K). Similar observations could be found for other MIM methods.

Probabilistic interpretation for CAE The MIM problem can be formulated in the probabilistic form, maximizing the probability of the predictions $\mathbf{Y}_m$ of the masked patches given the conditions, the visible patches $\mathbf{X}_v$, the positions $\mathbf{P}_v$ of the visible patches, and the positions $\mathbf{P}_m$ of the masked patches: $P(\mathbf{Y}_m \mid \mathbf{X}_v, \mathbf{P}_v, \mathbf{P}_m)$. It can be solved by introducing latent representations $\mathbf{Z}_m$ and $\mathbf{Z}_v$, with the assumption that $\mathbf{Z}_v$ and $\mathbf{P}_m$ ($\mathbf{Y}_m$ and $\mathbf{P}_v$) are conditionally independent (the probabilistic graphical model is given in Fig. 9):

$$\begin{aligned} & p(\mathbf{Y}_m \mid \mathbf{X}_v, \mathbf{P}_v, \mathbf{P}_m) && (2)\\ =& p(\mathbf{Z}_v \mid \mathbf{X}_v, \mathbf{P}_v, \mathbf{P}_m)\, p(\mathbf{Z}_m \mid \mathbf{Z}_v, \mathbf{P}_v, \mathbf{P}_m) && \\ & p(\mathbf{Y}_m \mid \mathbf{Z}_m, \mathbf{P}_v, \mathbf{P}_m) && (3)\\ =& p(\mathbf{Z}_v \mid \mathbf{X}_v, \mathbf{P}_v)\, p(\mathbf{Z}_m \mid \mathbf{Z}_v, \mathbf{P}_v, \mathbf{P}_m)\, p(\mathbf{Y}_m \mid \mathbf{Z}_m, \mathbf{P}_m). && (4) \end{aligned}$$

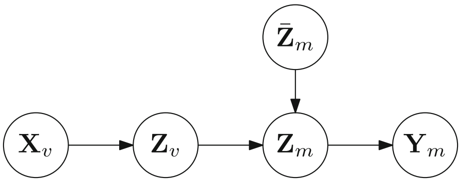

Fig. 9 The probabilistic graphical model of CAE. The other conditions of $\mathbf{Z}_v$, $\mathbf{Z}_m$, and $\mathbf{Y}_m$, the positions $\mathbf{P}_v$ and $\mathbf{P}_m$ of the visible and masked patches, are not plotted for simplicity

Here, the equation from (2) to (3) is obtained from the probabilistic graphical model of CAE shown in Fig. 9, and the removal of the condition $\mathbf{P}_m$ (from $p(\mathbf{Z}_v \mid \mathbf{X}_v, \mathbf{P}_v, \mathbf{P}_m)$ to $p(\mathbf{Z}_v \mid \mathbf{X}_v, \mathbf{P}_v)$), and the condition $\mathbf{P}_v$ (from $p(\mathbf{Y}_m \mid \mathbf{Z}_m, \mathbf{P}_v, \mathbf{P}_m)$ to $p(\mathbf{Y}_m \mid \mathbf{Z}_m, \mathbf{P}_m)$) from (3) to (4) is based on the conditional independence assumption. The three terms in (4) correspond to three parts of our CAE: the encoder, the latent contextual regressor, and the decoder, respectively.

Similarly, the latent representation alignment constraint can be written as a conditional probability, $P(\mathbf{Z}_m \mid \bar{\mathbf{Z}}_m)$, where $\bar{\mathbf{Z}}_m$ is the masked patch representations computed from the encoder.

Intuitive interpretation for the contrastive self-supervised learning We consider the case in ImageNet-1K that the object mainly lies in the center of an image.[3] There are N randomly sampled crops from an image, and each crop $\mathbf{I}_n$ contains a part of the center object, $\mathbf{O}_n$. To maximize the similarity between two crops $\mathbf{I}_m$ and $\mathbf{I}_n$, the pretraining might contain the processes: select the regions $\mathbf{O}_m$ and $\mathbf{O}_n$ from the two crops $\mathbf{I}_m$ and $\mathbf{I}_n$, extract their features $\mathbf{f}_{om}$ and $\mathbf{f}_{on}$, and predict the feature of the object, $\mathbf{f}_o$, from the part features $\mathbf{f}_{om}$ and $\mathbf{f}_{on}$. In this way, the features of the crops from the same image could be similar. Among the N random crops, most crops contain a part of the object in the center, and a few crops that do not contain a part of the center object could be viewed as noises when optimizing the contrastive loss.

After pretrained on ImageNet-1K (where the object mainly lies in the center) the encoder is able to learn the knowledge of the 1000 classes and localize the region containing the object belonging to the 1000 classes. It is not necessary that the object lies in the center for the testing

[2] Our encoder does not know that the subspace is about a dog, and just separates it from the subspaces of other categories.

[3] There are a few images in which the object does not lie in the center in ImageNet-1K. The images are actually viewed as noises and have little influence for contrastive self-supervised learning.

image, which is verified in Fig. 7. This further verifies that MoCo v3 (contrastive self-supervised pretraining) pretrained on ImageNet-1K tends to attend to the object region, corresponding to the center region of the original image as shown in Fig. 6.

5 Experiments

5.1 Implementation

We study the standard ViT small, base and large architectures, ViT-S (12 transformer blocks with dimension 384), ViT-B (12 transformer blocks with dimension 768) and ViT-L (24 transformer blocks with dimension 1024). The latent contextual regressor consists of 4 transformer blocks based on cross-attention in which self-attention over masked tokens and encoded visible patch representations is a choice but with slightly higher computation cost and a little lower performance, and the decoder consists of 4 transformer blocks based on self-attention, and an extra linear projection for making predictions.

5.2 Training Details

Pretraining The pretraining settings are almost the same as BEiT (Bao et al., 2021). We train the CAE on ImageNet-1K. We partition the image of 224 × 224 into 14 × 14 patches with the patch size being 16 × 16. We use standard random cropping and horizontal flipping for data augmentation. We use AdamW (Loshchilov & Hutter, 2017) for optimization and train the CAE for 300/800/1600 epochs with the batch size being 2048. We set the learning rate as 1.5e-3 with cosine learning rate decay. The weight decay is set as 0.05. The warmup epochs for 300/800/1600 epochs pre-training are 10/20/40, respectively. We employ drop path (Huang et al., 2016) rate 0.1 and dropout rate 0.

Linear probing We use the LARS (You et al., 2017) optimizer with momentum 0.9. The model is trained for 90 epochs. The batch size is 16384, the warmup epoch is 10 and the learning rate is 6.4. Following He et al. (2022), we adopt an extra BatchNorm layer (Ioffe & Szegedy, 2015) without affine transformation (`affine=False`) before the linear classifier. We do not use mixup (Zhang et al., 2017), cutmix (Yun et al., 2019), drop path (Huang et al., 2016), or color jittering, and we set weight decay as zero.

Attentive probing The parameters of the encoder are fixed during attentive probing. A cross-attention module, a BatchNorm layer (`affine=False`), and a linear classifier are appended after the encoder. The extra class token representation in cross-attention is learned as model parameters. The keys and the values are the patch representations output from the encoder. There is no MLP or skip connection operation in the extra cross-attention module. We use the SGD optimizer with momentum 0.9 and train the model for 90 epochs. The batch size is 8192, the warmup epoch is 10 and the learning rate is 0.4. Same as linear probing, we do not use mixup (Zhang et al., 2017), cutmix (Yun et al., 2019), drop path, or color jittering, and we set weight decay as zero.

Fine-tuning on ImageNet We follow the fine-tuning protocol in BEiT to use layer-wise learning rate decay, weight decay and AdamW. The batch size is 4096, the warmup epoch is 5 and the weight decay is 0.05. For ViT-S, we train 200 epochs with learning rate 1.6e-2 and layer-wise decay rate 0.75. For ViT-B, we train 100 epochs with learning rate 8e-3 and layer-wise decay rate 0.65. For ViT-L, we train 50 epochs with learning rate 2e-3 and layer-wise decay rate 0.75.

Semantic segmentation on ADE 20K We use AdamW as the optimizer. The input resolution is 512 × 512. The batch size is 16. For the ViT-B, the layer-wise decay rate is 0.65 and the drop path rate is 0.1. We search from four learning rates, 1e–4, 2e–4, 3e–4 and 4e–4, for all the results in Table 2. For the ViT-L, the layer-wise decay rate is 0.95 and the drop path rate is 0.15. We search from three learning rates for all the methods, 3e–5, 4e–5, and 5e–5, We conduct fine-tuning for 160K steps. We do not use multi-scale testing.

Object detection and instance segmentation on COCO We utilize multi-scale training and resize the image with the size of the short side between 480 and 800 and the long side no larger than 1333. The batch size is 32. For the ViT-S, the learning rate is 3e–4, the layer-wise decay rate is 0.75, and the drop path rate is 0.1. For the ViT-B, the learning rate is 3e–4, the layer-wise decay rate is 0.75, and the drop path rate is 0.2. For the ViT-L, the learning rate is 2e–4, the layer-wise decay rate is 0.8, and the drop path rate is 0.2. We train the network with the 1× schedule: 12 epochs with the learning rate decayed by 10× at epochs 9 and 11. We do not use multi-scale testing. The Mask R-CNN implementation follows MMDetection Chen et al. (2019).

5.3 Pretraining Evaluation

Linear probing Linear probing is widely used as a proxy of pretraining quality evaluation for self-supervised representation learning. It learns a linear classifier over the image-level representation output from the pretrained encoder by using the labels of the images, and then tests the performance on the validation set.

Attentive probing The output of the encoder pretrained with MIM methods are representations for all the patches. It is not suitable to linearly probe the representation, averagely-pooled from patch representations, because the image label in ImageNet-1K only corresponds to a portion of patches. It is also not suitable to use the default class token within the encoder because the default class token serves as a role of aggregating the patch representations for better patch rep-

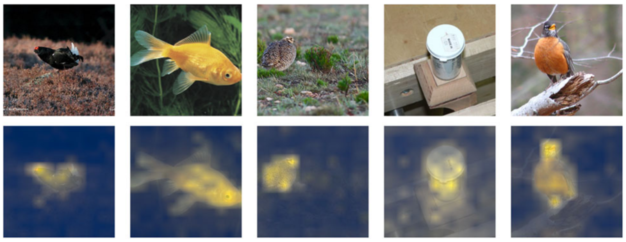

Fig. 10 Illustrating the cross-attention unit in attentive probing. The attention map (bottom) is the average of cross-attention maps over 12 heads between the extra class token and the patches. One can see that the attended region lies mainly in the object, which helps image classification

Table 1 Pretraining quality evaluation in terms of fine-tuning (FT), linear probing (LIN), and attentive probing (ATT)

Method	#Epochs	#Crops	FT	LIN	ATT
Methods using ViT-S:					
DeiT	300	–	–	–	79.9
MoCo v3	600‡	2	81.7	73.1	73.8
BEiT	300	1	81.7	15.7	23.6
CAE*	300	1	**82.0**	51.8	65.0
Methods using ViT-B:					
DeiT	300	–	–	–	81.8
MoCo v3	600‡	2	83.0	76.2	77.0
DINO	1600‡	12	83.3	77.3	77.8
BEiT	300	1	83.0	37.6	49.4
MAE	300	1	82.9	61.5	71.1
MAE	1600	1	83.6	67.8	74.2
SimMIM	800	1	83.8	56.7	–
iBOT	1600‡	12	83.8	79.5	79.8
CAE*	300	1	83.6	64.1	73.8
CAE*	800	1	83.8	68.6	75.9
CAE*	1600	1	**83.9**	70.4	77.1
CAE	1600	1	**83.9**	71.4	77.4
Methods using ViT-L:					
MoCo v3†	600‡	2	84.1	–	–
BEiT†	1600	1	85.2	–	–
MAE	1600	1	86.0	76.0	78.8
CAE*	1600	1	**86.3**	78.1	81.2
CAE	1600	1	**86.3**	77.9	81.2

‡ means the number of effective epochs in Zhou et al. (2021) as they adopt multi-crop augmentation (equivalently take a larger number of epochs compared to one-crop augmentation). We report the top-1 accuracy (in the column ATT) of the supervised training approach DeiT (Touvron et al., 2020) to show how far the ATT score is from supervised training. The scores for other models and our models are based on our implementations if not specified. Except that * denotes using the DALL-E tokenizer, CAE adopts the d-VAE tokenizer trained on ImageNet-1K only

resentation extraction and is not merely for the portion of patches corresponding to the image label.

To use the image-level label as a proxy of evaluating the pretraining quality for the encoder pretrained with MIM methods, we need to attend the patches that are related to the label. We introduce a simple modification by using a cross-attention unit with an extra class token (that is different from the class token in the encoder) as the query and the output patch representations of the encoder as the keys and the values, followed by a linear classifier. The introduced cross-attention unit is able to care mainly about the patches belonging to the 1000 classes in ImageNet-1K and remove the interference of other patches. Figure 10 illustrates the effect of the cross-attention unit, showing that the extra cross-attention unit can to some degree attend the regions that are related to the 1000 ImageNet-1K classes.

Results Table 1 shows the results with three schemes, linear probing (LIN), attentive probing (ATT), and fine-tuning (FT) for representative contrastive self-supervised pretraining (MoCo v3 and DINO) and MIM (BEiT and MAE) methods, as well as our approach with the targets formed with the DALL-E tokenizer (trained on 400M images) and the d-VAE tokenizer (trained on ImageNet-1K without using the labels), denoted as CAE* and CAE, respectively. The models of MAE with 300 epochs and BEiT are pretrained by us using the official implementations, and other models are officially released models.

We highlight a few observations. The fine-tuning performance for these methods are very similar and there is only a minor difference similar to the observation (Zhou et al., 2021). We think that the reason is that self-supervised pretraining and fine-tuning are conducted on the same dataset and no extra knowledge is introduced for image classification. The minor difference might come from the optimization aspect: different initialization (provided by pretrained models) for fine-tuning.

In terms of linear probing, the scores of the contrastive self-supervised learning methods, MoCo v3 and DINO, are higher than the MIM methods. This is as expected because contrastive self-supervised learning focuses mainly on learning the representations for 1000 classes (See discussion in Sect. 4). The pretraining is relatively easier than existing MIM methods as contrastive self-supervised learning mainly cares about the 1000 classes and MIM methods may care about the classes beyond the 1000 classes.

For the MIM methods, the scores of attentive probing are much larger than linear probing. This validates our analysis: the MIM methods extract the representations for all the patches, and the classification task needs to attend the corresponding portion of patches.

The LIN and ATT scores are similar for contrastive self-supervised pretraining on ViT-B, e.g., (76.2 vs. 77.0) for MoCo v3 and (77.3 vs. 77.8) for DINO. This means that the extra cross-attention in attentive probing does not make a big difference, which is one more evidence for our analysis in Sect. 4 that they already focus mainly on the region where the instance in the 1000 categories lies.

5.4 Downstream Tasks

Semantic segmentation on ADE20K (Zhou et al., 2017) We follow the implementation (Bao et al., 2021) to use UperNet (Xiao et al., 2018). The CAE with the tokenizers learned over ImageNet-1K performs almost the same as the tokenizers learned over 400M images provided by DALL-E (CAE*), implying that the tokenizer trained on ImageNet-1K (without using the labels) or a larger dataset does not affect the pretraining quality and accordingly the downstream task performance.

Table 2 shows that using the ViT-B, our CAE* with 300 training epochs performs better than DeiT, MoCo v3, DINO, MAE (1600 epochs) and BEiT. Our CAE* (1600 epochs) further improves the segmentation scores and outperforms MAE (1600 epochs), MoCo v3 and DeiT by 2.1, 3.0 and 3.2, respectively. Using ViT-L, our CAE* (1600 epochs) outperforms BEiT (1600 epochs) and MAE (1600 epochs) by 1.4 and 1.1, respectively.

The superior results over supervised and contrastive self-supervised pretraining methods, DeiT, MoCo v3 and DINO, stem from that our approach captures the knowledge beyond the 1000 classes in ImageNet-1K. The superior results over BEiT and MAE stems from that our CAE makes predictions in the encoded representation space and that representation learning and pretext task completion are separated.

Object detection and instance segmentation on COCO (Lin et al., 2014) We adopt the Mask R-CNN approach (He et al., 2017) that produces bounding boxes and instance masks simultaneously, with the ViT as the backbone. The results are given in Table 3. We report the box AP for object detection and the mask AP for instance segmentation. The observations are consistent with those for semantic segmentation in Table 2. Our CAE* (300 epochs, ViT-B) is superior to all the other models except that a little lower than MAE (1600 epochs). Our approach (1600 epochs) outperforms MAE (1600 epochs), MoCo v3 and DeiT by 1.6, 4.5 and 3.1, respectively. Using ViT-L, our CAE achieves 54.6 box AP and outperforms MAE by 0.6.

We also report the results of object detection and instance segmentation on COCO with the Cascaded Mask R-CNN framework (Cai & Vasconcelos, 2021) in Table 6. Results show that our CAE performs better than other methods.

In addition, we conduct experiments on the scaling ability of CAE on the detection task. The detection model is built upon ViT-Huge (Dosovitskiy et al., 2021), DINO (Zhang et al., 2023), and Group DETR (Chen et al., 2022) (see (Chen et al., 2022) for more details). The ViT-Huge is pretrained on ImageNet-22K (Deng et al., 2009) using CAE. We are the first to obtain 64.6 mAP on COCO *test-dev*, which outperforms previous methods with larger models and more training data (e.g., BEIT-3 (Wang et al., 2023) (63.7 mAP) and SwinV2-G (Liu et al., 2021) (63.1 mAP)).

Table 2 Semantic segmentation on ADE20K

Method	#Epochs	mIoU
Methods using ViT-B:		
SplitMask	–	45.7
BEiT	300	45.5
BEiT	800	46.5
mc-BEiT	800	47.0
DeiT	300	47.0
MoCo v3	600‡	47.2
DINO	1600‡	47.2
MAE	300	45.8
MAE	1600	48.1
Ge^2-AE	800	48.9
A^{2}MIM	800	49.0
iBOT	1600‡	50.0
CAE*	300	48.3
CAE*	800	49.7
CAE*	1600	**50.2**
CAE	1600	50.1
Methods using ViT-L:		
MoCo v3†	600‡	49.1
BEiT†	1600	53.3
MAE	1600	53.6
CAE*	1600	**54.7**
CAE	1600	54.6

All the results are based on the same implementation for semantic segmentation. #Epochs refers to the number of pretraining epochs. ‡ means the number of effective epochs in Zhou et al. (2021) as the method uses multi-crop pretraining augmentation (See Table 1). SplitMask (El-Nouby et al., 2021) is pretrained on ADE20K for 21000 epochs. †: these results are from He et al. (2022)

Classification We conduct fine-tuning experiments on three datasets: Food-101 (Bossard et al., 2014), Clipart (Castrejon et al., 2016), and Sketch (Castrejon et al., 2016). Results in Table 4 show that the proposed method outperforms the previous supervised method (DeiT) and self-supervised methods (DINO, MAE).

5.5 Ablation Studies

Decoder and alignment The CAE architecture contains several components for pretraining the encoder: regressor and alignment for masked representation prediction, decoder with a linear layer for masked patch reconstruction. We observe that if the pretraining task, masked patch recon-

Table 3 Object detection and instance segmentation on COCO. Mask R-CNN is adopted and trained with the 1× schedule

Method	#Epochs	Supervised	Self-supervised	Object detection			Instance segmentation		
				AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
Methods using ViT-S									
DeiT	300	√	×	43.1	65.2	46.6	38.4	61.8	40.6
MoCo v3	600‡	×	√	43.3	64.9	46.8	38.8	61.6	41.1
BEiT	300	×	√	35.6	56.7	38.3	32.6	53.3	34.2
CAE*	300	×	√	**44.1**	64.6	48.2	**39.2**	61.4	42.2
Methods using ViT-B:									
DeiT	300	√	×	46.9	68.9	51.0	41.5	65.5	44.4
MoCo v3	600‡	×	√	45.5	67.1	49.4	40.5	63.7	43.4
DINO	1600‡	×	√	46.8	68.6	50.9	41.5	65.3	44.5
BEiT	300	×	√	39.5	60.6	43.0	35.9	57.7	38.5
BEiT	800	×	√	42.1	63.3	46.0	37.8	60.1	40.6
MAE	300	×	√	45.4	66.4	49.6	40.6	63.4	43.7
MAE	1600	×	√	48.4	69.4	53.1	42.6	66.1	45.9
iBOT	1600‡	×	√	48.2	69.7	52.8	42.7	66.5	46.0
CAE*	300	×	√	48.4	69.2	52.9	42.6	66.1	45.8
CAE*	800	×	√	49.8	70.7	54.6	43.9	67.8	47.4
CAE*	1600	×	√	50.0	70.9	54.8	44.0	67.9	47.6
CAE	1600	×	√	**50.2**	71.0	54.9	**44.2**	68.3	47.9
Methods using ViT-L									
MAE	1600	×	√	54.0	74.3	59.5	47.1	71.5	51.0
CAE*	1600	×	√	54.5	75.2	60.1	47.6	72.2	51.9
CAE	1600	×	√	**54.6**	75.2	59.9	**47.6**	72.0	51.9

All the results are based on the same implementation for object detection and instance segmentation. #Epochs refers to the number of pretraining epochs on ImageNet-1K. ‡ means the number of effective epochs in Zhou et al. (2021) (See Table 1)

Table 4 Top-1 classification accuracy on the Food-101, Clipart and Sketch datasets

Method	Supervised	Self-supervised	Food-101	Clipart	Sketch
Random Init	×	×	82.77	52.90	46.42
DeiT	√	×	91.81	81.18	73.45
DINO	×	√	91.67	80.72	73.13
MAE	×	√	93.19	80.63	73.87
CAE*	×	√	**93.32**	**81.84**	**74.65**

The backbone is ViT-B

struction, is not included, the training collapses, leading to a trivial solution. We thus study the effect of the decoder (when the decoder is removed, we use a linear layer to predict the targets), which is helpful for target reconstruction, and the alignment, which is helpful for representation prediction.

Table 5 shows the ablation results. We report the scores for linear probing, attentive probing, fine-tuning and downstream tasks: semantic segmentation on the ADE20K dataset and object detection on COCO with the DALL-E tokenizer as the target. One can see that the downstream task performance is almost the same when only the decoder is added and that the performance increases when the decoder and the alignment are both added. This also verifies that the alignment is important for ensuring that the predicted representations of masked patches lie in the encoded representation space and thus the predictions are made in the encoded representation space, and accordingly improving the representation quality. Without the decoder, the performance drops. This is because the reconstruction from the semantic representation to the low-level targets cannot be done through a single linear layer, and no decoder will deteriorate the semantic quality of the encoder. The additional computational cost, i.e. the number of parameters and training time, brought by the decoder and alignment is relatively small, e.g., increasing the number of parameters to 1.23× and training time to 1.24×.

Table 5 Ablation studies for the decoder and the alignment constraint in our CAE

Decoder	Alignment	LIN	ATT	FT	ADE Seg	COCO Det	#Params	Training Time
×	×	60.3	71.2	82.9	47.0	46.9	120.32 M	1×
√	×	63.1	72.7	83.4	47.1	47.2	148.68 M	1.14×
×	√	62.0	71.5	83.4	47.1	47.2	120.32 M	1.12×
√	√	64.1	73.8	83.6	48.3	48.4	148.68 M	1.24×

All the models are pretrained on ImageNet-1K with 300 epochs

Table 6 The results of object detection and instance segmentation on COCO with the Cascaded Mask-RCNN framework (1× schedule)

Method	#Epochs	AP^b	AP^m
MAE He et al. (2022)	1600	51.3	44.3
mc-BEiT Zhou et al. (2021)	800	50.1	43.1
iBOT Zhou et al. (2021)	1600	51.2	44.2
CAE*	300	51.6	44.6
CAE*	800	52.8	45.5
CAE*	1600	**52.9**	**45.5**

ViT-B is used for all experiments. All the detection results are from our implementation

Table 7 The effect of mask ratios

Mask Ratio	LIN	ATT	ADE Seg
40%	63.1	73.0	47.2
50%	64.1	73.8	48.3
60%	64.8	74.2	48.1

The backbone is ViT-B. Models are trained for 300 epochs

Mask ratio We also conduct experiments with different mask ratios including 40%, 50%, and 60%. Results are listed in Table 7. We find that ratio 50% gets better results than ratio 40%. Adopting a higher mask ratio (60%) could further improve the performance of linear probing and attentive probing, while the semantic segmentation performance is reduced by 0.2%. We choose 50% in our work unless specified.

#layers in the regressor and decoder For the number of layers in the latent contextual regressor and decoder, we tried four choices: 1-layer, 2-layers, 4-layer, and 5-layer. The results for linear probing are 58.7, 62.1, 64.1, and 64.2. The results for attentive probing are 67.5, 71.1, 73.8, and 73.7. We empirically observed that 4-layer outperforms other choices overall.

Loss tradeoff parameter There is a tradeoff variable λ in the loss function given in Eq. 1. We did not do an extensive study and only tried three choices, $\lambda = 1$, $\lambda = 1.5$ and $\lambda = 2$. The linear probing results are 63.4, 63.7 and 64.1, respectively. The choice $\lambda = 1$ works also well, slightly worse than $\lambda = 2$ that is adopted in our experiment.

Table 8 The effect of reconstruction targets on the performance of CAE

Targets	LIN	ATT	ADE Seg
DALL-E tokenizer	70.4	77.1	50.2
d-VAE tokenizer	71.4	77.4	50.1
RGB pixel value	72.4	77.0	50.4

The backbone is ViT-B. Models are trained for 1600 epochs

Reconstruction targets To study the impact of different pretraining targets on model performance, we conduct additional experiments on the RGB pixel value target. Comparing the results with DALL-E tokenizer and d-VAE tokenizer trained on ImageNet-1K, the model shows better linear probe and segmentation results but inferior in attentive probe, as shown in Table 8. Pretraining with these three targets obtains similar performance, illustrating that CAE does not rely on specific pretraining targets.

6 Conclusion

The core design of our CAE architecture for masked image modeling is that predictions are made from visible patches to masked patches in the encoded representation space. We adopt two pretraining tasks: masked representation prediction and masked patch reconstruction. Experiments demonstrate the effectiveness of the CAE design. In addition, we also point out that the advantage of MIM methods over typical contrastive self-supervised pretraining and supervised pretraining on ImageNet-1K is that MIM learns the representations for all the patches, while typical contrastive self-supervised pretraining (e.g., MoCo and SimCLR) and supervised pretraining tend to learn semantics mainly from center patches of the original images and little from non-center patches.

Possible extensions, as mentioned in the arXiv version (Chen et al., 2022), include: investigating the possibility only considering the pretraining task, masked representation prediction, without masked patch reconstruction, pretraining a depth-wise convolution network with masked convolution, and pretraining with the CLIP targets (Zhang et al., 2022b).

Potential limitations The proposed method may face challenges when dealing with large and contiguous masked regions in an image, e.g., the whole object region is almost masked. Obtaining plausible and high-quality reconstruction for large areas can be particularly difficult, as the model has to infer the missing information based on limited available context. This is a common limitation of Masked Image Modeling methods, and our proposed method is not exempt from it.

Acknowledgements We would like to acknowledge Hangbo Bao, Xinlei Chen, Li Dong, Qi Han, Zhuowen Tu, Saining Xie, and Furu Wei for the helpful discussions.

Funding This work is partially supported by the National Key Research and Development Program of China (2020YFB1708002), National Natural Science Foundation of China (61632003, 61375022, 61403005), Grant SCITLAB-20017 of Intelligent Terminal Key Laboratory of SiChuan Province, Beijing Advanced Innovation Center for Intelligent Robots and Systems (2018IRS11), and PEK-SenseTime Joint Laboratory of Machine Vision. Ping Luo is supported by the General Research Fund of HK No.27208720, No.17212120, and No.17200622.

Data availability The datasets used in this paper are publicly available. ImageNet: https://www.image-net.org/, ADE20K: https://groups.csail.mit.edu/vision/datasets/ADE20K/, COCO: https://cocodataset.org/, Food-101: https://data.vision.ee.ethz.ch/cvl/datasets_extra/food-101/, Clipart: http://projects.csail.mit.edu/cmplaces/download.html, Sketch: http://projects.csail.mit.edu/cmplaces/download.html.

Code Availability Our code will be available at https://github.com/Atten4Vis/CAE.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

References

Asano, Y. M., Rupprecht, C., & Vedaldi, A. (2019). Self-labelling via simultaneous clustering and representation learning. arXiv:1911.05371

Atito, S., Awais, M., & Kittler, J. (2021). Sit: Self-supervised vision transformer. arXiv:2104.03602

Baevski, A., Hsu, W.-N., Xu, Q., Babu, A., Gu, J., & Auli, M. (2022). *data2vec: A general framework for self-supervised learning in Speech*. Technical report: Vision and Languags.

Bao, H., Dong, L., & Wei, F. (2021). BEiT: BERT pre-training of image transformers. arXiv:2106.08254

Bardes, A., Ponce, J., & LeCun, Y. (2021). Vicreg: Variance-invariance-covariance regularization for self-supervised learning. arXiv:2105.04906

Bossard, L., Guillaumin, M., & Van Gool, L. (2014). Food-101—Mining discriminative components with random forests. In *ECCV*.

Cai, Z., & Vasconcelos, N. (2021). Cascade r-cnn: High quality object detection and instance segmentation. *TPAMI, 43*, 1483–1498.

Caron, M., Bojanowski, P., Joulin, A., & Douze, M. (2018). Deep clustering for unsupervised learning of visual features. In *ECCV* (pp. 132–149).

Caron, M., Bojanowski, P., Mairal, J., & Joulin, A. (2019). Unsupervised pre-training of image features on non-curated data. In *ICCV* (pp. 2959–2968).

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., & Joulin, A. (2020). Unsupervised learning of visual features by contrasting cluster assignments. arXiv:2006.09882.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A. (2021). Emerging properties in self-supervised vision transformers. CoRR, arxiv:2104.14294.

Castrejon, L., Aytar, Y., Vondrick, C., Pirsiavash, H., & Torralba, A. (2016). Learning aligned cross-modal representations from weakly aligned data. In *CVPR* (pp. 2940–2949).

Chen, X., & He, K. (2021). Exploring simple Siamese representation learning. In *CVPR* (pp. 15750–15758).

Chen, Q., Chen, X., Wang, J., Feng, H., Han, J., Ding, E., Zeng, G., & Wamg, J. (2022). Group detr: Fast detr training with group-wise one-to-many assignment.

Chen, X., Ding, M., Wang, X., Xin, Y., Mo, S., Wang, Y., Han, S., Luo, P., Zeng, G., & Wang, J. (2022). Context autoencoder for self-supervised representation learning. CoRR, arxiv:2202.03026.

Chen, J., Hu, M., Li, B., & Elhoseiny, M. (2022). Efficient self-supervised vision pretraining with local masked reconstruction. arXiv:2206.00790

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. E. (2020). A simple framework for contrastive learning of visual representations. In *ICML, volume 119 of Proceedings of Machine Learning Research* (pp. 1597–1607). PMLR.

Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., & Sutskever, I. (2020). Generative pretraining from pixels. In *ICML* (pp. 1691–1703). PMLR.

Chen, Q., Wang, J., Han, C., Zhang, S., Li, Z., Chen, X., Chen, J., Wang, X., Han, S., Zhang, G., Feng, H., Yao, K., Han, J., Ding, E., & Wang, J. (2022). Group DETR v2: Strong object detector with encoder-decoder pretraining. CoRR, arxiv:2211.03594.

Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Xu, J., Zhang, Z., Cheng, D., Zhu, C., Cheng, T., Zhao, Q., Li, B., Lu, X., Zhu, R., Wu, Y., Dai, J., Wang, J., Shi, J., Ouyang, W., Loy, C. C., & Lin, D. (2019). MMDetection: Open mmlab detection toolbox and benchmark. arXiv:1906.07155.

Chen, X., Xie, S., & Kaiming, H. (2021). An empirical study of training self-supervised vision transformers. CoRR, arxiv:2104.02057.

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

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, & T. Solorio (Eds.), *NAACL-HLT* (pp. 4171–4186). Association for Computational Linguistics.

Doersch, C., Gupta, A., & Efros, Alexei, A. (2015). Unsupervised visual representation learning by context prediction. In *ICCV*.

Dong, X., Bao, J., Zhang, T., Chen, D., Zhang, Weiming,Y., Lu, C., Dong, W., Fang, & Yu, N. (2021). Peco: Perceptual codebook for bert pre-training of vision transformers. arXiv:2111.12710.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). *An image is worth 16x16 words: Transformers for image recognition at scale*. In ICLR: OpenReview.net.

Dosovitskiy, A., Fischer, P., Springenberg, J. T., Riedmiller, M., & Brox, T. (2015). Discriminative unsupervised feature learning with exemplar convolutional neural networks. *TPAMI, 38*(9), 1734–1747.

Dosovitskiy, A., Springenberg, J. T., Riedmiller, M., & Brox, T. (2014). Discriminative unsupervised feature learning with convolutional neural networks. *NeurIPS, 27*, 766–774.

El-Nouby, A., Izacard, G., Touvron, H., Laptev, I., Jegou, H., & Grave, E. (2021). Are large-scale datasets necessary for self-supervised pre-training? arXiv:2112.10740

Ermolov, A., Siarohin, A., Sangineto, E., & Sebe, N. (2021). Whitening for self-supervised representation learning. In *ICML* (pp. 3015–3024). PMLR.

Fang, Y., Dong, L., Bao, H., Wang, X., & Wei, F. (2022). Corrupted image modeling for self-supervised visual pre-training. arXiv:2202.03382

Gallinari, P., Lecun, Y., Thiria, S., & Soulie, F. F. (1987). Mémoires associatives distribuées: une comparaison (distributed associative memories: A comparison). In *Proceedings of COGNITIVA 87, Paris, La Villette, May 1987*. Cesta-Afcet.

Garrido, Q., Chen, Y., Bardes, A., Najman, L. (2022). On the duality between contrastive and non-contrastive self-supervised learning. CoRR, arXiv:2206.02574

Gidaris, S., Bursuc, A., Komodakis, N., Pérez, P., & Cord, M. (2020). Learning representations by predicting bags of visual words. In *CVPR* (pp. 6928–6938).

Gidaris, S., Bursuc, A., Puy, G., Komodakis, N., Cord, M., & Pérez, P. (2020). Online bag-of-visual-words generation for unsupervised representation learning. arXiv:2012.11552

Goyal, P., Caron, M., Lefaudeux, B., Xu, M., Wang, P., Pai, V., Singh, M., Liptchinsky, V., Misra, I., Joulin, A., et al. (2021). Self-supervised pretraining of visual features in the wild. arXiv:2103.01988

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Pires, B. A., Guo, Z. D., Azar, M. G., et al. (2020). Bootstrap your own latent: A new approach to self-supervised learning. arXiv:2006.07733

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., & Girshick, R. (2022). Masked autoencoders are scalable vision learners. In *CVPR*.

He, K., Fan, H., Wu, Y., Xie, S., & Girshick, R. B. (2020). Momentum contrast for unsupervised visual representation learning. In *CVPR* (pp. 9726–9735). Computer Vision Foundation/IEEE.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In *ICCV* (pp. 2961–2969).

Henaff, O. (2020). Data-efficient image recognition with contrastive predictive coding. In *ICML* (pp. 4182–4192). PMLR.

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science, 313*(5786), 504–507.

Hinton, G. E., & Zemel, R. S. (1994). Autoencoders, minimum description length, and helmholtz free energy. *NeurIPS, 6*, 3–10.

Huang, J., Dong, Q., Gong, S., & Zhu, X. (2019). Unsupervised deep learning by neighbourhood discovery. In *ICML* (pp. 2849–2858). PMLR.

Huang, Z., Jin, X., Lu, C., Hou, Q., Cheng, M.-M., Fu, D., Shen, X., & Feng, J. (2022). Contrastive masked autoencoders are stronger vision learners. arXiv:2207.13532

Huang, G., Sun, Y., Liu, Z., Sedra, D., & Weinberger, K. Q. (2016). Deep networks with stochastic depth. In *ECCV* (pp. 646–661). Springer.

Huang, L., You, S., Zheng, M., Wang, F., Qian, C., & Yamasaki, T. (2022). Green hierarchical vision transformer for masked image modeling. arXiv:2205.13515

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*.

Jing, L., Zhu, J., & LeCun, Y. (2022). Masked siamese convnets. arXiv:2206.07700

Kakogeorgiou, I., Gidaris, S., Psomas, B., Avrithis, Y., Bursuc, A., Karantzalos, K., & Komodakis, N. (2022). What to hide from your students: Attention-guided masked image modeling. In *ECCV*.

Kingma, Diederik P., & Welling, M. (2013). Auto-encoding variational bayes. arXiv:1312.6114.

Kong, X., & Zhang, X. (2022). Understanding masked image modeling via learning occlusion invariant feature. arXiv:2208.04164

LeCun, Y. (1987). Mod'eles connexionistes de l'apprentissage. PhD thesis, Universit'e de Paris VI.

Li, X., Ge, Y., Yi, K., Hu, Z., Shan, Y., & Duan, L.-Y. (2022). mc-beit: Multi-choice discretization for image bert pre-training. In *ECCV*.

Li, X., Wang, W., Yang, L., & Yang, J. (2022). Uniform masking: Enabling mae pre-training for pyramid-based vision transformers with locality. arXiv:2205.10063

Li, S., Wu, D., Wu, F., Zang, Z., Wang, K., Shang, L., Sun, B., Li, H., Li, S., et al. (2022). Architecture-agnostic masked image modeling-from vit back to cnn. arXiv:2205.13943

Li, G., Zheng, H., Liu, D., Su, B., & Zheng, C. (2022). Semmae: Semantic-guided masking for learning masked autoencoders. arXiv:2206.10207

Li, J., Zhou, P., Xiong, C., & Hoi, S. C. H. (2020). Prototypical contrastive learning of unsupervised representations. arXiv:2005.04966

Li, Z., Chen, Z., Yang, F., Li, W., Zhu, Y., Zhao, C., Deng, R., Liwei, W., Zhao, R., Tang, M., et al. (2021). Mst: Masked self-supervised transformer for visual representation. *NeurIPS, 34*, 13165–13176.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Lawrence Zitnick, C. (2014). Microsoft coco: Common objects in context. In *ECCV* (pp. 740–755). Springer.

Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., Ning, J., Cao, Y., Zhang, Z., Dong, L., Wei, F. & Guo, B. (2021). Swin transformer v2: Scaling up capacity and resolution. Cornell University.

Liu, H., Jiang, X., Li, X., Guo, A., Jiang, D., & Ren, B. (2022). The devil is in the frequency: Geminated gestalt autoencoder for self-supervised visual pre-training. arXiv:2204.08227

Loshchilov, I., & Hutter, F. (2017). Decoupled weight decay regularization. arXiv:1711.05101

Noroozi, M., & Favaro, P. (2016). Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV* (pp. 69–84). Springer.

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

Peng, X., Wang, K., Zhu, Z., & You, Y. (2022). Crafting better contrastive views for siamese representation learning. In *CVPR*.

Qi, J., Zhu, J., Ding, M., Chen, X., Luo, P., Wang, L., Wang, X., Liu, W., & Wang, J. (2023). *Understanding self-supervised pretraining with part-aware representation learning*. Report: Tech.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., & Sutskever, I. (2021). Zero-shot text-to-image generation. In M. Meila & T. Zhang (Eds.), *ICML* (Vol. 139, pp. 8821–8831). PMLR.

Ranzato, M., Poultney, C., Chopra, S., LeCun, Y., et al. (2007). Efficient learning of sparse representations with an energy-based model. *NeurIPS, 19*, 1137.

Russakovsky, O., Deng, J., Hao, S., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. *IJCV, 115*(3), 211–252.

Tao, C., Zhu, X., Huang, G., Qiao, Y., Wang, X. & Dai, J. (2022). Siamese image modeling for self-supervised vision representation learning. arXiv:2206.01204

Tian, Y., Xie, L., Fang, J., Shi, M., Peng, J., Zhang, X., Jiao, J., Tian, Q., & Ye, Q. (2022). Beyond masking: Demystifying token-based pre-training for vision transformers. arXiv:2203.14313

Tian, Y., Xie, L., Zhang, X., Fang, J., Xu, H., Huang, W., Jiao, J., Tian, Q., & Ye, Q. (2021). Semantic-aware generation for self-supervised visual representation learning. arXiv:2111.13163

Tian, Y., Sun, C., Poole, B., Krishnan, D., Schmid, C., & Isola, P. (2020). What makes for good views for contrastive learning? *NeurIPS, 33*, 6827–6839.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., & Jégou, H. (2020). Training data-efficient image transformers & distillation through attention. arXiv:2012.12877

van den Oord, A., Li, Y., & Vinyals, O. (2018). Representation learning with contrastive predictive coding. arXiv:1807.03748

Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research,9*(11).

Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. In *ICML* (pp. 1096–1103).

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., & Manzagol, P.-A. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research, 11*, 3371–3408.

Wang, W., Bao, H., Dong, L., Bjorck, J., Peng, Z., Liu, Q., Aggarwal, K., Khan, O., Singhal, S., Som, S., & Wei, F. (2023). Image as a foreign language: Beit pretraining for all vision and vision-language tasks.

Wang, L., Liang, F., Li, Y., Ouyang, W., Zhang, H., & Shao, J. (2022). Repre: Improving self-supervised vision transformer with reconstructive pre-training. arXiv:2201.06857

Wang, X., Zhang, R., Shen, C., Kong, T., & Li, L. (2021). Dense contrastive learning for self-supervised visual pre-training. In *CVPR* (pp. 3024–3033).

Wei, C., Fan, H., Xie, S., Wu, C.-Y., Yuille, A., & Feichtenhofer, C. (2021). Masked feature prediction for self-supervised visual pre-training. arXiv:2112.09133

Wei, L., Xie, L., Zhou, W., Li, H., & Tian, Q. (2022). Mvp: Multimodality-guided visual pre-training. In *ECCV*.

Wu, Z., Xiong, Y., Yu, S. X., & Lin, D. (2018). Unsupervised feature learning via non-parametric instance discrimination. In *CVPR* (pp. 3733–3742).

Xiao, T., Liu, Y., Zhou, B., Jiang, Y., & Sun, J. (2018). Unified perceptual parsing for scene understanding. In *ECCV* (pp. 418–434).

Xie, Z., Geng, Z., Hu, J., Zhang, Z., Hu, H., & Cao, Y. (2022). Revealing the dark secrets of masked image modeling. arXiv:2205.13543.

Xie, J., Girshick, R., & Farhadi, A. (2016). Unsupervised deep embedding for clustering analysis. In *ICML* (pp. 478–487). PMLR.

Xie, J., Li, W., Zhan, X., Liu, Z., Ong, Y. S., & Loy, C. C. (2022). Masked frequency modeling for self-supervised visual pre-training. arXiv:2206.07706

Xie, Z., Lin, Y., Zhang, Z., Cao, Y., Lin, Stephen, & Hu, H. (2021). Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. In *CVPR* (pp. 16684–16693).

Xie, Z., Zhang, Z., Cao, Y., Lin, Y., Bao, J., Yao, Z., Dai, Q., & Hu, H. (2021). Simmim: A simple framework for masked image modeling. arXiv:2111.09886

Xie, Z., Zhang, Z., Cao, Y., Lin, Y., Wei, Y., Dai, Q., & Hu, H. (2022). On data scaling in masked image modeling. arXiv:2206.04664

Yang, J., Parikh, D., & Batra, D. (2016). Joint unsupervised learning of deep representations and image clusters. In *CVPR* (pp. 5147–5156).

Yi, K., Ge, Y., Li, Xiaotong, Y., Shusheng, L., Dian, W., Jianping, S. Y., & Qie, X. (2022). Masked image modeling with denoising contrast. arXiv:2205.09616

You, Y., Gitman, I., & Ginsburg, B. (2017). Large batch training of convolutional networks. arXiv:1708.03888

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., & Yoo, Y. (2019). Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV* (pp. 6023–6032).

Zbontar, J., Jing, L., Misra, I., LeCun, Y., & Deny, S. (2021). Barlow twins: Self-supervised learning via redundancy reduction. arXiv:2103.03230

Zhang, X., Chen, J., Yuan, J., Chen, Q., Wang, J., Wang, X., Han, S., Chen, X., Pi, J., Yao, K., Han, J., Ding, E., & Wang, J. (2022). CAE v2: Context autoencoder with CLIP target. *CoRR* arxiv:2211.09799.

Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2017). mixup: Beyond empirical risk minimization. In *ICLR*.

Zhang, R., Isola, P., & Efros, A. A. (2016). Colorful image colorization. In *ECCV* (pp. 649–666). Springer.

Zhang, H., Li, F., Liu, S., Zhang, L., Hang, S., Zhu, J., Ni, L. M., & Shum, H.-Y. (2023). Dino: Detr with improved denoising anchor boxes for end-to-end object detection.

Zhang, X., Tian, Y., Huang, W., Ye, Q., Dai, Q., Xie, L., & Tian, Q. (2022). Hivit: Hierarchical vision transformer meets masked image modeling. arXiv:2205.14949

Zhou, J., Wei, C., Wang, H., Shen, W., Xie, C., Yuille, A., & Kong, T. (2021). Ibot: Image bert pre-training with online tokenizer. arxiv:2111.07832

Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., & Torralb, A.(2017). Scene parsing through ade20k dataset. In *CVPR* (pp. 633–641).

Zhuang, C., Zhai, A. L., & Yamins, D. (2019). Local aggregation for unsupervised learning of visual embeddings. In *ICCV* (pp. 6002–6012).

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.