Abstract—Masked image modeling (MIM) has achieved promising results on various vision tasks. However, the limited discriminability of learned representation manifests there is still plenty to go for making a stronger vision learner. Towards this goal, we propose Contrastive Masked Autoencoders (CMAE), a new self-supervised pre-training method for learning more comprehensive and capable vision representations. By elaborately unifying contrastive learning (CL) and masked image model (MIM) through novel designs, CMAE leverages their respective advantages and learns representations with both strong instance discriminability and local perceptibility. Specifically, CMAE consists of two branches where the online branch is an asymmetric encoder-decoder and the momentum branch is a momentum updated encoder. During training, the online encoder reconstructs original images from latent representations of masked images to learn holistic features and the momentum encoder, fed with the full images, enhances the feature discriminability via contrastive learning with its online counterpart. To make CL compatible with MIM, CMAE introduces two new components, i.e., pixel shifting for generating plausible positive views and feature decoder for complementing features of contrastive pairs. Thanks to these novel designs, CMAE effectively improves the representation quality and transfer performance over its MIM counterpart. CMAE achieves the state-of-the-art performance on highly competitive benchmarks of image classification, semantic segmentation and object detection. Notably, CMAE-Base achieves 85.3% top-1 accuracy on ImageNet and 52.5% mIoU on ADE20k, surpassing previous best results by 0.7% and 1.8% respectively.

Index Terms—Masked image modeling (MIM), contrastive learning, self-supervised learning.

I. INTRODUCTION

ASKED image modeling (MIM) [23], [30], [56] has been attracting increasing attention recently in the self-supervised learning field, due to its method simplicity and capability of learning rich and holistic representations. Following the idea of masked language modeling in NLP [19], they randomly mask a large portion of the training image patches and use an auto-encoder [31] to reconstruct the original signals (e.g., raw pixels, offline extracted features) of the masked patches. It has been shown in [23], [30], [56] that such a simple framework outperforms previous self-supervised learning methods in both ImageNet classification [18] and some downstream tasks, like object detection and semantic segmentation.

When we reflect on the success of MIM, it is inevitable to compare it with another well-proven and prevailing SSL method, i.e., contrastive learning (CL) [2], [42]. By adopting a simple discriminative idea that pulling closer representations from the same image and pushing away different images, CL methods naturally endow the pretrained model with strong instance discriminability. In contrast to CL, MIM focuses more on learning local relations in input image for fulfilling the reconstruction task, instead of modeling the relation among different images [35]. Therefore, it is suspected that MIM is less efficient in learning discriminative representations. This issue has been manifested by experimental results in [30], [56]. Based on above analysis, it is thus natural to ask such a question: can we leverage contrastive learning to further strengthen the representation learned by MIM methods? or, in other words, would MIM methods benefit from contrastive learning? Along this direction, a few contemporary works attempt to train vision representation models [46], [65] by simply combining contrastive learning and MIM learning objectives. But they only show marginal performance gain compared to MIM methods. These results signify that it is non-trivial to fully leverage the advantages of both frameworks. The challenges are ascribed to various distinctions between them, including input augmentations, training objectives, model architectures, etc.

To overcome the challenges and learn better image representations for downstream tasks, we aim to explore a possible way to boost the MIM with contrastive learning in a unified framework. With a series of careful studies, we find that input view augmentation and latent feature alignment play important roles in harmonizing MIM and contrastive learning. We thus put dedicated efforts to these components to develop our method.

An overview of the proposed method is shown in Fig. 1. More specifically, our method introduces a contrastive MAE (CMAE) framework for representation learning. It adopts a siamese architecture [5]. One branch is an online updated asymmetric encoder-decoder that learns latent representations to reconstruct masked images from a few visible patches, similar to MAE. The

Zhicheng Huang, Xiaojie Jin, Chengze Lu, Qibin Hou, Ming-Ming Cheng, Dongmei Fu, Xiaohui Shen, and Jiashi Feng
other branch is a momentum encoder that provides contrastive learning supervision. To leverage the contrastive learning to improve the feature quality of encoder output, we introduce an auxiliary feature decoder into the online branch, whose output features are used for contrastive learning with the momentum encoder outputs.

We carefully design each CMAE component to enable contrastive learning to benefit the MIM. Different from online encoder whose inputs only contain the visible patches, the CMAE momentum encoder is fed with the full set of image patches. This design ensures semantic integrity of its output features to guide the online encoder. Another notable design choice is: our method uses two decoders, one is to predict the image pixel and perform the MIM task; and another is to recover the features of masked tokens. Since the semantics of each patch are incomplete and ambiguous, it is problematic to use the features of patches directly for contrastive learning. Using an auxiliary feature decoder can address this issue and thus benefit the latent representation learning within the online branch. Moreover, different from existing methods that use strong spatial data augmentations for inputs, we propose a pixel shifting augmentation method for generating more plausible positive views in contrastive learning. Such a simple augmentation is proven effective for improving MIM with contrastive learning. With the above novel designs, the online encoder of our CMAE method can learn more discriminative features of holistic information and achieve state-of-the-art performance on various pre-training and transfer learning vision tasks.

Our contributions are summarized as follows. 1) We propose a new CMAE method to explore how to improve the representation of MIM by using contrastive learning. Its learned representations not only preserve the local context sensitive features but also model the instance discriminativeness among different images. 2) To impose contrastive learning upon MIM, we propose a feature decoder to complement the masked features and a weakly spatial shifting augmentation method for generating plausible contrastive views, both of which are effective in improving the encoder feature quality. 3) As shown in Fig. 2, our method significantly improves the learned representation of MIM and sets new state-of-the-art performance. Notably, compared with prior arts, CMAE achieves absolute gains of 0.7% on ImageNet-1k classification validation split, 1.8% mIoU on ADE20K semantic segmentation validation dataset and 0.4% APb and 0.5% APm on CoCo validation split.

II. RELATED WORK

Self-supervised learning is attracting increasing attention in computer vision. A bunch of methods have been proposed to advance this technique from different perspectives [4], [12], [22], [26], [30], [42], [60], [65]. Broadly speaking, these methods can be categorized into two groups depending on their employed pretext tasks, i.e., contrastive learning [27], [29], [32] and mask image modeling [3], [30], [56].

Contrastive learning aims to learn instance discriminative representations to distinguish an image from the others. This is achieved by pulling together the representation of different views of an individual image and pushing away the other images. Thus most contrastive methods adopt siamese network [13], [26], [29]. To create different view for the same image, a plentiful of methods have been deployed [12], [42], [47]. Among these methods, data augmentations are most commonly in contrastive learning and investigated in SimCLR, [12]. In practice, contrastive methods rely heavily on a large number of negative samples [12], [29], [47], [53]. To better utilize negative samples, MoCo [29] uses a large queue to cache negative examples in memory such that it can take in more negative examples for contrastive learning. BYOL [26] uses an online encoder to predict the output of a momentum encoder, where the momentum encoder is key to avoiding training collapse. To simplify BYOL, SimSiam [13] proposes the stop-gradient technique to replace the momentum updating. Besides, there are some methods to solve this issue from different views [7], [8], [9], [60]. SwAV [9] proposed an online clustering method and adopted the Sinkhorn-Knopp [16] transform to assist the clustering for each batch. Barlow Twins [60] adopts the cross-correlation matrix to constrain the network to void training collapse. Recently, MoCo-v3 [14] and DINo [10] are proposed to extend MoCo [29]
and BYOL [26] respectively by using Vision Transformer (ViT) as their model backbones. Although contrastive learning methods provide discriminative vision representations, most of them focus on learning global representations while lacking spatial-sensitive representation.

Mask image modeling [3], [11], [21] is inspired by the success of Masked Language Modeling in NLP [19], [43] and learns vision representation by constructing the original signal from partial observations. Based on the reconstruction target, these methods can be divided into: pixel-domain reconstruction [22], [30], [38], [50], [56] and auxiliary features/tokens prediction [3], [15], [20]. SimMIM [56] and MAE [30] propose to reconstruct the raw pixel values from either the full set of image patches (SimMIM) or partially observed patches (MAE) to reconstruct the raw image. Compared with SimMIM, MAE is more pre-training efficient because of masking out a large portion of input patches. To learn more semantic features, MaskFeat [50] introduces the low-level local features (HOG [17]) as the reconstruction target, and Ge²-AE adopts the MIM task in frequency domain [38], while CIM [22] opts for more complex input.

Several methods adopt an extra model to generate the target to pre-train the encoder. For instance, BEiT [3] uses the discretized tokens from an offline tokenizer [44] to train the encoder. PeCo [20] instead uses an offline visual vocabulary to guide the encoder. Differently, CAE [15] uses both the online target and offline network to guide the training of encoder. Furthermore, iBOT [65] introduces an online tokenizer to produce the target to distill the encoder. MVP [51] adopts a structure identical to BEiT [3] and replaces the tokenizer (e.g., d-VAE in BEiT [3]) with the vision branch of a multimodal model like CLIP that is pre-trained on image-text pairs. Similarly, SIM adopts the siamese network to reconstruct the representations of tokens, based on another masked view [46]. MSN [1] matches the representation of masked image to that of original image using a set of learnable prototypes.

Despite MIM models exhibiting favorable optimization properties [52] and delivering promising performance, their focus is on learning relationships among the tokens in the input image, rather than modeling the relation among different images as in contrastive learning, which results in less discriminative learned representations. Our method diverges from existing works by proposing innovative designs to fully leverage the advantages of MIM and contrastive learning, thereby providing local context-sensitive representations with the desired discriminativeness for input images.

III. METHOD

A. Framework

The overall framework of our method is illustrated in Fig. 3 that consists of three components. The online encoder and decoder learn to reconstruct input images from masked observations. Different from existing MIM methods (e.g., MAE [30] and SimMIM [56]), our method further processes the input image via a spatially shifted cropping operation. More importantly, our decoder incorporates an additional feature decoder for predicting the input image features. The momentum encoder transforms the augmented view of the input image into a feature embedding for contrastive learning with the predicted one from the online feature decoder. In this way, the learned representations by the online encoder capture not only holistic features of the input images but also discriminative features, thus achieving better generalization performance. We now elaborate on these components in detail.

Let us denote the input image $I$ to the online encoder as being tokenized into a sequence of $N$ image patch tokens $\{x_i^<\}_{i=1}^N$, where $N$ is the total number of image patches. For the masked version of $I$, the set of visible tokens is represented as $\{x^>\}$. Similarly, the input image $I$ is tokenized into a sequence
of image patch tokens \( \{ x_j^v \}_{j=1}^N \) to serve as the input to the momentum encoder.

**Online Encoder:** The online encoder \( F_s \) maps the visible tokens \( x_j^v \) to embedding features \( z_j^v \). Given the token sequence \( \{ x_j^v \}_{j=1}^N \), we mask out a large ratio of patches and feed the visited patches to the online encoder. The online encoder adopts the Vision Transformer (ViT) architecture [21], following MAE [30]. It first embeds the visible tokens \( x_j^v \) by linear projection as token embeddings, and adds the positional embeddings [49] \( p_j^v \). We feed the fused embedding to a sequence of transformer blocks, and get the embedding features \( z_j^v \).

\[
  z_j^v = F_s(x_j^v + p_j^v) \tag{1}
\]

After pre-training, the online encoder \( F_s \) is used for extracting image representations in downstream tasks.

**Momentum Encoder:** The momentum encoder is introduced for providing contrastive supervision for the online encoder to learn discriminative representations. Different from existing siamese-based methods [10], [65], our momentum encoder \( F_t \) only serves for contrastive learning, as well as guiding the online encoder to learn more discriminative features. It shares the same architecture as the online encoder \( F_s \), but takes the whole image as input, in order to reserve the semantic integrity and the discriminativeness of the learned representations. Using the whole image as input to the momentum encoder is important for the method performance, which is experimentally verified in Section IV-D. Unlike tokens in NLP, whose semantic are almost certain, image token is ambiguous in its semantic meaning [65]. To avoid ambiguity, we adopt global representations for contrastive learning. The mean-pooled feature of momentum encoder is used for its simplicity, i.e.,

\[
  z^t = \frac{1}{N} \sum_{j=1}^{N} F_t(x_j^v), \tag{2}
\]

where \( x_j^v \) is the input token for momentum encoder, \( z_j^v \) is the output sequence of momentum encoder and \( z^t \) is the feature obtained after performing mean pooling operation on the output sequence \( z_j^t \), which is used to represent the input image.

Different from the online encoder, we update parameters of the momentum encoder by exponential moving average (EMA). That is, denoting the parameters of \( F_s \) and \( F_t \) as \( \theta_s \) and \( \theta_t \) respectively, the parameters are updated by \( \theta_t \leftarrow \mu \theta_t + (1 - \mu) \theta_s \). Here \( \mu \) is fixed as 0.996 across the experiments. Momentum update is used since it stabilizes the training by fostering smooth feature changes, as found in MoCo [29] and BYOL [26].

**Online Decoder:** The decoder aims to map the latent features \( z_j^v \) and MASK token features to the feature space of the momentum encoder and the original images. Specifically, the decoder receives both the encoded visible tokens \( z_j^v \) and MASK tokens \( z_j^m \).

Similar to MAE [30], position embeddings are added to input tokens. Due to different mapping targets, our online decoder has two branches of decoder structure, one is a pixel decoder, and another is a feature decoder. The pixel decoder \( G_p \) learns to reconstruct the pixel of the masked patches. We use the full set of tokens, which contains both \( z_j^v \) and \( z_j^m \), to predict the pixel of patches \( y^m \). This module can promote the model to learn holistic representation for each patch in an image. We set the pixel decoder to be stacked transformer blocks:

\[
  y^m_m = \mathbb{I} \cdot G_p(z^v_j, z^m_j), \tag{3}
\]

where \( \mathbb{I} \) is an indicator to only select the prediction corresponding to masked tokens from output sequence \( y^p \), and \( y^m_m \) is the output prediction for the masked patches.

To align with the output of the momentum encoder, feature decoder \( G_f \) is applied to recover the feature of masked tokens. The feature decoder has the same structure as the pixel decoder but non-shared parameters for serving a different learning target. The prominence of such design choices will be discussed in the architecture part in Section III-D. Given the encoded visible tokens \( z^v_j \), we add the masked tokens \( z^m_j \) and use this full set to predict the feature of masked tokens. Similar as done in momentum encoder, we apply the mean pooling operation on the output of feature decoder \( y^f \) as the whole image representation \( y_s \), and then use this feature for contrastive learning.

\[
  y_s = \frac{1}{N} \sum G_f(z^v_j, z^m_j), \tag{4}
\]
where \( N \) is the total number of tokens in the full set.

### B. View Augmentations

Typically, masking image modeling pre-training tasks only utilizes a single view of the input image, which only contains visited patches. But contrastive learning often adopts two different augmented views. To make MIM and contrastive learning be compatible with each other, our method also generates two different views and feeds them to its online and momentum branches, respectively.

In contrastive learning, the most commonly used view augmentations can be divided into two types: spatial transfer (e.g., random resized cropping, flipping) and color transfer (e.g., color jittering and random grayscaling). For MIM tasks, color enhancements degrade the results [30], so we do not apply them to the input of the online branch. Spatial and color data augmentations are applied to the momentum branch input to avoid a trivial solution.

We first consider two branches using two different random crops, following the common practice in contrastive learning. However, we observe this recipe has an adverse effect on model performance (refer to Section IV-D). We conjecture that this issue is related with the large disparity between the inputs of online/momentum encoders when randomly cropped regions are far apart or scarcely semantic-relevant. Different with using intact paired views in usual contrastive methods, the operation of masking out a large portion of input in MIM may amplify such disparity and therefore creates false positive views. Consequently, performing contrastive learning on these misaligned positive pairs actually incurs noise and hampers the learning of discriminative and meaningful representations.

To address the above issue, we propose a weakly augmentation method named pixel shifting for generating the inputs of online/momentum encoders. The core idea is first to obtain a master image by a resized random cropping from the original image. Then two branches share the same master image and generate respective views by slightly shifting cropping locations over the master image. In more details, we denote the master image as \( I \). The shape of \( I \) is \((w + p, h + p, 3)\), where \( w, h \) is the width and height of target input size for our model and \( p \) is the longest shifting range allowed. For online branch, we use the region of \([0 : w, 0 : h, :]\) as our input image \( I_p \). For momentum branch, we use the region of \([r_w : r_w + w, r_h : r_h + h, :]\) as our input image \( I_t \). \( r_w \) and \( r_h \) are independent random values in the range of \([0, p)\). Subsequently, we only apply masking operation without color augmentation for the input of online encoder \( I_t \), which is consistent with MAE. For the momentum encoder’s input image, we utilize color augmentation but do not apply masking operation. The distinct augmentations for each encoder input generate different views of the same image to facilitate contrastive learning between the online and momentum encoders.

### C. Training Objective

**Reconstruction Loss:** Following [30], we use the normalized pixel as target in the reconstruction task. We adopt the Mean Squared Error (MSE) as loss function and compute the loss only on masked patches between the pixel decoder prediction and the original image. The math formulation is

\[
L_r = \frac{1}{N_m} \sum (y'_m - y_m)^2, \tag{5}
\]

where \( N_m \) is the number of masked patches in an image, and \( L_r \) is the loss value.

**Contrastive Loss:** For clarity, we describe the contrastive loss design of our method from two aspects: loss function and head structure. Two widely used loss functions are taken into consideration, i.e., InfoNCE [12], [29] loss and BYOL-style [10], [26] loss. The former seeks to simultaneously pull close positive views from the same sample and push away negative samples while the latter only maximizes the similarity between positive views. Although some recent works find they may be inherently unified [45], we still analyze them separately due to their diverse effects on representation learning. In our method, we observe better performance using InfoNCE [42] so we use it defaultly. Details are referred to Section IV-D. For the head structure, we adopt the widely used “projection-prediction” structure following [14], [26]. Specifically, we append the “projection-prediction” and “projection” head to feature decoder and momentum encoder respectively. The projection head \( \mathcal{H}_t \) with momentum encoder is also updated by exponential moving average. Due to the large differences on generating inputs for online/momentum encoder (refer to Section III-B), we use asymmetric contrastive loss, which is distinguished from previous methods [14], [26]. The representation from feature decoder \( y_s \) is transformed by the “projection-prediction” structure to get \( y_p^t \). Similarly for the representation from momentum encoder \( z^t \), we apply the projection head and get \( z_p^t \). We then compute the cosine similarity \( \rho \) between them:

\[
\rho = \frac{y_p^t \cdot z_p^t}{\|y_p^t\|_2 \|z_p^t\|_2}. \tag{6}
\]

We denote \( \rho^+ \) as the positive pairs cosine similarity, which is constructed by \( y_p^t \) and \( z_p^t \) from the same image, \( \rho^-_j \) indicates the cosine similarity for the \( j \)-th negative pair. We use the \( z_p^t \) from different images in a batch to construct negative pairs. The loss function of InfoNCE loss is

\[
L_c = - \log \frac{\exp(\rho^+ / \tau)}{\exp(\rho^+ / \tau) + \sum_{j=1}^{K-1} (\exp(\rho^-_j / \tau))}, \tag{7}
\]

where \( \tau \) is the temperature constant, which is set to 0.07. \( K \) is the batch size.

The overall learning target is a weighted combination of reconstruction loss \( L_r \) and contrastive loss \( L_c \) defined as:

\[
L = L_r + \lambda_c L_c. \tag{8}
\]

### D. Connections and Analysis

To elucidate the correlations and distinctions between the CMAE and preceding methodologies, we undertake comparative assessments from various perspectives such as training objective, input, and architecture. The outcomes are demonstrated in Table I. Our primary focus is on methodologies that utilize...
either contrastive information in MIM or masked image input. Approaches that solely employ MIM or contrastive learning are not within the purview of this discussion, as they are evidently distinct from our method.

Training Objective: The CMAE leverages both the reconstruction loss and contrastive loss during optimization. As inferred from (7), the contrastive loss in CMAE encompasses both intra-view matching and inter-image contrast. Consequently, the generated representations are encouraged to exhibit desirable characteristics of instance discrimination and spatial sensitivity. In contrast, methodologies such as MSN [1] and ExtreMA [54], which have divergent motivations from ours, disregard reconstruction loss and employ masked input for regularization or data augmentation purposes. iBot [65] exclusively adopts a distillation loss between positive views by maximizing intra-view matching scores, overlooking contrastive learning with negative samples. Additionally, CMAE only adopts an asymmetric loss for contrastive learning, which is less computationally expensive than iBot. Although SIM [46] also utilizes both losses, it differs from CMAE in terms of the reconstruction target. While CMAE restores the masked content of the same view, SIM reconstructs the features of another view. Our empirical results demonstrate that CMAE is not just simpler but also more effective in representation learning, as evidenced by superior performance.

Input: The majority of prior contrastive learning methods [12], [29] implement robust augmentation techniques (e.g., random crop, random scale) to generate positive views from the same image. These operations are also commonly used in contrastive learning models under the masked image modeling scenario, such as the iBot. However, considering that the masking operation, which utilizes a large masking ratio (e.g., 75% in [30]), already significantly degrades the input, applying these augmentations further could generate invalid positive views, thereby hindering contrastive learning. In contrast, we propose a novel, moderate data augmentation operation called pixel shifting for achieving better alignment between positive views. Compared to ExtreMA [54], which employs the exact same view in two siamese branches, pixel shifting introduces a moderate input variance, which proves beneficial for contrastive learning (refer to Table IV(a)).

Architecture: In CMAE, a lightweight feature decoder is appended after the online encoder to supplement the masked

### TABLE I

| Training objective | Input | Architecture | Accuracy |
|--------------------|-------|--------------|----------|
| reconstr | intra-view match | inter-image contrast | positive view alignment | feature complement encoder/decoder | |
| MSN [1] | ✓ | ✓ | ✓ | ✓ | ✓ | 83.4 |
| ExtreMA [54] | ✓ | ✓ | ✓ | ✓ | ✓ | 83.7 |
| MAE [30] | ✓ | ✓ | ✓ | N.A. | ✓ | 83.9 |
| CMAE [15] | ✓ | ✓ | ✓ | ✓ | ✓ | 84.0 |
| iBot [65] | ✓ | ✓ | ✓ | ✓ | ✓ | 84.1 |
| SIM [46] | ✓ | ✓ | ✓ | ✓ | ✓ | 84.2 |
| CMAE | ✓ | ✓ | ✓ | ✓ | ✓ | 84.7 |

The top-1 accuracy on ImageNet is also presented. Please refer to Section 3.4 for more detailed explanations.

### TABLE II

| Method | Pre-training epochs | Params.(M) | Supervision | Accuracy |
|--------|---------------------|-----------|-------------|----------|
| MoCo-v3 [14] | 300 | 86 | RGB | 83.2 |
| DINo [10] | 300 | 86 | RGB | 82.8 |
| CIM [22] | 300 | 86 | RGB | 83.3 |
| BEiT [3] | 800 | 86 | DALLE | 83.2 |
| SimMIM [56] | 800 | 86 | RGB | 83.8 |
| PeCo [20] | 800 | 86 | Perceptual Codebook | 84.5 |
| MaskFeat [50] | 1600 | 86 | HOG | 84.0 |
| CMAE [15] | 1600 | 86 | DALLE+RGB | 83.9 |
| iBot [65] | 1600 | 86 | RGB | 84.0 |
| SIM [46] | 1600 | 86 | RGB | 83.8 |
| MAE [30] | 1600 | 86 | RGB | 83.6 |
| CMAE (ours) | 1600 | 86 | RGB | 84.4 |
| CMAE (ours) | 1600 | 86 | RGB | 84.7 |
| ConvMAE* [23] | 800 | 86 | RGB | 84.6 |
| ConvMAE* [23] | 1600 | 86 | RGB | 84.6 |
| CMAE* (ours) | 800 | 86 | RGB | 85.0 |
| CMAE* (ours) | 1600 | 86 | RGB | 85.3 |

We evaluate them with the top-1 accuracy on ImageNet. The symbol of * throughout experiments denotes using convolutions instead of linear transformation as the tokenizer for visual patches.
features. This is a notable distinction from other methods, such as SIM and iBot, which directly utilize the representations of the visible patches to match that of the unmasked view. We contend that conducting contrastive learning between the features of the masked parts and the input images is impractical, considering they exhibit different levels of abstraction and semantic coverage. The feature decoder is anticipated to facilitate optimization by diminishing the distribution gap between contrastive features. The efficacy of the feature decoder is empirically validated, as shown in Table IV(c). Notably, the design of CMAE is non-intrusive, allowing for its straightforward application to existing MIM models, such as MAE and ConvMAE, without necessitating significant modifications to the MIM model.

IV. EXPERIMENTS
A. Implementation Details

Pre-Training: We follow the settings of MAE [30] to pre-train our model. We adopt AdamW [40] optimizer as default, and the momentum is set to $\beta_1 = 0.9, \beta_2 = 0.95$. Besides, the weight decay is set to 0.05. We use the linear scaling rule [25]: $lr = base_{lr} \times \text{batch size}/256$ to set the learning rate. The base learning rate is $1.5 \times 10^{-4}$ with a batch size of 4096. Cosine learning rate schedule [39] with a warmup of 40 epochs is adopted. All pre-training experiments are conducted on 32 NVIDIA A100 GPUs.

Encoder Structure: We use the ViT [21] base model as our default setting. To further validate the extensibility of our proposed model, we replace the ViT with a hybrid convolutional ViT which is also used by ConvMAE [23]. In the hybrid ViT, a multi-layer convolutional network [34] is used as token projection. Note the hybrid ViT is made to have the same model size as the ViT counterpart for fair comparison. We also experiment with scaled up encoders for evaluating the scalability of our method.

B. Results on ImageNet

Following existing works [3], [14], [30], [56], we use ImageNet-1K [18] which consists of 1.3 M images of 1k categories as the pre-training and fine-tuning dataset. The dataset contains two subsets: the training set and the validation set. We only use the training set to pre-train CMAE. After pre-training, the CMAE online encoder is used for fine-tuning on ImageNet-1k training set for 100 epochs. For the model pre-trained with 300 epochs, we adopt $5 \cdot 10^{-4}$ as the base learning rate.
rate in fine-tuning. Since the longer pre-training schedule (1600 epochs) makes the model learn better initialization weights for fine-tuning [57], we set a smaller base learning rate of $2.5e^{-4}$. Besides, we follow the common fine-tuning practices to regularize the model using mixup [61], cutmix [59], drop path [33], etc.

In Table II, we compare CMAE with competing methods on the fine-tuning classification accuracy on ImageNet. CMAE achieves a top-1 accuracy of 84.7%, which is 1.1% higher than MAE [30]. Among all models using ViT architecture, CMAE achieves the best performance. Compared with contrastive learning based methods Moco-v3 [14] and DINO [10], our model can significantly outperform them by 1.5% and 1.9% respectively. Compared with iBOT and SIM which also use contrastive objective in MIM, our CMAE achieves higher performance with a gain of 0.7% and 0.8%, respectively. Above results strongly evidence the superiority of CMAE.

When we replace the vanilla ViT encoder with a hybrid convolutional ViT, as done in ConvMAE [23], CMAE further improves to 85.0% and 85.3% with the pre-training of 800 epochs and 1600 epochs, respectively. These results surpass those of ConvMAE under the same pre-training setting by 0.4% and 0.7% respectively, verifying the excellent extendibility of CMAE to various network structures.

Remarkably, CMAE can gain a noticeable improvement with a prolonged training schedule (from 800 epochs to 1600 epochs) while ConvMAE is observed to saturate at 800 epochs. This result suggests the stronger capability of CMAE on learning better representations.

### C. Transfer Learning

To further validate the transferability of CMAE, we follow previous methods to evaluate pre-trained models on the semantic segmentation dataset ADE20K [64], the object detection dataset COCO2017 [37] and classification datasets.

**Semantic Segmentation:** ADE20K [64] has 25,562 images of 150 fine-grained categories. We adopt Upernet [55] as the default model for this task, following the settings of compared methods. The backbone ViT-B is initialized from pre-training while other modules are initialized with the Xavier [24] initialization. The model is fine-tuned on the training set of ADE20K and tested on standard validation split.

Following previous works, we report the Mean Intersection over Union (mIoU) performance of CMAE in Table III(a). We notice that CMAE significantly surpasses MAE by 2.9%, which verifies the stronger transferability of CMAE. Besides, CMAE also improve by 1.0% and 0.8% compared with iBOT [65] and CAE [15] respectively. With the same hybrid ViT backbone, CMAE significantly outperforms ConvMAE by 1.8%. Remarkably, CMAE sets a new state-of-the-art result of 52.5 by surpassing all competing methods with a large margin.

**Object Detection and Segmentation:** We adopt the widely used object detection and instance segmentation framework Mask-RCNN [28], [36] for benchmarking on this task. ViT-B is used as the backbone and initialized with our pre-trained model. Following MAE, we fine-tune the model on COCO train2017 split, and report box AP for object detection and mask AP for instance segmentation on val2017 split. We fine-tune the model for 100 epochs. The base learning rate is $1e^{-4}$ with a cosine annealing schedule, and the weight decay is set to 0.1.

The comparison results with other self-supervised learning methods are shown in Table III(b). As one can see, CMAE improves over MAE from 51.7 to 52.4 on AP$^b$ and from 45.9 to 46.5 on AP$^m$. With the hybrid ViT structure, CMAE consistently surpasses the competing method ConvMAE: AP$^b$ increases from 52.5 to 52.9 and AP$^m$ increases from 46.5 to 47.0. Above promising results again verify the effectiveness of our method.

**Classification Tasks:** To further study transfer learning on classification tasks, we validate our model on the iNaturalists [48] and Places [63]. Experiments across four classification tasks on these datasets showed in Table V consistent improvements of 1.0% to 1.7% in top-1 accuracy over the MAE [30]. These results provide further evidence for the efficacy of our method in enhancing the discriminative capabilities of pre-training model.

#### D. Method Analysis

To understand the effects of key components and validate design choices we adopt in CMAE, we conduct a series of ablation experiments. Unless otherwise stated, we report the performance of our model with 300 pre-training epochs in this subsection. The ablation results are listed in Table IV. In the following, we verify the effectiveness of our main design ideas, then conduct ablation experiments for each component separately.

**Ablation of Components:** In Table IV(a), we show how each component, i.e., contrastive learning, pixel shifting data augmentation and feature decoder affects model’s performance. We start with a vanilla implementation of contrastive learning on MAE. Specifically, following the input generation approach in contrastive methods, random cropped regions with masking are fed into online/momentum encoder. The same contrastive objective as introduced in Section III-C are optimized between the output of online encoder and momentum encoder. As can be seen from Table IV(a), such an intuitive approach only leads to marginal improvement (0.2%). Apparently, the power of contrastive learning is not fully unleashed due to ignoring its compatibility with MIM. By using the proposed moderate data augmentation, i.e., pixel shifting, the result can increase from 83.1% to 83.6%, which evidences the advantage of pixel shifting. Moreover, applying feature decoder further boosts the model’s learning capability by improving the performance to 83.8%, demonstrating its effectiveness in our method.

**Contrastive Loss:** To explore the effect of contrastive loss in CMAE, we experiment with various loss weights, i.e., $\lambda_c$ in (8). The results are shown in Table IV(d). Note CMAE degenerates to the baseline MAE when loss weight is 0. When increasing...
that both contrastive loss and reconstructive loss are critical phenomena of imbalanced training occurs which adversely affects the final performance. This experiment demonstrates that both contrastive loss and reconstructive loss are critical for learning capable representations. We therefore set $\lambda_c = 1$ throughout our experiments.

We also conduct controlled experiments with different contrastive loss forms to compare their influences on pre-training. Under the same configuration, we observe that the model trained with InfoNCE loss achieves higher performance than BYOL-style loss (83.8% versus 83.4%). This result suggests that the way of utilizing negative samples in InfoNCE is more effective in our method.

**Pixel Shifting Augmentation:** In this section, we ablative on the importance of data augmentations. In contrast to the common practices of applying heavy data augmentation in contrastive learning, we find a moderate data augmentation is more effective in aligning contrastive learning and MIM. We divide data augmentation methods into two kinds: spatial transfer and color transfer, and evaluate their effect respectively. For spatial transfer, we compare our proposed pixel shifting with the commonly used randomly resized cropping. For color transfer, we compare two cases, i.e., with or without using color jittering for the momentum branch.

As can be seen from Table IV(b), pixel shifting significantly surpasses random crop (83.4% versus 83.0%). The superiority of pixel shifting should be attributed to its ability of generating more plausible positive views. As introduced in Section III-B, this property helps contrastive learning better collaborate with MIM in our framework. By using color transfer, the result further improves to 83.8%, suggesting color transfer is complementary to our method.

We investigate the impact of different pixel shift ranges by varying the maximum allowable shift. Intuitively, larger shift ranges introduce greater misalignment between the two augmented views. As evident in Table VI, excessive shifts severely degrade model performance, conforming to our hypothesis that severely misaligned positive pairs introduce noise detrimental to contrastive learning. The results demonstrate an optimal balance exists between view diversity and alignment. Based on the observed performance across different shift ranges, we select a maximum pixel shift of 31 as the default parameter setting to maximize contrastive learning while preserving sufficient alignment.

**Feature Decoder:** Different from existing works, we introduce a feature decoder to recover the features of masked patches when performing contrastive learning. To investigate its effectiveness, we present experiments under following two settings: sharing the weight between feature decoder and pixel decoder or not, and changing the depth of feature decoder.

In Table IV(c), number “0” means not using feature decoder, i.e., the output of the online encoder which contains only the features of visible tokens is used for contrastive learning. Under this setting, our method performs worse than using a lightweight two-layer feature decoder. When increasing the depth of feature decoder, there is no significant impact on performance. However, when the depth increases to 8, we obtain a trivial solution, possibly due to the optimization difficulty caused by deeper structure. To strike a balance between efficiency and effectiveness, we set the depth to be 2. Besides, when the feature decoder shares the weight with the pixel decoder, the method performs the worst. A plausible explanation is that the two branches have different targets, thus should adopt independent weights.

**Masking Ratio for the Momentum Branch:** In this experiment, we investigate whether masking a portion of image patches for the momentum branch affects the model performance. Following previous works, we select a set of masking ratios, including $\{0, 0.25, 0.5, 0.65, 0.75\}$ for the momentum branch and see how the performance changes. As shown in Table IV(e), one can observe that using the complete set of the image tokens yields the best results. A possible reason is that: since the aim of adding the momentum branch is to provide our model with the contrastive supervision, incorporating the full semantics of an image is preferred. Otherwise, the masked input with degenerated semantic information may lead to a sub-optimal solution in contrastive learning. Based on this observation, the momentum branch in our model uses the whole image as input throughout our experiments.

**Convergence Speed:** To further show our method’s effectiveness, we compare the convergence behavior of CMAE and MAE when fine-tuning on ImageNet-1k. The pre-trained weights with 1600 epochs are used as initialization. As shown in Fig. 5, we observe that CMAE converges much faster compared with MAE: with only 55 fine-tuning epochs, CMAE already surpasses the

**TABLE VI**

| Shift value | random crop | 0 | 0.15 | 0.31 | 0.47 | 0.63 | 32-47 | 48-63 |
|-------------|-------------|---|------|------|------|------|-------|-------|
| Accuracy    | 83.29       | 83.68 | 83.78 | 83.82 | 83.71 | 83.64 | 83.54 | 83.48 |

"Random crop" serves as the baseline method for comparison against our proposed pixel shift approach.

Fig. 5. Convergence speed compared with MAE. All models are pre-trained with 1600 epochs.
HUANG et al.: CONTRASTIVE MASKED AUTOENCODERS ARE STRONGER VISION LEARNERS 2515

Fig. 6. Analysis of features of pre-training models. In terms of intra-class distance, our model achieves a lower standard deviation relative to MAE on the ImageNet-1k validation set. With respect to inter-class center distance, our model displays greater inter-class distances than MAE for 811 out of the 1000 categories, and manifests a reduced standard deviation of distances among category centers.

final performance of MAE. This result demonstrates that the representations learned by CMAE can be more easily adapted for specific tasks, an appealing property which is in line with the purposes of self-supervised pre-training.

E. Partial Fine-Tuning and Linear Probing

In the context of task-specific training, both partial fine-tuning [30], [41], [58], [62] and linear probing methodologies retain the majority of the model components in a frozen state. However, a key distinction lies in the nature of the head being tuned: partial fine-tuning employs a non-linear head, whereas linear probing utilizes a linear one. As underscored by [30], given that linear probing exhibits minimal correlation with transfer learning performance, partial fine-tuning emerges as a superior protocol for the evaluation of non-linear, yet more potent, representations. In light of these observations, our study also places emphasis on the partial fine-tuning metric.

Specifically, we follow the experimental settings of [30] to ablate the CMAE base model with 1600 epoch pre-training. As shown in Fig. 4(a), the performances of our model are consistently better than MAE in all tested settings, e.g., when fine-tuning one block, we get a 2.5% gain over MAE. Above results demonstrate that our model can effectively improve the representation quality of baseline method. Note when the number of fine-tuned blocks is “0”, it degenerates to linear probing. In this case, our model achieves significant improvement (5.9%) over MAE. These results indicate that our method is able to improve the representation quality under both evaluation metrics. Furthermore, in comparison to the typical contrastive model MoCo-V3 [14], MoCo-V3 exhibits superior performance in the linear probing setting. However, under the partial fine-tuning setting, CMAE surpasses MoCo-V3 in all aspects, particularly when fine-tuning only one block, where CMAE yields a 3.8% enhancement. This also serves as evidence that the features learned by our model are of higher quality.

F. Model Scaling

To study the scalability of our method for models of different sizes, we adopt ViT-small, ViT-base, and ViT-large as encoders and report their performance on ImageNet-1k fine-tuning. As shown in Fig. 4(b), CMAE can consistently boost the performance of MAE at all scales. These results clearly demonstrate the excellent scalability of CMAE.

G. Feature Analysis

In order to more effectively scrutinize the features obtained by our model, we utilize the ViT-base model for our investigation. Following the completion of pre-training, we extract features from the ImageNet-1k validation set and compute the following metrics:

- Average intra-class distance: This metric measures the mean distance between all pairs of images within the same class.
- Standard deviation of intra-class distances: This metric measures the variation in distances between images in the same class.
- Average inter-class distance: This metric measures the mean distance between all pairs of images from different class centers.
- Standard deviation of inter-class distances: This metric measures the variation in distances between images from different class centers.

When we compute the average intra-class distance, CMAE achieves a lower average intra-class distance of 0.0377 than MAE’s 0.0380. Furthermore, as shown in Fig. 6(a), CMAE has a smaller standard deviation of intra-class distances than MAE (0.0189 versus 0.0371). These results suggest that the features extracted by CMAE are more compactly clustered in the latent space. Regarding inter-class distances, we compute the average distance of each class center to other class centers and the standard deviation of distances to other class centers. As shown in Fig. 6(b) and (c), CMAE demonstrates larger average inter-class distances (0.0340 versus 0.0309) and a smaller standard deviation for inter-class distances (0.0152 versus 0.0310). This indicates that the features extracted by CMAE have a more uniform distribution for each category in the latent space and larger inter-class distances.

In summary, the aforementioned results provide evidence that our model is capable of learning superior visual representations with enhanced discriminability compared to MAE.
V. CONCLUSION

This paper introduces a novel self-supervised learning framework named contrastive masked autoencoder (CMAE) which aims to improve the representation quality of MIM by leveraging contrastive learning. In CMAE, we propose two novel designs from the perspective of input generation and architectures respectively to harmonize MIM and contrastive learning. Through extensive experiments, it is demonstrated that CMAE can significantly improve the quality of learned representation in pre-training.

Notably, on three well-established downstream tasks, i.e., image classification/segmentation/detection, CMAE achieves state-of-the-art performance. In the future, we will investigate the scaling up of CMAE to larger datasets and incorporate image-dense caption as another view for contrastive learning training based on CMAE.

REFERENCES

[1] M. Assran et al., “Masked siamese networks for label-efficient learning,” 2022, arXiv:2204.07141.
[2] P. Bachman, R. D. Hjelm, and W. Buchwalter, “Learning representations by maximizing mutual information across views,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2019, Art. no. 1392.
[3] H. Bao, L. Dong, and F. Wei, “BEiT: BERT pre-training of image transformers,” 2021, arXiv:2106.08254.
[4] A. Bardes, J. Ponce, and Y. LeCun, “VICReg: Variance-invariance covariance regularization for self-supervised learning,” 2021, arXiv:2105.04906.
[5] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, “Signature verification using a “Siamese” time delay neural network,” in Proc. Int. Conf. Neural Inf. Process. Syst., 1993, pp. 737–744.
[6] Z. Cai and N. Vasconcelos, “Cascade R-CNN: High quality object detection and instance segmentation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 43, no. 5, pp. 1483–1498, May 2021.
[7] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, “Deep clustering for unsupervised learning of visual features,” in Proc. Eur. Conf. Comput. Vis., Springer, 2018, pp. 132–149.
[8] M. Caron, P. Bojanowski, J. Mairal, and A. Joulin, “Unsupervised pre-training of image features on non-curated data,” in Proc. Int. Conf. Comput. Vis., 2019, pp. 2599–2598.
[9] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, “Unsupervised learning of visual features by contrasting cluster assignments,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2020, pp. 9912–9924.
[10] M. Caron et al., “Emerging properties in self-supervised vision transformers,” in Proc. Int. Conf. Comput. Vis., 2021, pp. 9650–9660.
[11] M. Chen et al., “Generative pretraining from pixels,” in Proc. Int. Conf. Mach. Learn., 2020, pp. 1691–1703.
[12] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” in Proc. Int. Conf. Mach. Learn., 2020, pp. 1597–1607.
[13] X. Chen and K. He, “Exploring simple siamese representation learning,” in Proc. Comput. Vis. Pattern Recognit., 2021, pp. 15750–15758.
[14] X. Chen, S. Xie, and K. He, “An empirical study of training self-supervised vision transformers,” in Proc. Int. Conf. Comput. Vis., 2021, pp. 9640–9649.
[15] X. Chen et al., “Context autoencoder for self-supervised representation learning,” 2022, arXiv:2202.03026.
[16] M. Cuturi, “Sinkhorn distances: Lightspeed computation of optimal transport,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2013, pp. 2292–2300.
[17] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in Proc. Comput. Vis. Pattern Recognit., 2005, pp. 886–893.
[18] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in Proc. Comput. Vis. Pattern Recognit., 2009, pp. 248–255.
[19] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” 2018, arXiv:1810.04805.
[20] X. Dong et al., “PeCo: Perceptual codebook for BERT pre-training of vision transformers,” 2021, arXiv:2111.12710.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Zhicheng Huang received the master’s degree in artificial intelligence from the University of Science and Technology Beijing, China, in 2019. He is currently working toward the PhD degree with the School of Automation and Electrical Engineering. His research interests include self-supervised learning, visual-language pre-training, and computer vision.

Xiaojie Jin received the PhD degree from the National University of Singapore, in 2018. He is working as a research scientist with Bytedance Inc. in the USA. His current research interests include self-supervised learning, multi-modal understanding, intelligent video editing, and efficient deep model. His research works have been published in top-tier conferences/journals, including ICCV, ECCV, CVPR, ICML, NeurIPS, ICLR, IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Transactions on Neural Networks and Learning Systems, etc.

Chengze Lu received the BE degree from Xidian University, in 2020. He is currently working toward the master’s degree with the College of Computer Science, Nankai University, under the supervision of Prof. Ming-Ming Cheng. His research interests include deep learning and computer vision.

Qibin Hou received the PhD degree from the School of Computer Science, Nankai University. Then, he worked with the National University of Singapore as a research fellow. Now, he is an associate professor with the School of Computer Science, Nankai University. He has published more than 20 papers on top conferences/journals, including the IEEE Transactions on Pattern Analysis and Machine Intelligence, CVPR, ICCV, NeurIPS, etc. His research interests include deep learning, image processing, and computer vision.

Dongmei Fu received the MS degree from North-western Polytechnical University, in 1984, and the PhD degree in automation science from the University of Science and Technology Beijing (USTB), China, in 2006, where she is currently a professor and a doctoral supervisor. She has taken charge of several national projects about corrosion data mining and infrared image processing. Her current research interests include automation control theory, image processing, and data mining.

Xiaohui Shen received the BS and MS degrees from the Department of Automation, Tsinghua University, China, and the PhD degree from the Department of EECS, Northwestern University. He is a research manager with ByteDance Inc. Before that, he was a senior research scientist with Adobe Research. His research interests include computer vision and deep learning.

Jiashi Feng received the PhD degree from the National University of Singapore (NUS), in 2014. He is currently a research scientist with ByteDance. Before joining ByteDance, he was an assistant professor with the Department of Electrical and Computer Engineering, National University of Singapore and a postdoc researcher with the EECS Department and ICSI, University of California, Berkeley. His research areas include deep learning and their applications in computer vision. His recent research interest focuses on deep learning models, representation learning, and 3D vision. He received the best technical demo award from ACM MM 2012, best paper award from TASK-CV ICCV 2015, best student paper award from ACM MM 2018. He is also the recipient of Innovators Under 35 Asia, MIT Technology Review 2018. He served as the area chairs for NeurIPS, ICML, CVPR, ICLR, WACV, and program chair for ICMR 2017.